



Universidade do Estado do Rio de Janeiro
Centro de Tecnologia e Ciências
Instituto Politécnico

Octavio Vieira de Aguiar

**Algoritmo de otimização de sistemas modelados por equações
diferenciais ordinárias aplicado a processos de fermentação em
batelada alimentada**

v. 1

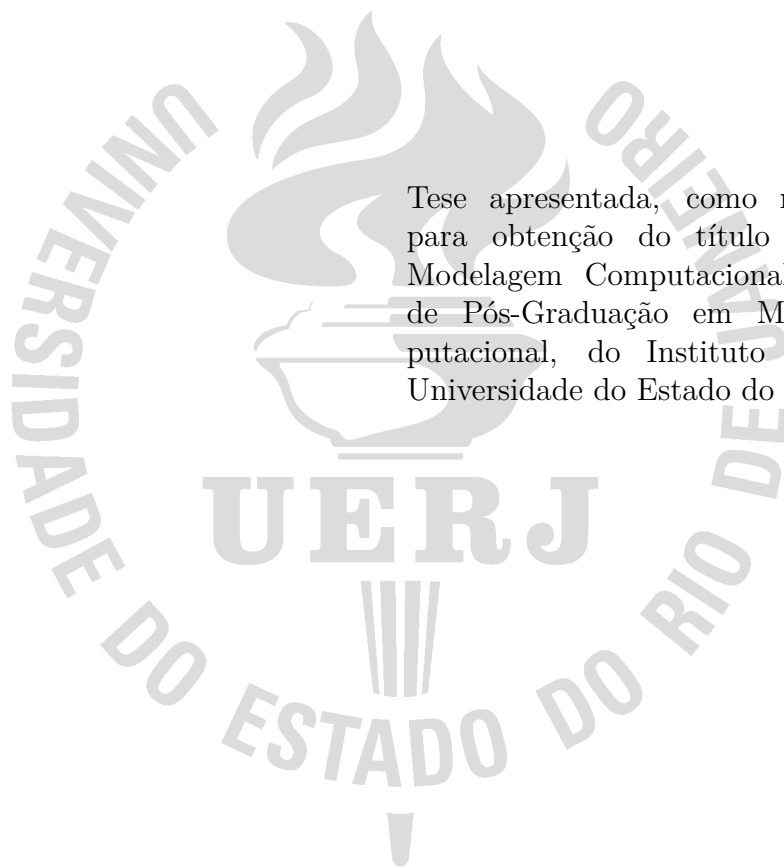
Nova Friburgo

2023

Octavio Vieira de Aguiar

Algoritmo de otimização de sistemas modelados por equações diferenciais ordinárias aplicado a processos de fermentação em batelada alimentada

v. 1



Tese apresentada, como requisito parcial para obtenção do título de Doutor em Modelagem Computacional, ao Programa de Pós-Graduação em Modelagem Computacional, do Instituto Politécnico, da Universidade do Estado do Rio de Janeiro.

Orientador: Roberto Pinheiro Domingos

Orientador: Gustavo Mendes Platt

Nova Friburgo

2023

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC/E

A282 Aguiar, Octavio Vieira de.
Algoritmo de otimização de sistemas modelados por equações
diferenciais ordinárias aplicado a processos de fermentação em
batelada alimentada / Octavio Vieira de Aguiar. - 2023.
2 v. (340 f.) : il.

Orientadores: Roberto Pinheiro Domingos
Gustavo Mendes Platt

Tese (doutorado) – Universidade do Estado do Rio de Janeiro,
Instituto Politécnico.

1. Equações diferenciais ordinárias – Teses. 2. Otimização
matemática – Teses. 3. Fermentação – Teses. I. Domingos, Roberto
Pinheiro. II. Platt, Gustavo Mendes. III. Universidade do Estado
do Rio de Janeiro. Instituto Politécnico. IV. Título.

CDU 517.9

Bibliotecária Pâmela Lisboa CRB7/ 5965

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta tese,
desde que citada a fonte.

Assinatura

Data

Octavio Vieira de Aguiar

Algoritmo de otimização de sistemas modelados por equações diferenciais ordinárias aplicado a processos de fermentação em batelada alimentada

Tese apresentada, como requisito parcial para obtenção do título de Doutor em Modelagem Computacional, ao Programa de Pós-Graduação em Modelagem Computacional, do Instituto Politécnico, da Universidade do Estado do Rio de Janeiro.

Aprovada em 28 de agosto de 2023.

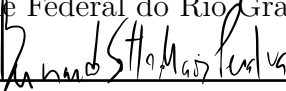
Banca Examinadora:



Prof. Dr. Roberto Pinheiro Domingos (Orientador)
Instituto Politécnico - UERJ



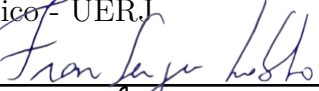
Prof. Dr. Gustavo Mendes Platt (Orientador)
Universidade Federal do Rio Grande




Prof. Dr. Bernardo Sotto-Maior Peralva
Instituto Politécnico - UERJ



Prof. Dr. Gustavo Barbosa Libotte
Instituto Politécnico - UERJ



Prof. Dr. Fran Sérgio Lobato
Universidade Federal de Uberlândia



Prof. Dr. Fernando Cunha Peixoto
Instituto Militar de Engenharia

Nova Friburgo

2023

DEDICATÓRIA

Dedico este trabalho a Deus e a todos aqueles que estiveram ao meu lado, me ajudaram, apoiaram e acreditaram em mim. Sem vocês, nada teria valido a pena.

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me concedido viver essa jornada, me dado força e sabedoria para concluir mais esta etapa da minha vida e por me permitir continuar sempre sonhando com o futuro.

À minha família, pelo apoio e compreensão durante esta caminhada, especialmente nos momentos em que estive ausente, não podendo ser tão participativo quanto gostaria na vida de cada um de vocês.

Ao meu orientador Roberto Pinheiro Domingos, grande amigo desde os tempos da minha graduação, que durante toda a nossa amizade e convívio, sempre foi solícito, divertido e responsável, contribuindo sempre para meu desenvolvimento acadêmico e pessoal. Muito obrigado por ter aceitado me orientar, pelo apoio durante toda esta jornada, pela dedicação em revisar cada página deste trabalho e por ter confiado e acreditado em mim.

Ao meu orientador Gustavo Mendes Platt, que aceitou participar desta jornada, sendo sempre prestativo e cordial durante nosso convívio. Obrigado por ter acreditado neste trabalho, pelo empenho em revisá-lo, pela disponibilidade em me orientar, pela confiança depositada em mim e pela amizade construída ao longo destes anos.

Ao professor Hermes Alves Filho, pelo apoio e prestatividade, sendo sempre solícito às minhas demandas, buscando sempre, dentro das possibilidades, atendê-las da melhor forma possível. Também não poderia deixar de agradecer pela companhia e pelas conversas no laboratório, que sempre somaram ao meu crescimento pessoal e científico, e pela amizade desenvolvida ao longo desta trajetória.

Ao professor Bernardo Sotto-Maior Peralva, pela convívio, sempre divertido, no laboratório, pelas trocas de ideias, que sempre foram enriquecedoras, pela amizade edificada ao longo desta caminhada e por aceitar o convite para compor esta banca examinadora, abdicando do seu tempo para revisar e contribuir com esta tese.

Aos professores Gustavo Barbosa Libotte, Fran Sérgio Lobato e Fernando Cunha Peixoto, por terem aceitado o convite para participar desta banca examinadora e por dispensarem seu tempo para avaliar e contribuir com este trabalho.

Aos membros da secretaria de pós de graduação de modelagem computacional, em especial a Roberto Carlos Candeias Vieira, Melissa de Almeida Kalas e Marcia Maria Cidade, pela disponibilidade, celeridade e solicitude ao atender as minhas demandas e pela amizade construída ao longo destes anos. Obrigado e parabéns pelo excelente trabalho.

Aos integrantes da biblioteca do instituto politécnico, em especial a Pâmela Lisbôa Flegr, Cleide Sancho, Daiany da Silva Pereira Pinto e Josué Anadias Santos, por toda presteza e dedicação dispensadas na revisão normativa desta tese e no atendimento as demandas que apresentei, além da amizade e convívio edificados nesta caminhada. Congratulações pelo trabalho e gratidão por tudo.

Aos membros da guarda patrimonial e vigilantes, que durante essa jornada, me fizeram companhia nas longas noites em que estive trabalhado no laboratório. Obrigado pela animação, cordialidade, parceria e amizade, que estiveram sempre presentes durante todos os anos desta jornada.

Aos meus colegas de pós-graduação, que caminharam ao meu lado, nas disciplinas, avaliações, seminários, reuniões e tantas outras atividades. Muito obrigado pela convivência enriquecedora, pelos conhecimentos trocados, pelo apoio nas horas de necessidade, pela parceria nas atividades e pela amizade que desenvolvemos.

À todos os professores, técnicos, funcionários e colaboradores do instituto politécnico que comigo conviveram e compartilharam momentos durante esta jornada. Tenham certeza de que cada um de vocês contribuiu um pouco para formar quem sou hoje.

À CAPES, pelo apoio financeiro, pois o presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Finalmente, agradeço a todas as pessoas que de alguma forma contribuíram com o desenvolvimento desta tese, direta ou indiretamente, seja por meio de sugestões, críticas, ideias, apoio, incentivo ou qualquer outra forma de auxílio.

À todos vocês, o meu muito obrigado! Tenham a certeza de que cada um estará para sempre marcado na minha história e nas minhas lembranças.

Sinceramente,

Octavio Vieira de Aguiar

Tudo concorre para o bem daqueles que amam a Deus.

São Paulo

RESUMO

AGUIAR, O. V. *Algoritmo de otimização de sistemas modelados por equações diferenciais ordinárias aplicado a processos de fermentação em batelada alimentada*. 2023. 340 f. Tese (Doutorado em Modelagem Computacional) – Instituto Politécnico, Universidade do Estado do Rio de Janeiro, Nova Friburgo, 2023.

Os modelos matemáticos permitem a simulação de diversos processos, em diferentes configurações, proporcionando estimar seus resultados, antes de serem experimentalmente executados. Neste sentido, os processos modelados por equações diferenciais ordinárias apresentam grande importância para várias áreas do conhecimento. Existem inúmeras situações em que se deseja otimizar tais processos afim de minimizar seus custos e maximizar seu desempenho. Neste contexto, a ausência de metodologias abrangentes capazes de otimizar estes processos leva a criação de inúmeras metodologias distintas voltadas à otimização de apenas um caso particular destes processos. Tais metodologias comumente empregam o uso de diferentes ferramentas que operam em conjunto por meio de algoritmos de integração, fatores que provocam maior lentidão no processo de otimização. Neste contexto, se encontram os processos de fermentação, que possuem considerável relevância industrial e acadêmica. Por esta razão, existe um grande interesse na otimização destes processos, a fim de que estes tornem-se mais baratos e produtivos. Neste trabalho, foi proposto um algoritmo para otimização de processos modelados por equações diferenciais ordinárias, que permite a modelagem de problemas de otimização mono e multiobjetivo sobre este tipo de processo e sua solução por diferentes métodos de otimização, interpolação e para EDOs. Além da proposição do algoritmo, foi desenvolvida uma biblioteca de *software* que o implementa, construída integralmente em uma única plataforma de programação, evitando os custos computacionais e temporais presentes em metodologias que envolvem a integração entre mais de uma ferramenta, além de dispor de uma série de métodos de otimização, interpolação e para EDOs, dos quais o usuário pode fazer uso. Esta tese também se propôs a realizar dois estudos de caso envolvendo otimização de processos de fermentação em batelada alimentada, que empregam o algoritmo e ferramenta desenvolvidos nela. O primeiro estudo de caso envolveu a otimização de um processo de geração de L-lisina por meio da resolução de problemas propostos em outro trabalho, cujos os resultados obtidos foram analisados, discutidos e comparados aos logrados neste último. Os resultados deste estudo mostraram que soluções que apresentam maior produtividade mostram menor rendimento e menor tempo de operação; e vice-versa, além de mostrarem a confiabilidade do algoritmo, uma vez que os resultados aqui obtidos assemelham-se aos obtidos no trabalho que propôs os problemas. O segundo estudo de caso envolveu a otimização de um processo de produção alcoólica de uma planta industrial real por meio da proposição e resolução de problemas relevantes para ao processo em questão, dos quais os resultados obtidos foram analisados e discutidos. Os resultados deste estudo mostraram que soluções que geram mais etanol preservam menos levedura e vice-versa, além de apresentar soluções que fornecem relevante ganho de tempo e de preservação de levedura sem prejuízo na geração de etanol, quando comparadas à solução em uso na planta.

Palavras-chave: Otimização. Fermentação. Equações diferenciais ordinárias. Algoritmos.

ABSTRACT

AGUIAR, O. V. *Ordinary differential equations modeled system optimization algorithm applied in fed-batch fermentation processes*. 2023. 340 f. Tese (Doutorado em Modelagem Computacional) – Instituto Politécnico, Universidade do Estado do Rio de Janeiro, Nova Friburgo, 2023.

Mathematical models allow the simulation of sundry processes, within distinct shapes, providing estimation of their results, before being experimentally executed. In this sense, the processes modeled by ordinary differential equations are of great importance for several knowledge areas. There are countless situations in which it is want to optimize such processes, in order to minimize costs and maximize performance. In this scenario, the absence of comprehensive methodologies capable of optimizing these processes leads to the creation of numberless different methodologies aimed at optimizing only one particular case of these processes. Such methodologies commonly employ the use of different tools that operate together through an integration algorithms, factors that cause slowness in the optimization process. The processes of fermentation are found in this context, which have considerable relevance for the industrial and academic environments. For this reason, there is great interest in optimizing such processes, in order to make them cheaper and more productive. In this work, an algorithm for optimization of processes modeled by ordinary differential equations was proposed, which allows the modeling of single and multiobjective optimization problems on this type of process and its solution by different methods of optimization, interpolation and for ODEs. In addition to the proposition of the algorithm, a software library was developed that implements it, built entirely on a single programming platform, avoiding the computational and time costs present in methodologies that involve the integration between more than one tool, in addition to have a series of optimization, interpolation and ODE methods available to the user. This thesis also proposed to carry out two case studies involving optimization of fed-batch fermentation processes, which employ the algorithm and tool developed therein. The first case study involved the optimization of an L-lysine generation process by solving problems proposed in another work, whose results were analyzed, discussed and compared to those achieved in the latter. The results of this study showed that solutions that have higher productivity show lower performance and shorter operating time; and vice versa, in addition to showing the reliability of the algorithm, since the results obtained here are similar to those obtained in the work that proposed the problems. The second case study involved the optimization of an alcoholic production process in a real industrial plant through the proposition and resolution of problems relevant to the process in question, from which the results obtained were analyzed and discussed. The results of this study showed that solutions that generate more ethanol preserve less yeast and vice versa, in addition to presenting solutions that provide significant gain in time and yeast preservation without prejudice to ethanol generation, when compared to the solution in use in the plant.

Keywords: Optimization. Fermentation. Ordinary differential equations. Algorithms.

LISTA DE FIGURAS

Figura 1	- Equivalência entre problemas de minimização e maximização.	46
Figura 2	- Minimizador global e mínimo global.	53
Figura 3	- Um minimizador local e seu mínimo local correspondente.	54
Figura 4	- Minimizador local fraco e minimizador local estrito.	55
Figura 5	- Fronteira de Pareto.	66
Figura 6	- Conjunto solução de um problema de otimização multiobjetivo e conceitos correlatos.	67
Figura 7	- Ótimo ideal.	68
Figura 8	- Fronteira local de Pareto.	71
Figura 9	- Conjunto solução local de um problema de otimização multiobjetivo e conceitos correlatos.	72
Figura 10	- Convergência e diversidade.	74
Figura 11	- Conflito entre convergência e diversidade.	75
Figura 12	- Cromossomo codificado computacionalmente.	78
Figura 13	- Gene e <i>locus</i> em um cromossomo codificado computacionalmente.	79
Figura 14	- Alelo em um cromossomo codificado computacionalmente.	79
Figura 15	- População de cromossomos codificados computacionalmente.	81
Figura 16	- Diagrama de estados do modelo da catraca.	85
Figura 17	- Representação esquemática do funcionamento do operador de mutação padrão.	87
Figura 18	- Representação esquemática do funcionamento do operador de recombinação de um ponto.	89
Figura 19	- Representação esquemática do funcionamento do operador de recombinação de múltiplos pontos.	90
Figura 20	- Representação esquemática do funcionamento do operador de recombinação uniforme.	91
Figura 21	- Representação esquemática do funcionamento do operador de recombinação aritmética simples.	93
Figura 22	- Representação esquemática do funcionamento do operador de recombinação aritmética individual.	94
Figura 23	- Representação esquemática do funcionamento do operador de recombinação aritmética total.	95
Figura 24	- Fluxograma genérico de um algoritmo genético.	100
Figura 25	- Fluxograma simplificado de um algoritmo de estimação e distribuição.	106
Figura 26	- Etapas da evolução diferencial.	115
Figura 27	- Distância de aglomeração.	122

Figura 28 - Etapas do NSGA-II.	124
Figura 29 - Ponto mais ao centro.	130
Figura 30 - Principais processos de uma planta sucroalcooleira.	142
Figura 31 - Fermentação em batelada.	144
Figura 32 - Modelo prático da fermentação em batelada.	145
Figura 33 - Fermentação em batelada alimentada.	146
Figura 34 - Processo industrial de fermentação em batelada alimentada.	147
Figura 35 - Fermentação contínua.	148
Figura 36 - Processo industrial de fermentação contínua.	150
Figura 37 - Fórmulas estruturais da D-glicose e D-frutose.	153
Figura 38 - Concentração de células em cada uma das fases.	155
Figura 39 - Processo industrial de fermentação de L-lisina.	169
Figura 40 - Esquema simplificado do metabolismo central e vias de síntese de lisina na <i>Corynebacterium glutamicum</i>	172
Figura 41 - Etapas do CODEMSOA.	179
Figura 42 - Diagrama de classes das estruturas utilitárias.	181
Figura 43 - Diagrama de classes das estruturas genéricas.	182
Figura 44 - Diagrama de classes do solucionador de equações diferenciais ordinárias.	184
Figura 45 - Diagrama de classes do interpolador de soluções.	186
Figura 46 - Diagrama de classes do otimizador.	187
Figura 47 - Diagrama de classes total do modelo.	189
Figura 48 - Diagrama de engenharia de um reator de lisina.	191
Figura 49 - Diagrama de engenharia da planta de fermentação.	200
Figura 50 - Exemplo de chamada ao CODEMSOA na biblioteca de código desen- volvida.	338

LISTA DE GRÁFICOS

Gráfico 1 - Produção mundial de etanol.	31
Gráfico 2 - Produção mundial de L-lisina.	32
Gráfico 3 - Fronteira de Pareto do Problema A do Estudo de Caso 1 obtida pelo NSGA-II.	212
Gráfico 4 - Conjunto ótimo de Pareto do Problema A do Estudo de Caso 1 obtido pelo NSGA-II.	213
Gráfico 5 - Produtividade correspondente aos otimizadores de Pareto do Problema A do Estudo de Caso 1 obtidos pelo NSGA-II.	215
Gráfico 6 - Rendimento correspondente aos otimizadores de Pareto do Problema A do Estudo de Caso 1 obtidos pelo NSGA-II.	217
Gráfico 7 - Fronteira de Pareto do Problema A do Estudo de Caso 1 obtida pelo AGE-MOEA.	219
Gráfico 8 - Conjunto ótimo de Pareto do Problema A do Estudo de Caso 1 obtido pelo AGE-MOEA.	220
Gráfico 9 - Produtividade correspondente aos otimizadores de Pareto do Problema A do Estudo de Caso 1 obtidos pelo AGE-MOEA.	222
Gráfico 10 - Rendimento correspondente aos otimizadores de Pareto do Problema A do Estudo de Caso 1 obtidos pelo AGE-MOEA.	224
Gráfico 11 - Fronteira de Pareto do Problema A do Estudo de Caso 1 obtida pelo C-TAEA.	226
Gráfico 12 - Conjunto ótimo de Pareto do Problema A do Estudo de Caso 1 obtido pelo C-TAEA.	227
Gráfico 13 - Produtividade correspondente aos otimizadores de Pareto do Problema A do Estudo de Caso 1 obtidos pelo C-TAEA.	229
Gráfico 14 - Rendimento correspondente aos otimizadores de Pareto do Problema A do Estudo de Caso 1 obtidos pelo C-TAEA.	231
Gráfico 15 - Fronteira de Pareto do Problema B do Estudo de Caso 1 obtida pelo NSGA-II.	234
Gráfico 16 - Conjunto ótimo de Pareto do Problema B do Estudo de Caso 1 obtido pelo NSGA-II.	235
Gráfico 17 - Produtividade correspondente aos otimizadores de Pareto do Problema B do Estudo de Caso 1 obtidos pelo NSGA-II.	237
Gráfico 18 - Tempo final correspondente aos otimizadores de Pareto do Problema B do Estudo de Caso 1 obtidos pelo NSGA-II.	239
Gráfico 19 - Fronteira de Pareto do Problema B do Estudo de Caso 1 obtida pelo AGE-MOEA.	240

Gráfico 20 - Conjunto ótimo de Pareto do Problema B do Estudo de Caso 1 obtido pelo AGE-MOEA.	241
Gráfico 21 - Produtividade correspondente aos otimizadores de Pareto do Problema B do Estudo de Caso 1 obtidos pelo AGE-MOEA.	243
Gráfico 22 - Tempo final correspondente aos otimizadores de Pareto do Problema B do Estudo de Caso 1 obtidos pelo AGE-MOEA.	245
Gráfico 23 - Fronteira de Pareto do Problema B do Estudo de Caso 1 obtida pelo C-TAEA.	246
Gráfico 24 - Conjunto ótimo de Pareto do Problema B do Estudo de Caso 1 obtido pelo C-TAEA.	247
Gráfico 25 - Produtividade correspondente aos otimizadores de Pareto do Problema B do Estudo de Caso 1 obtidos pelo C-TAEA.	249
Gráfico 26 - Tempo final correspondente aos otimizadores de Pareto do Problema B do Estudo de Caso 1 obtidos pelo C-TAEA.	251
Gráfico 27 - Concentrações ao longo do tempo no processo que adota a solução dada no Cenário 1 levantado para o Problema A do Estudo de Caso 1. . . .	255
Gráfico 28 - Concentrações ao longo do tempo no processo que adota a solução dada no Cenário 2 levantado para o Problema A do Estudo de Caso 1. . . .	256
Gráfico 29 - Concentrações ao longo do tempo no processo que adota a solução dada no Cenário 3 levantado para o Problema A do Estudo de Caso 1. . . .	257
Gráfico 30 - Concentrações ao longo do tempo no processo que adota a solução dada no Cenário 1 levantado para o Problema B do Estudo de Caso 1. . . .	260
Gráfico 31 - Concentrações ao longo do tempo no processo que adota a solução dada no Cenário 2 levantado para o Problema B do Estudo de Caso 1. . . .	261
Gráfico 32 - Concentrações ao longo do tempo no processo que adota a solução dada no Cenário 3 levantado para o Problema B do Estudo de Caso 1. . . .	262
Gráfico 33 - Concentrações dos componentes no decorrer do processo fermentativo original considerado no Estudo de Caso 2.	265
Gráfico 34 - Conjunto ótimo de Pareto do Problema A do Estudo de Caso 2 obtido pelo NSGA-II.	266
Gráfico 35 - Fronteira de Pareto do Problema A do Estudo de Caso 2 obtida pelo NSGA-II e suas projeções.	267
Gráfico 36 - Concentrações de açúcares correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo NSGA-II.	268
Gráfico 37 - Concentrações de levedura correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo NSGA-II.	269
Gráfico 38 - Concentrações de etanol correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo NSGA-II.	270

Gráfico 39 - Conjunto ótimo de Pareto do Problema A do Estudo de Caso 2 obtido pelo AGE-MOEA.	271
Gráfico 40 - Fronteira de Pareto do Problema A do Estudo de Caso 2 obtida pelo AGE-MOEA e suas projeções.	272
Gráfico 41 - Concentrações de açúcares correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo AGE-MOEA.	273
Gráfico 42 - Concentrações de levedura correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo AGE-MOEA.	274
Gráfico 43 - Concentrações de etanol correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo AGE-MOEA.	275
Gráfico 44 - Conjunto ótimo de Pareto do Problema A do Estudo de Caso 2 obtido pelo C-TAEA.	276
Gráfico 45 - Fronteira de Pareto do Problema A do Estudo de Caso 2 obtida pelo C-TAEA e suas projeções.	277
Gráfico 46 - Concentrações de açúcares correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo C-TAEA.	278
Gráfico 47 - Concentrações de levedura correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo C-TAEA.	279
Gráfico 48 - Concentrações de etanol correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo C-TAEA.	280
Gráfico 49 - Fronteira de Pareto do Problema B do Estudo de Caso 2 obtida pelo NSGA-II e suas projeções.	283
Gráfico 50 - Conjunto ótimo de Pareto do Problema B do Estudo de Caso 2 obtido pelo NSGA-II.	284
Gráfico 51 - Concentrações de açúcares correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo NSGA-II.	285
Gráfico 52 - Concentrações de levedura correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo NSGA-II.	286
Gráfico 53 - Concentrações de etanol correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo NSGA-II.	287
Gráfico 54 - Fronteira de Pareto do Problema B do Estudo de Caso 2 obtida pelo AGE-MOEA e suas projeções.	289
Gráfico 55 - Conjunto ótimo de Pareto do Problema B do Estudo de Caso 2 obtido pelo AGE-MOEA.	290
Gráfico 56 - Concentrações de açúcares correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo AGE-MOEA.	291
Gráfico 57 - Concentrações de levedura correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo AGE-MOEA.	292

Gráfico 58 - Concentrações de etanol correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo AGE-MOEA.	293
Gráfico 59 - Fronteira de Pareto do Problema B do Estudo de Caso 2 obtida pelo C-TAEA e suas projeções.	295
Gráfico 60 - Conjunto ótimo de Pareto do Problema B do Estudo de Caso 2 obtido pelo C-TAEA.	296
Gráfico 61 - Concentrações de açúcares correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo C-TAEA.	297
Gráfico 62 - Concentrações de levedura correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo C-TAEA.	298
Gráfico 63 - Concentrações de etanol correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo C-TAEA.	299
Gráfico 64 - Concentrações ao longo do tempo na solução original e na solução considerada no Cenário 1 levantado para o Problema A do Estudo de Caso 2.	304
Gráfico 65 - Concentrações ao longo do tempo na solução original e na solução considerada no Cenário 2 levantado para o Problema A do Estudo de Caso 2.	306
Gráfico 66 - Concentrações ao longo do tempo na solução original e na solução considerada no Cenário 1 levantado para o Problema A do Estudo de Caso 2.	310
Gráfico 67 - Concentrações ao longo do tempo na solução original e na solução considerada no Cenário 2 levantado para o Problema B do Estudo de Caso 2.	312

LISTA DE QUADROS

Quadro 1 - Tabela de transição de estados do modelo da catraca.	84
Quadro 2 - Bibliotecas de códigos usadas.	337
Quadro 3 - Configurações dos computadores que executaram as simulações.	340

LISTA DE TABELAS

Tabela 1	- Valores dos parâmetros usados no modelo cinético de Ohno e Nakanishi (1976).	192
Tabela 2	- Instantes de início das etapas do reator no Estudo de Caso 1.	193
Tabela 3	- Valores dos parâmetros de vazão de entrada do reator no Estudo de Caso 1.	194
Tabela 4	- Concentrações iniciais do reator no Estudo de Caso 1.	194
Tabela 5	- Valores dos parâmetros usados no modelo cinético de Ghose e Tyagi (1979).	202
Tabela 6	- Instantes de início das etapas do reator no Estudo de Caso 2.	204
Tabela 7	- Valores dos parâmetros de vazão de entrada e saída do reator no Estudo de Caso 2.	205
Tabela 8	- Capacidade máxima do reator no Estudo de Caso 2.	205
Tabela 9	- Concentrações iniciais do reator no Estudo de Caso 2.	205
Tabela 10	- Soluções Notáveis do Problema A do Estudo de Caso 1.	233
Tabela 11	- Soluções Notáveis do Problema B do Estudo de Caso 1.	252
Tabela 12	- Concentrações dos componentes ao final do processo fermentativo original considerado no Estudo de Caso 2.	264
Tabela 13	- Soluções Notáveis do Problema A do Estudo de Caso 2.	281
Tabela 14	- Soluções Notáveis do Problema B do Estudo de Caso 2.	300
Tabela 15	- Valores das variáveis e objetivos referentes à solução original e a considerada no Cenário 1 levantado para o Problema A do Estudo de Caso 2.	303
Tabela 16	- Valores das variáveis e objetivos referentes à solução original e a considerada no Cenário 2 levantado para o Problema A do Estudo de Caso 2.	305
Tabela 17	- Valores das variáveis e objetivos referentes à solução original e a considerada no Cenário 1 levantado para o Problema B do Estudo de Caso 2.	309
Tabela 18	- Valores das variáveis e objetivos referentes à solução original e a considerada no Cenário 2 levantado para o Problema B do Estudo de Caso 2.	313

LISTA DE ALGORITMOS

Algoritmo 1 - Algoritmo de seleção da roleta.	101
Algoritmo 2 - Algoritmo de seleção por torneio.	103
Algoritmo 3 - Mutação diferencial.	112
Algoritmo 4 - Cruzamento diferencial.	113
Algoritmo 5 - Seleção diferencial.	114
Algoritmo 6 - Evolução diferencial.	115
Algoritmo 7 - Ordenação em frentes não dominadas.	119
Algoritmo 8 - Distância de aglomeração.	122
Algoritmo 9 - NSGA-II.	126
Algoritmo 10 - Normalizar objetivos.	129
Algoritmo 11 - Geometria da frente.	131
Algoritmo 12 - Pontuação de sobrevivência.	134
Algoritmo 13 - AGE-MOEA.	136
Algoritmo 14 - Algoritmo de Valoração Funcional.	176
Algoritmo 15 - Solucionador de Equações Diferenciais Ordinárias.	177
Algoritmo 16 - Interpolador de Soluções.	178
Algoritmo 17 - CODEMSOA.	180

LISTA DE ABREVIATURAS E SIGLAS

θ -DEA	θ Dominance-Based Evolutionary Algorithm
ADAMS	Método Preditor-Corretor Não-Rígido de Adams
AED	Algoritmo de Estimaco e Distribuico
AEOM	Algoritmo Evolucionrio para Otimizaco Multiobjetivo
AG	Algoritmo Gentico
AGE-MOEA	<i>Adaptive Evolutionary Algorithm Based on Non Euclidean Geometry</i>
AKIMA	Interpolaco de Akima
AR-MOEA	Multi-Objective Evolutionary Algorithm Based AR Dominance
BDF	<i>Backward Differentiation Formula</i>
BOA	<i>Bayesian Optimization Algorithm</i>
CMA-ES	<i>Covariance Matrix Adaptation Evolution Strategy</i>
CODEMSOA	<i>Computational Ordinary Differential Equations Modeled System Optimization Algorithm</i>
C-TAEA	<i>Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization</i>
CTC	Centro de Tecnologia Canavieira
DA	Diferenciao Algoritmica
ED	Evoluo Diferencial
EDO	Equaco Diferencial Ordinria
ES	Estratgia Evolutiva
GrEA	<i>Grid-Based Evolutionary Algorithm</i>
HT	<i>Hyper-Threading</i>
ISRES	<i>Improved Stochastic Ranking Evolutionary Strategy</i>
KROGH	Interpolaco de Krogh
MLA	Mtodo dos Lagrangeanos Aumentados
MOEA/D	<i>Multiobjective Evolutionary Algorithm Based on Decomposition</i>
MOGA	<i>Multi-objective Genetic Algorithm</i>
NELDER-MEAD	Mtodo de Nelder-Mead
NPGA	<i>Niched Pareto Genetic Algorithm for Multi-Objective Optimization</i>
NSGA	<i>Non-dominated Sorting Genetic Algorithm</i>
NSGA-II	<i>Non-dominated Sorting Genetic Algorithm II</i>
NSGA-III	<i>Non-dominated Sorting Genetic Algorithm III</i>
OPEP	Organizaco dos Pases Exportadores de Petrleo
PCHIP	Interpolaco de Hermite Cbica por Partes
PCO	Problema de Controle timo
PESA	<i>Pareto Envelope-based Selection Algorithm</i>

PG	Programação Genética
POC	Problema de Otimização Convexa
PPL	Problema de Programação Linear
PPNL	Problema de Programação Não Linear
PROÁLCOOL	Programa Nacional do Álcool
PS	Busca de Padrões
RK-2	Método de Runge-Kutta de 2 ^a Ordem
RK-4	Método de Runge-Kutta de 4 ^a Ordem
RK-8	Método de Runge-Kutta de 8 ^a Ordem
R-NSGA-II	<i>Reference Point Based Non-dominated Sorting Genetic Algorithm II</i>
R-NSGA-III	<i>Reference Point Based Non-dominated Sorting Genetic Algorithm III</i>
RVEA	<i>Reference Vector Guided Evolutionary Algorithm</i>
SMS-EMOA	<i>Multiobjective Selection Based on Dominated Hypervolume</i>
SPEA	<i>Strength Pareto Evolutionary Algorithm</i>
SRES	<i>Stochastic Ranking Evolutionary Strategy</i>
TCA	<i>Tricarboxylic Cycle Acid</i>
UMDA	<i>Univariate Marginal Distribution Algorithm</i>
UML	<i>Unified Modeling Language</i>
U-NSGA-III	<i>Unified Non-dominated Sorting Genetic Algorithm III</i>
VEGA	<i>Vector Evaluated Genetic Algorithm</i>

SUMÁRIO

	INTRODUÇÃO	30
1	OTIMIZAÇÃO	41
1.1	Princípios da Otimização	41
1.1.1	<u>Processo Não Otimizado</u>	41
1.1.2	<u>Processo Otimizado</u>	42
1.1.3	<u>Vantagens do Processo Otimizado</u>	43
1.1.4	<u>Conceito de Otimização</u>	43
1.1.5	<u>Otimização Contínua</u>	43
1.1.6	<u>Otimização Discreta</u>	44
1.2	Otimização Mono-Objetivo	44
1.2.1	<u>Problema de Minimização e Problema Maximização</u>	45
1.2.1.1	Conversão entre Problemas de Minimização e Maximização	45
1.2.2	<u>Otimização Irrestrita</u>	46
1.2.2.1	Representação de um Problema de Otimização Irrestrita	46
1.2.3	<u>Otimização Restrita</u>	47
1.2.3.1	Formato Tradicional do Conjunto Viável	47
1.2.3.2	Restrições Diretas e Funcionais	48
1.2.3.3	Transformações entre Restrições de Igualdade e de Desigualdade	48
1.2.3.4	Classificação de Problemas de Otimização Segundo às Restrições	49
1.2.3.5	Representação de um Problema de Otimização Restrita	49
1.2.4	<u>Problema de Otimização Convexa</u>	50
1.2.5	<u>Programação Linear e Não Linear</u>	51
1.2.6	<u>Conjunto Objetivo de um Problema de Otimização</u>	52
1.2.7	<u>Solução de um Problema de Otimização Mono-Objetivo</u>	52
1.2.7.1	Minimizador Global	52
1.2.7.2	Minimizador Local	52
1.2.8	<u>Minimizador Local Estrito</u>	54
1.2.8.1	Minimizador em Problemas de Otimização Convexa	55
1.3	Otimização Multiobjetivo	55
1.3.1	<u>Problema de Minimização e Problema de Maximização</u>	56
1.3.1.1	Problema Misto de Minimização e Maximização	56
1.3.1.2	Conversão entre Problemas de Minimização e Maximização	57
1.3.2	<u>Validade dos Conceitos da Otimização Mono-Objetivo na Multiobjetivo</u>	57
1.3.3	<u>Representação de um Problema de Otimização Multiobjetivo</u>	58
1.3.4	<u>Solução de um Problema de Otimização Multiobjetivo</u>	59
1.3.4.1	Conflitos entre Objetivos	60

1.3.4.2	Dominância	60
1.3.4.2.1	Dominância Fraca	61
1.3.4.2.2	Dominância	61
1.3.4.2.3	Dominância Forte	61
1.3.4.2.4	Inexistência de Relação de Dominância	62
1.3.4.2.5	Considerações sobre Dominância e Otimização Multiobjetivo	62
1.3.4.3	Otimidade de Pareto	63
1.3.4.3.1	Equilíbrio e Eficiência de Pareto	63
1.3.4.3.2	Otimizador de Pareto	64
1.3.4.3.3	Ótimo de Pareto	64
1.3.4.3.4	Conjunto Ótimo de Pareto	64
1.3.4.3.5	Fronteira de Pareto	65
1.3.4.3.6	Conjunto Solução de um Problema de Otimização	65
1.3.4.3.7	Ótimo Ideal	65
1.3.4.3.8	Solução Local de um Problema de Otimização Multiobjetivo	68
1.3.4.3.9	Otimizador Local de Pareto	69
1.3.4.3.10	Ótimo Local de Pareto	69
1.3.4.3.11	Conjunto Ótimo Local de Pareto	70
1.3.4.3.12	Fronteira Local de Pareto	70
1.3.4.3.13	Conjunto Solução Local de um Problema de Otimização Multiobjetivo	71
1.3.4.4	Qualidade do Conjunto Solução	72
1.3.4.4.1	Convergência	73
1.3.4.4.2	Diversidade	73
1.3.4.4.3	Conflito entre Convergência e Diversidade	73
1.4	Referências com Informações Complementares	74
2	ALGORITMOS EVOLUCIONÁRIOS	77
2.1	Modelagem Computacional dos Conceitos Evolutivos	77
2.1.1	<u>Cromossomo</u>	78
2.1.2	<u>Gene</u>	78
2.1.3	<u>Alelo</u>	78
2.1.4	<u>Codificação</u>	79
2.1.4.1	Codificação Binária	80
2.1.4.2	Codificação em Ponto Flutuante	80
2.1.5	<u>População</u>	80
2.1.6	<u>Aptidão ou <i>Fitness</i></u>	80
2.1.7	<u>Progenitor</u>	81
2.1.8	<u>Reprodução</u>	81
2.1.9	<u>Descendentes</u>	82
2.1.10	<u>Geração</u>	82

2.1.11	<u>Seleção</u>	82
2.1.12	<u>Máquinas de Estados Finitos</u>	83
2.1.12.1	Formalismo Matemático	83
2.1.12.2	Funcionamento das Máquinas de Estados	83
2.1.12.3	Representação Estruturada de Máquinas de Estados	84
2.2	Operadores Evolucionários de Reprodução	85
2.2.1	<u>Tipos de Reprodução</u>	86
2.2.2	<u>Operador de Mutação</u>	86
2.2.2.1	Mutação na Representação Binária	86
2.2.2.1.1	Operador de Mutação Padrão	86
2.2.2.2	Mutação na Representação em Ponto Flutuante	87
2.2.2.2.1	Operador de Mutação Uniforme	87
2.2.2.2.2	Operador de Mutação Gaussiana	87
2.2.3	<u>Operador de Cruzamento</u>	88
2.2.3.1	Cruzamento na Representação Binária	88
2.2.3.2	Operador de Recombinação de Um Ponto	88
2.2.3.3	Operador de Recombinação de Múltiplos Pontos	89
2.2.3.4	Operador de Recombinação Uniforme	90
2.2.3.5	Cruzamento na Representação em Ponto Flutuante	90
2.2.3.6	Recombinação Discreta	91
2.2.3.7	Recombinação Aritmética	92
2.2.3.7.1	Operador de Recombinação Aritmética Simples	92
2.2.3.7.2	Operador de Recombinação Aritmética Individual	93
2.2.3.7.3	Operador de Recombinação Aritmética Total	94
2.2.3.8	Cruzamento com Múltiplos Pais	95
2.2.3.8.1	Operadores de Eleição P-Sexual	95
2.2.3.8.2	Operadores de Recombinação Diagonal	96
2.2.3.8.3	Operadores de Recombinação Massiva	96
2.2.3.8.4	Vantagens do Cruzamento com Múltiplos Pais	96
2.3	Algoritmos Evolutivos	96
2.3.1	<u>Algoritmos Evolucionários Canônicos</u>	97
2.3.1.1	Programação Evolucionária	97
2.3.1.2	Estratégias Evolutivas	98
2.3.1.3	Algoritmos Genéticos	99
2.3.1.3.1	Codificação dos Indivíduos nos Algoritmos Genéticos	100
2.3.1.3.2	Método de Seleção da Roleta	101
2.3.1.3.3	Seleção Elitista	102
2.3.1.3.4	Seleção Baseada em Rank	102
2.3.1.3.5	Seleção por Torneio	102

2.3.2	<u>Algoritmos Evolucionários Não Canônicos</u>	103
2.3.2.1	Programação Genética	103
2.3.2.2	Algoritmos Micro Genéticos	104
2.3.2.3	Algoritmos de Estimção e Distribuição	105
2.3.2.4	Algoritmos Evolutivos para Otimização Multiobjetivo	106
2.3.2.4.1	Métodos Baseados em Pesos	106
2.3.2.4.2	Métodos de Múltiplas Soluções	107
2.3.2.4.3	Dominância	107
2.3.2.4.4	Seleção Baseada em Dominância	108
2.3.2.4.5	Algoritmos Elitistas e Não Elitistas	108
2.4	Referências com Informações Complementares	109
3	MÉTODOS DE OTIMIZAÇÃO	110
3.1	Métodos Metaheurísticos de Otimização Mono-Objetivo	110
3.1.1	<u>Evolução Diferencial (ED)</u>	111
3.1.1.1	Mutação Diferencial	111
3.1.1.2	Cruzamento Diferencial	112
3.1.1.3	Seleção Diferencial	113
3.1.1.4	Algoritmo Geral da Evolução Diferencial	114
3.1.1.5	Aplicação da Evolução Diferencial em Problemas de Otimização Restrita	116
3.1.2	<u>Algoritmo Genético (AG)</u>	116
3.1.2.1	Características do Algoritmo Genético para Otimização	116
3.1.2.2	Aplicação de Algoritmos Genéticos em Problemas de Otimização Restrita	116
3.2	Métodos Metaheurísticos de Otimização Multiobjetivo	117
3.2.1	<u>Non-dominated Sorting Genetic Algorithm II (NSGA-II)</u>	118
3.2.1.1	Ordenação em Frentes Não Dominadas	118
3.2.1.2	Diversidade de Soluções	119
3.2.1.3	Distância de Aglomeração	120
3.2.1.4	Comparação de Aglomeração	121
3.2.1.5	Algoritmo Principal do NSGA-II	123
3.2.1.6	Aplicação do NSGA-II a Problemas de Otimização Restrita	125
3.2.2	<u>Adaptive Evol. Algorithm Based on Non Euclidean Geometry (AGE-MOEA)</u>	127
3.2.2.1	Ordenação em Frentes Não Dominadas	127
3.2.2.2	Normalização de Objetivos	127
3.2.2.3	Estimção da Geometria da Frente de Pareto	128
3.2.2.4	Proximidade e Diversidade de Soluções	130
3.2.2.5	Pontuação de Sobrevivência	131
3.2.2.6	Algoritmo Principal do AGE-MOEA	133
3.2.2.7	Emprego do AGE-MOEA em Problemas de Otimização Restrita	135
3.3	Referências com Informações Complementares	135

4	FERMENTAÇÃO ALCOÓLICA	138
4.1	Ascensão e Consolidação da Indústria Alcooleira	138
4.2	Etapas do Processo de Produção Alcoólica	139
4.2.1	<u>Lavagem</u>	140
4.2.2	<u>Moagem</u>	140
4.2.3	<u>Purificação</u>	140
4.2.4	<u>Fermentação</u>	140
4.2.5	<u>Destilação</u>	141
4.2.6	<u>Desidratação</u>	141
4.2.7	<u>Armazenagem</u>	141
4.3	Modelos das Plantas Sucroalcooleiras do Brasil	142
4.4	Tipos de Processos Fermentativos	143
4.4.1	<u>Fermentação em Batelada</u>	143
4.4.1.1	Modelo Prático de Fermentação em Batelada Alimentada	143
4.4.1.2	Vantagens da Fermentação em Batelada	144
4.4.1.3	Problemáticas da Fermentação em Batelada	145
4.4.2	<u>Fermentação em Batelada Alimentada</u>	145
4.4.2.1	Processo Industrial de Fermentação em Batelada Alimentada	146
4.4.2.2	Vantagens e Desvantagens da Fermentação em Batelada Alimentada	147
4.4.3	<u>Fermentação Contínua</u>	148
4.4.3.1	Configurações de Arranjo de Reatores na Fermentação Contínua	148
4.4.3.2	Processo Industrial de Fermentação Contínua	149
4.4.3.3	Vantagens e Problemáticas da Fermentação Contínua	149
4.4.4	<u>Contraposição Entre os Tipos de Processos Fermentativos</u>	150
4.5	Bioquímica do Processo Fermentativo	152
4.5.1	<u>Conceitos Fundamentais</u>	152
4.5.2	<u>Processo Bioquímico</u>	153
4.5.3	<u>Fases do Processo Fermentativo</u>	154
4.5.4	<u>Fases do Crescimento Microbiano no Processo Fermentativo</u>	154
4.6	Modelagem Matemática do Processo Fermentativo	155
4.6.1	<u>Modelos Cinéticos do Processo Fermentativo</u>	156
4.6.1.1	Tipos de Modelos Cinéticos	156
4.6.1.2	Complexidade e Limitações dos Modelos	157
4.6.1.3	A Equação de Monod	157
4.6.1.4	Modelos Baseados na Cinética de Michaelis-Menten	158
4.6.1.5	Importância da Modelagem e Simulação	160
4.6.2	<u>Modelo Elementar do Processo de Fermentação</u>	160
4.7	Referências com Informações Complementares	161
5	FERMENTAÇÃO DE LISINA	163

5.1	Grupos R e os L-aminoácidos	163
5.2	A Importância da Indústria de L-lisina	164
5.3	Processo Produtivo da L-lisina	165
5.4	Etapas do Processo Fermentativo	166
5.4.1	<u>Preparação</u>	166
5.4.2	<u>Fermentação</u>	166
5.4.2.1	Fermentação em Batelada	167
5.4.2.2	Fermentação em Batelada Alimentada	167
5.4.3	<u>Ultrafiltração</u>	167
5.4.4	<u>Cromatografia</u>	167
5.4.5	<u>Concentração</u>	167
5.4.6	<u>Acidificação</u>	168
5.4.7	<u>Cristalização</u>	168
5.4.8	<u>Secagem</u>	168
5.4.9	<u>Embalagem e Armazenagem</u>	168
5.5	Processo Fermentativo	168
5.5.1	<u>Processo Industrial de Produção de Lisina</u>	168
5.6	Biossíntese da L-lisina	169
5.6.1	<u>Processo Bioquímico</u>	170
5.6.2	<u>Rendimento Teórico Máximo</u>	171
5.7	Modelagem Matemática do Processo Fermentativo	172
5.8	Referências com Informações Complementares	173
6	ALGORITMO DESENVOLVIDO	175
6.1	Algoritmo	175
6.1.1	<u>Algoritmo de Valoração Funcional</u>	175
6.1.2	<u>Solucionador de Equações Diferenciais Ordinárias</u>	176
6.1.3	<u>Interpolador de Soluções</u>	177
6.1.4	<u>Algoritmo Principal do CODEMSOA</u>	178
6.2	Modelo Computacional	180
6.2.1	<u>Estruturas Utilitárias</u>	181
6.2.2	<u>Estruturas Genéricas</u>	182
6.2.3	<u>Modelo do Solucionador de Equações Diferenciais Ordinárias</u>	182
6.2.4	<u>Modelo do Interpolador de Soluções</u>	183
6.2.5	<u>Modelo do Otimizador</u>	185
6.2.6	<u>Modelo Principal</u>	188
6.3	Detalhes de Implementação	188
7	FORMULAÇÃO DOS PROCESSOS E PROBLEMAS	190
7.1	Estudo de Caso 1: Fermentação de Lisina	190
7.2	Processo de Fermentação	190

7.2.1	<u>Modelo Cinético do Reator</u>	190
7.2.2	<u>Especificações e Parâmetros de Simulação</u>	192
7.3	Processo de Otimização	194
7.3.1	<u>Produtividade e Rendimento</u>	195
7.3.1.1	Produtividade	195
7.3.1.2	Rendimento	195
7.3.2	<u>Problema de Otimização A</u>	196
7.3.3	<u>Problema de Otimização B</u>	197
7.3.4	<u>Inconsistências Identificadas nas Formulações Originais</u>	197
7.4	Estudo de Caso 2: Fermentação Alcoólica	199
7.5	Processo de Fermentação	199
7.5.1	<u>Modelo Cinético do Reator</u>	200
7.5.2	<u>Especificações e Parâmetros de Simulação</u>	202
7.6	Processo de Otimização	206
7.6.1	<u>Problema de Otimização A</u>	206
7.6.2	<u>Problema de Otimização B</u>	208
8	SIMULAÇÕES, RESULTADOS, ANÁLISE E DISCUSSÃO	210
8.1	Ambiente	210
8.2	Simulações e Resultados do Estudo de Caso 1	210
8.2.1	<u>Problema de Otimização A</u>	211
8.2.1.1	Simulação com o NSGA-II	212
8.2.1.1.1	Fronteira de Pareto	212
8.2.1.1.2	Conjunto Ótimo de Pareto	212
8.2.1.1.3	Relação entre os Otimizadores de Pareto e os Objetivos	214
8.2.1.2	Simulação com o AGE-MOEA	219
8.2.1.2.1	Fronteira de Pareto	219
8.2.1.2.2	Conjunto Ótimo de Pareto	219
8.2.1.2.3	Relação entre os Otimizadores de Pareto e os Objetivos	221
8.2.1.3	Simulação com o C-TAEA	226
8.2.1.3.1	Fronteira de Pareto	226
8.2.1.3.2	Conjunto Ótimo de Pareto	226
8.2.1.3.3	Relação entre os Otimizadores de Pareto e os Objetivos	228
8.2.1.4	Soluções Notáveis	233
8.2.2	<u>Problema de Otimização B</u>	233
8.2.2.1	Simulação com o NSGA-II	233
8.2.2.1.1	Fronteira de Pareto	234
8.2.2.1.2	Conjunto Ótimo de Pareto	234
8.2.2.1.3	Relação entre os Otimizadores de Pareto e os Objetivos	234
8.2.2.2	Simulação com o AGE-MOEA	240

8.2.2.2.1	Fronteira de Pareto	240
8.2.2.2.2	Conjunto Ótimo de Pareto	240
8.2.2.2.3	Relação entre os Otimizadores de Pareto e os Objetivos	242
8.2.2.3	Simulação com o C-TAEA	246
8.2.2.3.1	Fronteira de Pareto	246
8.2.2.3.2	Conjunto Ótimo de Pareto	246
8.2.2.3.3	Relação entre os Otimizadores de Pareto e os Objetivos	248
8.2.2.4	Soluções Notáveis	252
8.3	Análise e Discussão do Estudo de Caso 1	252
8.3.1	<u>Problema de Otimização A</u>	252
8.3.1.1	Considerações sobre os Métodos de Otimização	253
8.3.1.2	Considerações sobre os Resultados	253
8.3.1.3	Análise de Cenários	254
8.3.1.3.1	Cenário 1	254
8.3.1.3.2	Cenário 2	255
8.3.1.3.3	Cenário 3	257
8.3.2	<u>Problema de Otimização B</u>	258
8.3.2.1	Considerações sobre os Métodos de Otimização	258
8.3.2.2	Considerações sobre os Resultados	258
8.3.2.3	Análise de Cenários	259
8.3.2.3.1	Cenário 1	259
8.3.2.3.2	Cenário 2	260
8.3.2.3.3	Cenário 3	262
8.3.3	<u>Comparação dos Resultados Obtidos com os Resultados Originais</u>	263
8.4	Simulações e Resultados do Estudo de Caso 2	263
8.4.1	<u>Processo Não Otimizado</u>	264
8.4.1.1	Simulação com o RK-2	264
8.4.1.2	Concentrações Obtidas ao Final do Processo	264
8.4.2	<u>Problema de Otimização A</u>	266
8.4.2.1	Simulação com o NSGA-II	266
8.4.2.1.1	Conjunto Ótimo de Pareto	266
8.4.2.1.2	Fronteira de Pareto	267
8.4.2.1.3	Relação entre os Otimizadores de Pareto e os Objetivos	267
8.4.2.2	Simulação com o AGE-MOEA	271
8.4.2.2.1	Conjunto Ótimo de Pareto	271
8.4.2.2.2	Fronteira de Pareto	271
8.4.2.2.3	Relação entre os Otimizadores de Pareto e os Objetivos	273
8.4.2.3	Simulação com o C-TAEA	276
8.4.2.3.1	Conjunto Ótimo de Pareto	276

8.4.2.3.2	Fronteira de Pareto	276
8.4.2.3.3	Relação entre os Otimizadores de Pareto e os Objetivos	278
8.4.2.4	Soluções Notáveis	281
8.4.3	<u>Problema de Otimização B</u>	282
8.4.3.1	Simulação com o NSGA-II	282
8.4.3.1.1	Fronteira de Pareto	282
8.4.3.1.2	Conjunto Ótimo de Pareto	282
8.4.3.1.3	Relação entre os Otimizadores de Pareto e os Objetivos	282
8.4.3.2	Simulação com o AGE-MOEA	288
8.4.3.2.1	Fronteira de Pareto	288
8.4.3.2.2	Conjunto Ótimo de Pareto	288
8.4.3.2.3	Relação entre os Otimizadores de Pareto e os Objetivos	288
8.4.3.3	Simulação com o C-TAEA	294
8.4.3.3.1	Fronteira de Pareto	294
8.4.3.3.2	Conjunto Ótimo de Pareto	294
8.4.3.3.3	Relação entre os Otimizadores de Pareto e os Objetivos	294
8.4.3.4	Soluções Notáveis	300
8.5	Análise e Discussão do Estudo de Caso 2	300
8.5.1	<u>Problema de Otimização A</u>	301
8.5.1.1	Considerações sobre os Métodos de Otimização	301
8.5.1.2	Considerações sobre os Resultados	301
8.5.1.3	Análise de Cenários	303
8.5.1.3.1	Cenário 1	303
8.5.1.3.2	Cenário 2	305
8.5.2	<u>Problema de Otimização B</u>	307
8.5.2.1	Considerações sobre os Métodos de Otimização	307
8.5.2.2	Considerações sobre os Resultados	308
8.5.2.3	Análise de Cenários	309
8.5.2.3.1	Cenário 1	309
8.5.2.3.2	Cenário 2	311
	CONCLUSÃO	314
	REFERÊNCIAS	320
	APÊNDICE A – Métodos Disponíveis na Biblioteca de <i>Software</i> desenvolvida	333
	APÊNDICE B – Linguagem e Bibliotecas Usadas pela Ferramenta Computacional Desenvolvida	336
	APÊNDICE C – Exemplo de Chamada ao CODEMSOA na Biblioteca de Código Desenvolvida	338

APÊNDICE D – Configurações do <i>Hardware</i> onde as Simulações foram Realizadas	340
---	-----

INTRODUÇÃO

Posso até morrer sem realizar nenhum sonho, mas lutarei até o último dia da minha vida por cada um deles.

Liliane Mendes

Inúmeras áreas do conhecimento, tais como física, química, biologia, economia e medicina, têm parte de seus problemas passíveis de serem modelados por equações diferenciais ordinárias. Este tipo de problema apresenta relevante interesse por parte da comunidade científica e do mercado.

Segundo enunciado por Roberts (2010), as equações diferenciais ordinárias possuem importantes campos de aplicação. Problemas que envolvem crescimento populacional, movimento, queda, ligações atômicas e moleculares, mistura de substâncias, resfriamento e aquecimento, circuitos elétricos, desenvolvimento de microrganismos, orbitação de satélites e decaimento radioativo podem ser descritos por equações diferenciais ordinárias.

Dessa forma, existe um grande interesse nos problemas de equações diferenciais ordinárias e em seus métodos de solução em virtude do seu amplo espectro de aplicabilidade prática e teórica (CHICONE, 2006).

Uma outra grande área de interesse da comunidade acadêmica e do mercado é a otimização, que tem o intuito de encontrar as condições mais favoráveis para o desenvolvimento de um processo ou produto; ou encontrar o melhor valor dentre um conjunto de alternativas disponíveis.

Assim, atualmente, a otimização se tornou um procedimento fundamental aplicado no desenvolvimento de métodos, produtos, serviços e processos, fazendo com que o processo fique mais enxuto, gere melhores resultados, consuma menos tempo e utilize menos recursos, comumente objetivando reduzir custos e ampliar desempenho (DIEHL *et al.*, 2014).

Diante disto, os métodos e estudos da área de otimização têm ganhado particular destaque nos mais variados setores produtivos do mercado, recebendo especial atenção, os estudos voltados a atividades que apresentam alto potencial econômico (PEREIRA *et al.*, 2021).

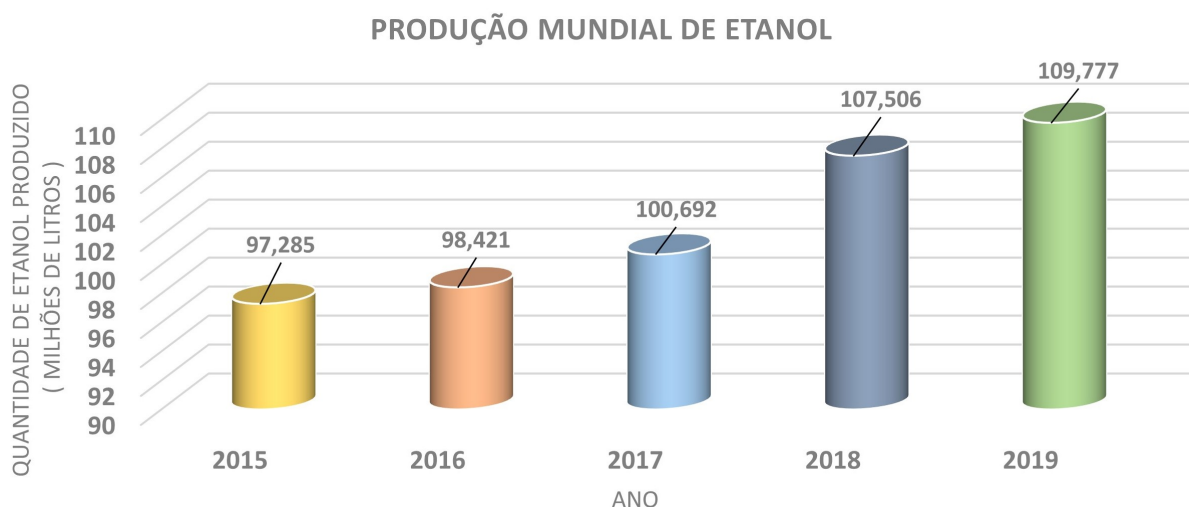
Neste contexto, existem inúmeros processos que são modelados por equações diferenciais ordinárias nos quais há um grande interesse em otimizá-los, a fim de que tornem-se mais baratos, produtivos e eficientes.

Uma área bastante promissora que apresenta processos que podem ser descritos por equações diferenciais ordinárias e, devido ao seu alto potencial econômico e industrial, desperta particular interesse na otimização de seus processos é a área de fermentação alcooleira (LIMA; MARCONDES, 2002).

Para mensurar a relevância desta área, pode-se mencionar que a produção de etanol no mundo em 2019 totalizou, segundo a RFA (2022) cerca de 29 milhões de galões, o que representa aproximadamente 109,8 milhões de litros. Em 2015, a produção mundial de etanol, também segundo a RFA (2022), foi de 25,7 milhões galões ou 97,3 milhões de litros.

Logo, podemos perceber que em cinco anos, a produção mundial aumentou em aproximadamente 12,95%. Esse aumento na produção reflete a elevação da demanda por etanol que vem ocorrendo ao longo dos anos, conforme mostra o Gráfico 1, que apresenta a produção mundial de etanol estimada por ano, de acordo com a RFA (2022).

Gráfico 1 - Produção mundial de etanol.



Fonte: O autor, 2023.

Ademais, o etanol está presente em uma infinidade de produtos que estão disponíveis no mercado. Conforme Nova Cana (2021), na sua forma pura, denominada álcool anidro, ele é empregado na produção de vernizes, tintas, solventes, perfumes, éter, ácido acético, acetaldeído e diversos outros produtos.

O etanol com aproximadamente 4% de água, chamado de álcool hidratado, é usado na produção de bebidas, alimentos, cosméticos, aromatizantes, produtos de limpeza, medicamentos, vacinas e em vários outros tipos de produto.

Cabe ainda dizer que quando se trata de combustível, especialmente o veicular, o etanol pode ser usado integralmente como combustível, em sua versão hidratada, ou misturado com outros combustíveis como gasolina e diesel, em sua versão pura.

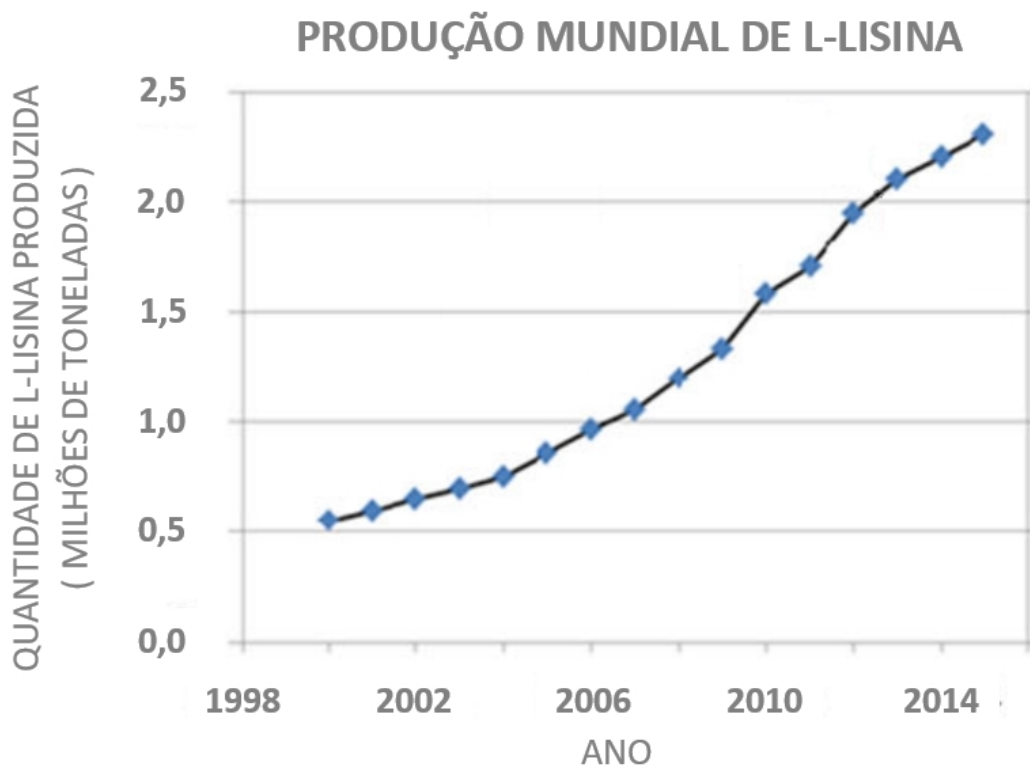
Embora o etanol não seja o combustível mais considerado para ser aplicado em larga escala na aviação, até mesmo ela já o empregou como combustível. Esse marco foi atingido em 2005, no lançamento do Ipanema EMB pela Embraer, a primeira aeronave a utilizar etanol como combustível (NOVA CANA, 2021).

Outra área de bastante relevância, que também possui processos que podem ser descritos por equações diferenciais ordinárias e apresenta grande importância no mercado econômico, fato que provoca forte interesse na otimização de seus processos, é a fermentação e produção de L-lisina, um aminoácido essencial que é fundamental para o ser humano e alguns animais.

Para ponderar a importância desta área, podemos aludir que, segundo Eggeling e Bott (2015), em 2015, a produção mundial estimada de L-lisina foi de 2.200.000 toneladas, tendo o Brasil uma posição de destaque mundial, atrás dos Estados Unidos, China e Indonésia. No ano de 2000, segundo Kircher e Pfefferle (2001), a produção mundial de L-lisina foi estimada em 400.000 toneladas.

Logo, nota-se que do período de 2000 a 2015, houve um crescimento de 450% na produção mundial de L-lisina. Essa expressiva elevação na produção é a resposta ao grande aumento da demanda pela L-lisina, que vem se intensificando com o passar dos anos, conforme ilustra o Gráfico 2, que mostra a produção mundial de L-lisina por ano, segundo Eggeling e Bott (2015).

Gráfico 2 - Produção mundial de L-lisina.



Fonte: Adaptado de Eggeling e Bott, 2015.

A principal aplicação da L-lisina, do ponto de vista industrial, é na suplementação da alimentação animal, especialmente para a criação de bovinos, suínos e aves, tendo também a possibilidade de seu emprego na piscicultura, como na criação de tilápias.

Além da aplicação principal, a L-lisina é também usada na suplementação alimentar humana. Embora comumente um ser humano obtenha quantidade suficiente de L-lisina através da alimentação, existem casos em que a suplementação se faz necessária (HAND, 2015).

A L-lisina também tem aplicação na indústria farmacêutica, estando presente como princípio ativo de drogas medicamentosas na forma de cloridrato de lisina, que é usada para prevenção da Herpes recorrente em adultos, e na forma de clonixinato de lisina, em virtude de seu efeito analgésico.

Motivação

Os processos de fermentação possuem parâmetros que influem no custo, na eficiência, nos riscos e nos resultados do processo, sendo desejável que os custos e riscos sejam reduzidos e a eficiência e resultados melhorados. Assim, quanto mais barato e melhor for o processo, maior a quantidade de produto gerada por um menor preço.

Um menor custo de produção provoca não somente o barateamento do produto em si, mas também de todos os derivados que possuem em sua composição uma quantidade relevante do produto em questão.

Essa eficiência na produção garante maior acesso e competitividade do produto e de seus derivados no mercado, mitigando, inclusive, prejuízos em eventuais quedas na demanda, como no caso do etanol, que ocorreu em 2020, onde o álcool brasileiro foi negociado abaixo do custo de produção (CASTRO, 2021).

Deste modo, otimizar o processo fermentativo, com intuito de reduzir consideravelmente os gastos e insumos envolvidos na fermentação; e ampliar significativamente a produtividade, é de grande interesse do mercado e conseqüentemente da academia.

Cabe ressaltar também que, diante deste grande interesse na área de fermentação, uma ampla gama de trabalhos que envolvem simulações, experimentos, comparações entre processos, revisões e diversas outras atividades voltadas para a área são desenvolvidos e publicados continuamente em periódicos, dos quais podemos citar Darmayanti *et al.* (2023), Villa *et al.* (2023), Thakur *et al.* (2023), Sharma *et al.* (2023), Wang e Tepfenhart (2019) e Ta *et al.* (2023).

No entanto, em determinadas publicações, a simulação promovida pelo estudo que originou o artigo tem os valores de seus parâmetros pautados em estudos anteriores ou valores experimentais. Por esta razão, não existem quaisquer garantias de que os resultados obtidos ao término da simulação são os mais otimizados possíveis.

Um cenário diferente ocorreria se o valor de tais parâmetros fosse oriundo da otimização do processo de fermentação. A otimização traria certa garantia de que os valores de parâmetros obtidos forneceriam ao processo de fermentação simulado, os melhores e

mais eficientes resultados possíveis.

Diante disto, diversos trabalhos foram desenvolvidos no sentido de otimizar processos de fermentação, envolvendo os mais variados modelos e parâmetros de simulação do processo fermentativo, dos quais pode-se mencionar Darkwah, Knutson e Seay (2018), Haringa *et al.* (2018), Gujarathi, Patel e Siyabi (2023), Liu *et al.* (2022) e Santharam *et al.* (2017).

Entretanto, uma problemática observada nestes trabalhos é que cada um possui uma metodologia distinta para o seu desenvolvimento, que envolve o uso de diferentes ferramentas e tecnologias computacionais trabalhando em conjunto. Este uso conjunto envolve a construção de algoritmos para realizar a integração entre as diversas ferramentas empregadas no processo.

A construção de toda esta estrutura apresenta complexidade considerável de implementação, uma vez que envolve diferentes linguagens de programação e tecnologias que terão de trabalhar conjuntamente, além de que, por vezes, as velocidades de operação das ferramentas podem diferir, gerando um gargalo de tempo na estrutura.

Outro problema reside na integração das ferramentas, que envolve diversas operações de entrada e saída de dados, além dos processos de tradução dos formatos de dados entre ferramentas. Tudo isto incorre em um grande custo de tempo para que todos estes procedimentos ocorram, impactando consideravelmente o tempo computacional de otimização.

Diante dessa problemática, um algoritmo implementado na forma de uma biblioteca ou *framework*, que fornecesse o arcabouço necessário para que problemas de otimização envolvendo sistemas de equações diferenciais ordinárias fossem resolvidos e simulados de forma unificada em uma mesma plataforma, sem envolver ferramentas externas e processos de integração, reduziria consideravelmente a complexidade de implementação e também o tempo de otimização do processo, além de permitir que modelos distintos sejam otimizados e simulados em uma mesma ferramenta.

Concepção

Esta seção apresenta o processo de formação presente tese, enunciando cronologicamente os fatores que nortearam o desenvolvimento desta, desde sua concepção inicial até sua forma final.

Relevantes trabalhos para o meio científico, que são bem consolidados e que apresentam considerável número de citações, como Sarkar e Modak (2005), Al-Siyabi, Gujarathi e Sivakumar (2017), Lobato (2008) e Ani, Gujarathi e Vakili-Nezhaad (2021) são voltados a estudos de caso envolvendo a aplicação de algoritmos de otimização à resolução de problemas dinâmicos inerentes ao processo fermentativo de reatores de fermentação em

batelada alimentada.

Tais produções, ainda que com propósitos distintos, acabam por buscar, como objetivo ou metodologia, maximizar a produtividade e o rendimento de reatores de fermentação em batelada alimentada através da otimização envolvendo diversos modelos cinéticos matemáticos que descrevem o processo fermentativo que ocorre dentro do reator.

Neste sentido, a presente tese foi concebida nesta linha, a fim de contribuir de forma relevante e original, tendo inicialmente apenas o intuito de otimizar o processo de fermentação em batelada alimentada de uma planta de produção alcooleira brasileira real, que foi descrito e relatado por Fonseca, Costa e Cruz (2017), que até o presente momento, acredita-se não ter sido objeto deste tipo estudo.

Dessa forma, a princípio, a relevância, ineditismo e originalidade do presente trabalho consistiriam apenas na análise e estudo do caso de otimização do mencionado processo de fermentação etanólica.

Para realizar o estudo de caso em questão, durante o seu desenvolvimento, foi sendo construído um algoritmo cujo o objetivo era resolver problemas de otimização envolvendo o processo fermentativo considerado.

Entretanto, no decorrer do desenvolvimento deste algoritmo e de sua implementação computacional, percebeu-se o potencial de seu emprego na otimização de outros processos envolvendo equações diferenciais ordinárias, inclusive processos inerentes a outras áreas que não sejam a de fermentação.

Como, até onde se sabe, não foi desenvolvido e publicado um algoritmo e uma ferramenta computacional que forneça as estruturas requeridas para que problemas de otimização envolvendo equações diferenciais ordinárias sejam resolvidos e simulados de forma unificada em uma mesma plataforma, esta pode ser considerada também uma contribuição relevante, original e inédita da presente tese, que se tornou o principal foco deste trabalho.

Finalmente, para atestar a flexibilidade do algoritmo desenvolvido e implementado, elencou-se um dos estudos de caso que foi considerado e resolvido por Ani, Gujarathi e Vakili-Nezhaad (2021) a fim de que seus problemas fossem resolvidos pela ferramenta construída e os resultados obtidos por ela fossem contrastados com os obtidos pelo trabalho original, além de analisar e discutir os referidos resultados.

Objetivos

Diante do apresentado até aqui, o principal propósito desta tese consistiu no desenvolvimento de um algoritmo que permite a modelagem e solução de problemas de otimização envolvendo equações diferenciais ordinárias, possibilitando o emprego de diversos métodos de otimização mono-objetivo e multiobjetivo; e de solução de equações

diferenciais ordinárias.

Além do algoritmo propriamente dito, este trabalho teve o intuito de desenvolver uma biblioteca de código que implementa o algoritmo proposto, dando suporte a variados métodos de otimização, interpolação e para equações diferenciais ordinárias.

Esta tese também teve por objetivo, otimizar o processo de fermentação de L-lisina considerado no segundo estudo de caso feito por Ani, Gujarathi e Vakili-Nezhaad (2021), sendo decomposto nos objetivos específicos descritos abaixo.

Para simular o computacionalmente o processo de fermentação de L-lisina considerado, o presente trabalho teve o propósito de implementar computacionalmente o modelo referente ao processo de fermentação em questão.

Com o intuito de otimizar o processo implementado, esta produção considerou os problemas de otimização propostos no segundo estudo de caso realizado por Ani, Gujarathi e Vakili-Nezhaad (2021), se propondo a implementá-los e empregar métodos de otimização para resolvê-los.

Depois de obtidos os resultados, esta tese teve o intuito de realizar um breve contraponto entre os resultados obtidos aqui com os obtidos por Ani, Gujarathi e Vakili-Nezhaad (2021), além de discuti-los e analisá-los.

Por final, esta produção acadêmica teve por meta, a otimização do processo de fermentação em batelada alimentada da planta de produção etanólica relatada por Fonseca, Costa e Cruz (2017), sendo desmembrada nos objetivos específicos descritos a seguir.

Com intuito de simular o processo fermentativo, este trabalho objetivou inicialmente, implementar computacionalmente o modelo pertinente ao processo de fermentação em questão.

Para a proposição dos problemas de otimização sobre o processo implementado, esta tese teve como objetivo, realizar uma avaliação preliminar a fim de determinar quais variáveis, objetivos e restrições são relevantes para o processo de otimização.

Afim de determinar casos relevantes de otimização, a presente produção teve o intuito de propor e modelar problemas de otimização sobre o processo de fermentação, que sejam relevantes para a comunidade.

Para de fato otimizar o processo em questão, este trabalho se propôs a implementar computacionalmente os problemas de otimização propostos e empregar métodos de otimização para solucioná-los.

Após a obtenção dos resultados, a presente tese teve o propósito de analisá-los e expressar as competentes conclusões aferidas sobre eles.

Metodologia

Com o propósito de alcançar os objetivos especificados, foi adotada para o integral desenvolvimento da presente tese e seus artefatos, a metodologia especificada e descrita a seguir.

Para implementar o algoritmo proposto para resolução de problemas de otimização envolvendo equações diferenciais ordinárias foi realizada através da construção de uma biblioteca de código na linguagem de programação Python, cuja escolha se deu pelo fato desta possuir diversos recursos e bibliotecas que fornecem um considerável suporte ao desenvolvimento de projetos científicos, fato este que a faz dela a linguagem mais usada na computação científica (SRIVASTAVA, 2021).

Com relação aos paradigmas de programação adotados no desenvolvimento da biblioteca computacional descrita acima, optou-se pelo uso de uma abordagem multi-paradigma envolvendo o paradigma orientado a objetos, por conferir ao código, organização, manutenibilidade, modularização e encapsulamento, e o paradigma funcional, por atribuir flexibilidade, alta separação de responsabilidades, manipulação de funções como dados e notação concisa ao código (THANOSHAN, 2021).

No que tange a implementação e simulação sistemática dos processos fermentativos, fez-se uso de métodos numéricos para equações diferenciais ordinárias para solucionar os modelos. A solução numérica foi adotada em virtude de se estabelecer uma metodologia capaz de resolver uma gama maior de problemas que a solucionável por métodos analíticos.

Devido à natureza discreta da solução numérica, não é possível encontrar a solução em qualquer ponto dentro do intervalo estabelecido. Com objetivo de permitir que a solução possa ser estimada em qualquer ponto do intervalo, foi aplicada sistematicamente uma interpolação sobre a solução discreta. A obtenção da solução contínua oriunda do processo de interpolação visa garantir que o algoritmo de otimização consiga acessá-la em qualquer ponto, evitando que este incorra no erro de acessar um ponto de solução não definido.

No tocante a otimização, os problemas foram modelados como problemas de otimização multiobjetivo e resolvidos através da aplicação de métodos de otimização multiobjetivo, de forma a obter um conjunto de soluções ótimas que possam ser avaliadas e selecionadas conforme o cenário em questão.

Estrutura da Tese

Esta tese se inicia por seu Capítulo Introdutório, que apresenta e descreve o contexto no qual ela está inserida; e mostra as motivações, objetivos e metodologia relativos a este trabalho.

A seguir, em decorrência da necessidade de se conhecer os temas e assuntos envolvidos nesta tese, inclusive os citados neste capítulo, são apresentados os Capítulos de 1 a 5 que versam sobre tais assuntos, de maneira a fornecer a fundamentação teórica e revisão literária da presente tese.

Subsequentemente, são enunciados os Capítulos 6, 7 e 8, que apresentam as produções algorítmicas e computacionais deste trabalho, a formulação dos processos e problemas considerados e propostos nesta tese, além da descrição das simulações realizadas e dos resultados obtidos, seguidos das análises e discussões acerca destes.

Esta tese é encerrada pelo Capítulo Conclusivo, que compila os elementos que justificaram a realização do presente trabalho e apresenta uma síntese de seus resultados, suas contribuições, produções e perspectivas futuras.

Após a conclusão, são apresentados os Apêndices A, B, C e D, que trazem informações, dados e outros detalhes inerentes às produções geradas neste trabalho.

Considerando o ordenamento apresentado acima, esta tese foi estruturada e dividida em 8 capítulos e 4 apêndices, dispostos e descritos conforme especificado abaixo:

- Na Introdução, foram feitas as considerações iniciais, apresentada a motivação que levou ao desenvolvimento da presente tese, enunciados os objetivos do trabalho, descrita a metodologia empregada e especificada a estrutura da produção.
- No Capítulo 1, discorre-se sobre os conceitos e fundamentos da otimização, bem como, aborda-se a otimização mono-objetivo, que trata dos problemas que envolvem apenas um objetivo a ser otimizado, e a otimização multiobjetivo, que abarca os problemas que envolvem múltiplos objetivos, uma vez que os problemas propostos nesta tese são dessa natureza, além de serem suportados pela biblioteca de código desenvolvida.
- No Capítulo 2, são descritos os algoritmos evolucionários elementares e suas características, precedidos pela apresentação da metodologia de modelagem computacional dos conceitos oriundos das teorias evolutivas e dos principais operadores evolutivos de reprodução, que constituem as estruturas e processos elementares empregados pelos algoritmos evolucionários, uma vez que as metaheurísticas empregadas nesta tese consistem neste tipo de algoritmo.
- No Capítulo 3, é introduzido o conceito de métodos de otimização e são elencados técnicas de otimização metaheurísticas voltadas tanto à resolução de problemas de otimização mono-objetivo quanto a problemas de otimização multibjetivo. Este tipo de método será utilizado para solucionar os problemas de otimização propostos neste trabalho, além de estarem disponíveis na biblioteca de *software* desenvolvida.
- No Capítulo 4, discorre-se acerca da fermentação alcoólica, sua relevância, características, além dos tipos de processo fermentativo, bioquímica e modelagem matemá-

tica da fermentação, pois um dos processos considerados nesta tese pertence a área de fermentação etanólica.

- No Capítulo 5, é apresentado o processo fermentativo de L-lisina, seus conceitos, importância, características, processo de fermentação, bioquímica e modelagem matemática, pois um dos processos considerados nesta tese pertence a área de fermentação de L-lisina.
- No Capítulo 6, se encontra a proposição e especificação do algoritmo para otimização de sistemas envolvendo equações diferenciais ordinária desenvolvido nesta tese, bem como sua implementação computacional em uma biblioteca de *software*, acompanhada da descrição de sua arquitetura e detalhes de implementação.
- No Capítulo 7, está apresentado e definido o processo de fermentação considerado neste trabalho, com suas especificações e detalhes, bem como estão descritas e explicadas as formulações dos problemas de otimização, que estão sendo propostos por esta tese.
- No Capítulo 8, estão especificadas e descritas as simulações computacionais propostas e realizadas neste trabalho, apresentados, relacionados, analisados e discutidos os resultados obtidos a partir as simulações promovidas.
- Na Conclusão, foram feitas as considerações finais, elencadas as principais produções geradas e as contribuições fornecidas por esta tese, além de descrever as perspectivas futuras esperadas para este trabalho.
- No Apêndice A, são enunciados os métodos de otimização, interpolação e para equações diferenciais ordinárias, que estão disponíveis na biblioteca de *software* desenvolvida.
- No Apêndice B, estão mostradas as bibliotecas de código usadas pela biblioteca desenvolvida e a linguagem na qual esta foi desenvolvida, acompanhadas de suas versões e descrições.
- No Apêndice C, está exemplificada uma chamada ao algoritmo proposto a nível de código, na implementação dada pela biblioteca de *software* proposta.
- No Apêndice D, estão descritas as configurações de hardware das máquinas onde foram executadas as simulações computacionais realizadas neste trabalho.

Nota ao Leitor 1 *As letras e símbolos matemáticos usados nesta tese mantêm seu significado e definição apenas no capítulo em que os mesmos foram estabelecidos. Assim, letras e símbolos iguais em capítulos diferentes podem ter diferentes significados e definições. Portanto, uma letra ou símbolo será definido em um capítulo e sua semântica será*

válida apenas no capítulo em questão. A única exceção ocorre quando em um determinado capítulo é dito explicitamente que determinadas letras e símbolos têm seu significado herdado de outro capítulo.

Nota ao Leitor 2 *As proposições, lemas, teoremas e corolários enunciados nesta tese não serão demonstrados ou provados aqui, uma vez que este não é o cerne deste trabalho. As demonstrações poderão ser encontradas na bibliografia sugerida ao final do capítulo em que os mesmos foram enunciados.*

1 OTIMIZAÇÃO

Se pude enxergar mais longe, foi porque me apoiei nos ombros de gigantes.

Isaac Newton

Este capítulo abordará o conceito de otimização, bem como os seus fundamentos nos contextos mono e multi-objetivo.

Conhecer os temas relacionados a otimização é imprescindível para o pleno entendimento desta tese, uma vez que os problemas de otimização e seus conceitos constituem parte importante deste trabalho.

1.1 Princípios da Otimização

Quando desenvolvemos ou usamos um determinado processo, especialmente processos de engenharia, desejamos ampliar ou melhorar os produtos resultantes deste processo e reduzir ou mitigar os custos que este acarreta.

Todo processo possui variáveis e objetivos inerentes a ele. As variáveis representam os insumos e os objetivos, os produtos, podendo ambos serem tangíveis ou intangíveis. Para atender aos anseios citados anteriormente, deseja-se manipular as variáveis de forma a melhorar os resultados para um objetivo.

Estabelecer o valor das variáveis de um problema, de maneira a melhorar resultados para alguns de seus objetivos consiste, na verdade, em resolver o que chamamos de problema de otimização.

Antes de abordar o conceito de otimização, é interessante entender as diferenças entre um processo não otimizado e um otimizado e quais as vantagens de se trabalhar com este último.

1.1.1 Processo Não Otimizado

Um processo tradicional não otimizado se baseia na análise de diversas soluções conhecidas e também na viabilidade de implementá-lo.

Nesse tipo de processo, as decisões, aprimoramentos e soluções são feitas de forma subjetiva pelo engenheiro e por isso, essas decisões estarão ligadas à experiência, habilidade, conhecimento e intuição do profissional. Isso pode ser negativo, pois um profissional pode ter esses atributos em nível mais alto ou mais baixo, além de que, esses níveis podem

variar conforme o caso específico de um determinado projeto (GUIMARAES; SOUZA; JUNIOR, 2007).

Nessa abordagem, muitas especificidades do processo em questão não são consideradas, além de não existir uma forma sistemática para embasar a tomada de decisões, realização de aprimoramentos e solução de problemas no processo.

1.1.2 Processo Otimizado

Um processo otimizado, por outro lado, possui um método sistemático para embasar a tomada de decisões e para definir suas características.

Para otimizar um sistema real, primeiramente, deve-se identificar as variáveis deste. Em seguida, é preciso analisá-las e selecionar aquelas cuja variação de seus valores afetem significativamente o desempenho do problema.

Essa seleção de variáveis se faz necessária, pois quanto mais variáveis tem um problema de otimização, maior sua complexidade, sendo o aumento de complexidade exponencial devido ao aumento das combinações possíveis entre valores de variáveis. Assim, é importante adotar a menor quantidade possível de variáveis, escolhendo apenas as que sejam relevantes ao problema.

Depois, estabelece-se os objetivos. São eles que vão mensurar o mérito relativo de cada uma das possíveis soluções. Deve-se estabelecer também, caso existam, as restrições do projeto. Elas vão refletir as limitações e inviabilidades do processo.

Uma vez definidas essas grandezas, deve-se formular um modelo que represente matematicamente o processo real, relacionando as variáveis aos resultados obtidos no processo. Trabalhar com modelos, ainda que estes representem de forma simplificada o processo real em questão, traz viabilidade e eficiência para o processo de análise e otimização, uma vez que usar o sistema real para realizar testes e experimentos, pode ser demorado e caro, além de que, em certas aplicações, pode trazer consequências não toleráveis, caso algo inesperado ocorra. Fatores que tornam, por vezes, inviável a otimização sem o uso de modelos.

Com o modelo definido e o problema formulado, seleciona-se um método de otimização apropriado para o problema em questão e emprega-o para se chegar à solução ou às soluções ótimas dentro dos parâmetros estabelecidos.

Esse tipo de processo depende muito menos dos atributos do engenheiro que um processo não otimizado, suas soluções estão muito mais associadas a abordagem empregada pelo algoritmo de otimização.

1.1.3 Vantagens do Processo Otimizado

Trabalhar com um processo otimizado traz soluções mais eficientes e econômicas, além de se possibilitar que tais soluções sejam atingidas em um tempo relativamente curto se comparado ao gasto nas diversas tentativas que precisariam ser feitas pelo profissional em um processo tradicional. Além de que, mesmo que estas tentativas forem feitas, ainda assim, nada garante que uma solução ótima será alcançada por meio delas (BRASIL; SILVA, 2019).

1.1.4 Conceito de Otimização

Um problema de otimização consiste em determinar o valor das variáveis de decisão elencadas que levam aos melhores valores para os objetivos considerados.

De uma perspectiva simplista, otimizar significa encontrar a melhor maneira e meio de executar uma tarefa. Constantemente, estamos buscando otimizar as tarefas e processos do nosso cotidiano, tais como, gastar menos nas compras ou ser aprovado no máximo de disciplinas.

Brasil e Silva (2019) definem otimização como o processo de se determinar dentre várias opções, aquela que é a melhor possível dentro de certos critérios de escolha e limitações, com os recursos disponíveis.

Matematicamente, podemos dizer que a otimização é o estudo dos problemas que consistem na minimização ou maximização de um ou mais objetivos por meio da escolha sistemática dos valores das variáveis deste problema, respeitando um conjunto viável de valores.

1.1.5 Otimização Contínua

Problemas de otimização que envolvem variáveis que podem assumir valores de conjuntos infinitos não enumeráveis, tratando-se normalmente do conjunto dos reais, são denominados **Problemas de Otimização Contínua**.

Os problemas de otimização contínua geralmente possuem resolução mais simples, uma vez que comumente estes apresentam funções objetivo e de restrições suaves, o que permite inferir o comportamento destas funções em uma vizinhança de um ponto (ANDRETTA, 2010c).

Os algoritmos de solução de problemas de otimização contínua utilizam das propriedades supracitadas para encontrar soluções para os problemas.

1.1.6 Otimização Discreta

Existem problemas de otimização cujas variáveis só admitem valores inteiros, devido a própria natureza do problema. Esse tipo de problema é denominado **Problema de Otimização Discreta** ou **Problema de Programação Inteira**, uma vez que, as variáveis só podem assumir valores de um conjunto finito.

Comumente, para solucionar este tipo de problema, supõe-se que as variáveis possam assumir valores contínuos, emprega-se o algoritmo de solução, arredondando as soluções a fim de que estas apresentem valores inteiros.

Os problemas de programação inteira normalmente são mais complexos de se solucionar, pois duas soluções vizinhas podem levar a valores de objetivo muito distantes, inviabilizando seu tratamento como um problema de otimização contínua.

No caso supracitado, objetivando contornar o problema apresentado, pode-se considerar apenas as soluções possíveis e dentre elas buscar a que leva ao melhor valor de objetivo. No entanto, em problemas com muitas variáveis, essa busca pode ser inviável (ANDRETTA, 2010c).

1.2 Otimização Mono-Objetivo

A presente seção apresenta as características e conceitos dos problemas de otimização que envolvem um único objetivo. Esse tipo de problema será abarcado pelo algoritmo desenvolvido nesta tese e suportado pela biblioteca de *software* desenvolvida, além de muitos conceitos apresentados aqui serem válidos também para a otimização multiobjetivo.

Quando temos um problema de otimização que apresenta apenas um único objetivo a ser otimizado, o denominamos problema de Otimização Mono-Objetivo ou problema de Otimização Clássica. Formalmente, um problema de otimização mono-objetivo é expresso pela Definição 1.2.1.

Definição 1.2.1 (Problema de Otimização Mono-Objetivo) *Seja um conjunto $\Omega \subseteq \mathbb{R}^n$ e uma função $f : \Omega \rightarrow \mathbb{R}$, um problema de otimização mono-objetivo pode ser denotado pela Equação (1), onde o conjunto Ω é denominado **conjunto viável**, a função f é denominada **função objetivo** e x é um vetor de forma $x = (x_1, x_2, x_3, \dots, x_n)$ que configura o vetor de **variáveis** a serem manipuladas a fim de otimizar a função objetivo, sendo n o número total de variáveis.*

$$\begin{array}{ll} \underset{x}{\text{Minimizar}} & f(x) \\ \text{Sujeito a} & x \in \Omega \end{array} \tag{1}$$

1.2.1 Problema de Minimização e Problema Maximização

O problema de otimização definido na Equação (1) trata-se de um **Problema de Minimização**, uma vez que por simplicidade, um problema de otimização pode ser definido como um problema de minimização.

Um problema de otimização também pode ser um **Problema de Maximização**, onde se busca encontrar para quais valores das variáveis de x , a função f apresenta o seu maior valor, sendo denotado na forma expressa pela Equação (2).

$$\begin{aligned} \text{Maximizar} \quad & f(x) \\ \text{Sujeito a} \quad & x \in \Omega \end{aligned} \tag{2}$$

Quando consideramos a função objetivo de um problema de otimização, esta pode apresentar nomenclaturas alternativas, que mudam com relação ao tipo de problema que está sendo tratado.

A denominação **Função Custo** é dada à função objetivo de um problema de minimização, enquanto a nomenclatura **Função Utilidade** é atribuída à função objetivo de um problema de maximização.

1.2.1.1 Conversão entre Problemas de Minimização e Maximização

O problema de maximização mostrado na Equação (2) pode ser transformado em um problema de minimização equivalente através da inversão do sinal da função objetivo, conforme mostrado na Equação (3).

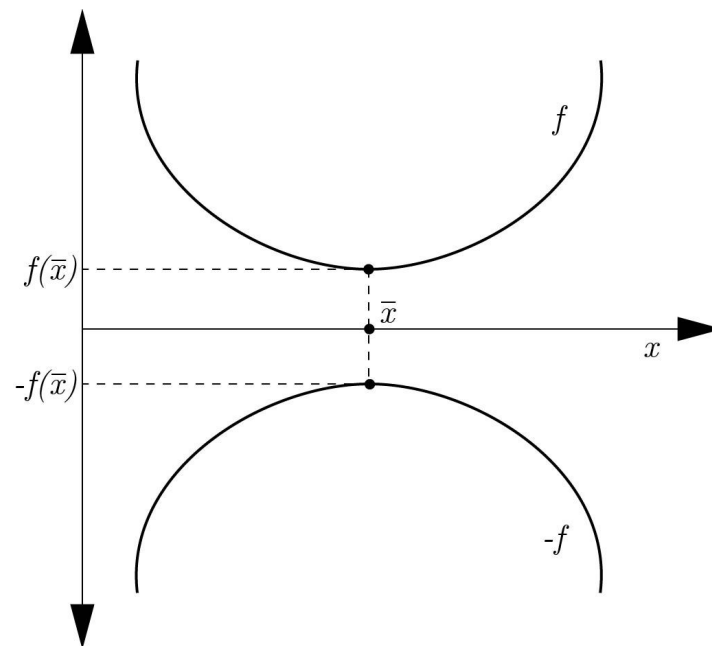
$$\begin{aligned} \text{Minimizar} \quad & -f(x) \\ \text{Sujeito a} \quad & x \in \Omega \end{aligned} \tag{3}$$

Essa transformação é possível porque as soluções de um problema de minimização são as mesmas de um problema de maximização, que tem a mesma função do problema de minimização com sinal trocado, como pode ser visualizado no exemplo ilustrado na Figura 1 (IZMAILOV; SOLODOV, 2014).

Por esta razão, matematicamente, não existe diferença relevante entre a abordagem de um problema de minimização e a de um problema de maximização.

Nota ao Leitor 3 *Devido a inexistência de diferenças relevantes nos conceitos e na solução de problemas de minimização e maximização, uma vez que um pode ser facilmente transformado no outro, neste trabalho, consideraremos apenas problemas de minimização.*

Figura 1 - Equivalência entre problemas de minimização e maximização.



Legenda: O vetor \bar{x} é tanto minimizador de f , quanto maximizador de $-f$.

Fonte: O autor, 2023.

1.2.2 Otimização Irrestrita

Em um problema de otimização irrestrita, as variáveis podem assumir qualquer valor dentro do \mathbb{R}^n com a finalidade de atingir o ponto ótimo da função objetivo. Esse fator faz com que este tipo de problema não seja muito comum na realidade, uma vez que na maioria das situações, existem limitações de natureza física, financeira ou mesmo de engenharia.

A Definição 1.2.2 expressa o conceito de problema de otimização irrestrito.

Definição 1.2.2 (Problema de Otimização Irrestrita) *Um problema de otimização é dito irrestrito quando $\Omega = \mathbb{R}^n$, ou seja, quando o conjunto viável é o \mathbb{R}^n .*

1.2.2.1 Representação de um Problema de Otimização Irrestrita

Este tipo de problema pode ser representado de forma particular, conforme mostra a Equação (4).

$$\underset{x}{\text{Minimizar}} \quad f(x) \tag{4}$$

A omissão do termo (*Sujeito a* $x \in \Omega$) deixa subentendido que o problema está *sujeito a* $x \in \mathbb{R}^n$.

Embora a notação dada pela Equação (4) possa ser adotada, nada impede a adoção da notação padrão mostrada pela Equação (5).

$$\begin{array}{ll} \underset{x}{\text{Minimizar}} & f(x) \\ \text{Sujeito a} & x \in \mathbb{R}^n \end{array} \quad (5)$$

1.2.3 Otimização Restrita

Os problemas de otimização restrita impõem restrições nos valores que as variáveis do problema podem assumir. Essas restrições refletem matematicamente as limitações do problema real no modelo matemático.

Por exemplo, variáveis que representem quantidades de elementos não podem assumir valores negativos, uma vez que isso é uma inconsistência física. Ainda que exista um ponto ótimo para valores negativos, não poderemos adotá-los, pois na realidade não haverá como reproduzir tais condições, tendo que nos satisfazer com o melhor ponto que atenda as restrições, ainda que exista um melhor quando estas não são consideradas.

O conceito de problema de otimização restrito é apresentado na Definição 1.2.3.

Definição 1.2.3 (Problema de Otimização Restrita) *Um problema de otimização é dito restrito quando $\Omega \subsetneq \mathbb{R}^n$, isto é, quando o conjunto viável é um subconjunto de \mathbb{R}^n , diferente de \mathbb{R}^n .*

1.2.3.1 Formato Tradicional do Conjunto Viável

Nos problemas de otimização restrita, usualmente, o conjunto viável de um problema de otimização é definido por um conjunto de igualdades, desigualdades e uma inclusão, tendo a forma mostrada na Equação (6), onde o conjunto $\Upsilon \subset \mathbb{R}^n$ é composto pelos elementos que x , a princípio, pode assumir, a função $h : \Omega \rightarrow \mathbb{R}^k$, cujas componentes têm a forma $h(x) = \langle h_1(x), h_2(x), h_3(x), \dots, h_k(x) \rangle$, constitui as **Restrições de Igualdade** e a função $g : \Omega \rightarrow \mathbb{R}^l$, cujas componentes são de forma $g(x) = \langle g_1(x), g_2(x), g_3(x), \dots, g_l(x) \rangle$, constitui as **Restrições de Desigualdade**.

$$\Omega = \{x \in \Upsilon \mid h(x) = (0, 0, 0, \dots, 0); g(x) \leq (0, 0, 0, \dots, 0)\} \quad (6)$$

Em uma notação mais expandida, podemos denotar um conjunto viável acima pela

Equação (7).

$$\Omega = \{x \in \Upsilon \mid h_a(x) = 0, a = 1, 2, 3, \dots, k; g_b(x) \leq 0, b = 1, 2, 3, \dots, l\} \quad (7)$$

Cabe frisar que podem existir restrições funcionais nas formas $A(x) = B(x)$, $A(x) \neq B(x)$, $A(x) \leq B(x)$, $A(x) < B(x)$, $A(x) > B(x)$, $A(x) \geq B(x)$, onde A e B são funções quaisquer de x . No entanto, através de manipulações algébricas, todas podem ser postas na forma $h(x) = 0$ ou $g(x) \leq 0$, permitindo manter a forma dada pela Equação (6) sem perda de generalidade.

1.2.3.2 Restrições Diretas e Funcionais

A inclusão $x \in \Upsilon$ define as denominadas **Restrições Diretas**, enquanto as restrições de igualdade e desigualdade constituem as chamadas **Restrições Funcionais**.

As restrições diretas sempre podem ser escritas como restrições funcionais. Por exemplo, suponhamos as restrições diretas $x \in \Upsilon$, onde $\Upsilon = \mathbb{R}_+^n$, podem ser expressas pelas restrições funcionais $x_i \geq 0, i = 1, 2, 3, \dots, n$. Por este motivo, a separação entre restrições diretas e funcionais é apenas uma formalidade, não existindo implicações relevantes à mesma.

1.2.3.3 Transformações entre Restrições de Igualdade e de Desigualdade

Existe a possibilidade de converter uma restrição de igualdade em duas restrições desigualdade que funcionam de forma equivalente, conforme mostra a Equação (8).

$$h_a(x) = 0 \Leftrightarrow h_a(x) \leq 0 \wedge -h_a(x) \leq 0 \quad (8)$$

Uma restrição de desigualdade também pode ser convertida em uma restrição de igualdade por meio da introdução de uma variável artificial $z \in \mathbb{R}^l$, comumente chamada variável de folga, conforme evidencia a Equação (9)

$$g_b(x) \leq 0 \Leftrightarrow g_b(x) + z_b^2 = 0 \quad (9)$$

Embora essas transformações permitam escrever o conjunto viável de forma mais simples, realizar estas conversões amplia o número de restrições, elevando o grau de dificuldade da resolução do problema de otimização, tanto do ponto de vista teórico, quanto do ponto de vista computacional (IZMAILOV; SOLODOV, 2014).

Nota ao Leitor 4 *Devido às razões supracitadas, adotaremos, nesta tese, problemas de*

otimização no formato geral de restrições mistas de igualdade e desigualdade.

1.2.3.4 Classificação de Problemas de Otimização Segundo às Restrições

Quando em um problema de otimização temos $\Upsilon \in \mathbb{R}^n$, $k > 0$ e $l = 0$, este problema é denominado **Problema com Restrições de Igualdade**. Por outro lado, quando $\Upsilon \in \mathbb{R}^n$, $k = 0$ e $l > 0$, o problema é chamado de **Problema com Restrições de Desigualdade**.

No caso de um problema em que $\Upsilon \in \mathbb{R}^n$, $k > 0$ e $l > 0$, este é dito **Problema com Restrições Mistas**. Cabe ressaltar que quando $\Upsilon = \mathbb{R}^n$, $k = 0$ e $l = 0$, recaímos em um **problema de otimização irrestrita**, não podendo assim, um problema de otimização restrita ter esta configuração.

1.2.3.5 Representação de um Problema de Otimização Restrita

Embora a notação dada pela Equação (1) represente canonicamente um problema de otimização, usar uma notação para representar o problema e outra para definir o conjunto viável não é muito adotada na literatura, que opta, de maneira ampla, pelo uso de uma notação condensada, tal como a denotada pela Equação (10), que apresenta o problema e suas restrições ao mesmo tempo.

$$\begin{aligned}
 & \underset{x}{\text{Minimizar}} && f(x) \\
 & \text{Sujeito a} && h(x) = (0, \dots, 0) \\
 & && g(x) \leq (0, \dots, 0) \\
 & && x \in \Upsilon
 \end{aligned} \tag{10}$$

Uma outra explicitação na notação adotada pela literatura são as **Restrições de Intervalo** das variáveis do problema de otimização, também chamadas de **Restrições de Caixa**. Essa explicitação não introduz perda de generalidade ao problema, uma vez que uma variável sem restrição em seu intervalo de valores, na verdade, se encontra no intervalo $(-\infty, \infty)$.

Embora esses intervalos possam ser expressos como restrições de desigualdade, sendo duas restrições por variável, apresentar esses intervalos de forma explícita na notação, evita sobrecarregá-la com muitas restrições de desigualdade e deixa tal informação mais clara, especialmente quando estes forem ser resolvidos numericamente, onde a existência dos intervalos é mandatória.

A Equação (11) mostra a notação com as restrições de caixa explicitadas, onde ϵ é

um vetor da forma $\epsilon = (\epsilon_1, \epsilon_2, \epsilon_3, \dots, \epsilon_n)$, que indica os limites inferiores das variáveis e ω é um vetor da forma $\omega = (\omega_1, \omega_2, \omega_3, \dots, \omega_n)$, que define os limites superiores das variáveis do problema de otimização.

$$\begin{aligned}
 &\text{Minimizar}_{x} && f(x) \\
 &\text{Sujeito a} && h(x) = (0, \dots, 0) \\
 & && g(x) \leq (0, \dots, 0) \\
 & && \epsilon \leq x \leq \omega \\
 & && x \in \Upsilon
 \end{aligned} \tag{11}$$

Uma outra forma de notação muito encontrada da literatura é a apresentada na Equação (12). Ela representa os elementos dos vetores diretamente na notação, apresentando todos os elementos do problema diretamente nela, dispensando as definições prévias dos mesmos.

$$\begin{aligned}
 &\text{Minimizar}_{x_1, x_2, \dots, x_n} && f(x_1, x_2, \dots, x_n) \\
 &\text{Sujeito a} && h_1(x_1, x_2, \dots, x_n) = 0 \\
 & && h_2(x_1, x_2, \dots, x_n) = 0 \\
 & && \vdots \\
 & && h_k(x_1, x_2, \dots, x_n) = 0 \\
 & && g_1(x_1, x_2, \dots, x_n) \leq 0 \\
 & && g_2(x_1, x_2, \dots, x_n) \leq 0 \\
 & && \vdots \\
 & && g_l(x_1, x_2, \dots, x_n) \leq 0 \\
 & && \epsilon_1 \leq x_1 \leq \omega_1 \\
 & && \epsilon_2 \leq x_2 \leq \omega_2 \\
 & && \vdots \\
 & && \epsilon_n \leq x_n \leq \omega_n \\
 & && x \in \Upsilon
 \end{aligned} \tag{12}$$

1.2.4 Problema de Otimização Convexa

Uma importante classificação dos problemas de otimização é quanto à convexidade das funções envolvidas nele, tratando-se estas da função objetivo e das funções que configuram as restrições de igualdade e desigualdade.

Quando todas as funções presentes no problema de otimização forem convexas ou afins, o problema é dito **Problema de Otimização Convexa (POC)**, conforme enuncia a Definição 1.2.4. Este tipo de problemas possui características e propriedades exclusivas no que tange à otimização.

Definição 1.2.4 (Problema de Otimização Convexa) *O problema de minimização dado pela Equação (10) é dito convexo se, e somente se, a função f e todas as componentes de g e h forem funções convexas.*

1.2.5 Programação Linear e Não Linear

Outra importante classificação dos problemas de otimização é quanto à linearidade das funções envolvidas no problema, sendo estas compostas pela função objetivo e as funções presentes nas restrições de igualdade e de desigualdade (ANDRETTA, 2016).

Quando todas as funções presentes no problema de otimização forem lineares, o problema é denominado **Problema de Programação Linear (PPL)**, conforme denota a Definição 1.2.5.

Definição 1.2.5 (Problema de Programação Linear) *O problema de minimização dado pela Equação (10) é denominado problema de programação linear se, e somente se, a função f e todas as componentes de g e h forem funções lineares.*

Caso apenas uma das funções envolvidas no problema de otimização não for linear, o problema deixa de ser considerado PPL e passa a ser chamado de **Problema de Programação Não Linear (PPNL)**, conforme estabelece a Definição 1.2.6.

Definição 1.2.6 (Problema de Programação Não Linear) *O problema de minimização dado pela Equação (10) é denominado problema de programação não linear se, e somente se, a função f ou pelo menos uma das componentes de g e h não for uma função linear.*

Grande parte dos problemas de otimização, especialmente os problemas reais e os que requerem dados experimentais, são problemas que possuem não linearidades em suas funções, tratando-se de problemas de programação não linear (BAZARAA; JARVIS, 1977).

Nota ao Leitor 5 *Nesta tese, objetivamos abordar, com generalidade, os problemas de otimização, admitindo tanto problemas lineares, quanto não lineares. Por este motivo, não trataremos de aspectos e comportamentos particulares dos problemas lineares.*

1.2.6 Conjunto Objetivo de um Problema de Otimização

O conjunto viável de um problema de otimização não é sua solução, mas um conjunto que possui candidatos à solução, uma vez que seus elementos respeitam as restrições impostas pelo problema, podendo ser entendido como o domínio do problema de otimização.

Em contrapartida, temos o conceito de *Conjunto Objetivo* ou espaço objetivo, que é definido na Equação (13). Este conjunto consiste, em um problema de otimização mono-objetivo, na imagem da função objetivo $f : \Omega \rightarrow \mathbb{R}$, podendo ser entendido como a imagem do problema de otimização.

$$\mathcal{U} = \text{Im}(f) = \{f(x), \forall x \in \Omega\} \quad (13)$$

1.2.7 Solução de um Problema de Otimização Mono-Objetivo

A solução de um problema de otimização, também denominada **Minimizador**, em um problema de otimização mono-objetivo, consiste no vetor de variáveis $x \in \Omega$ que corresponde ao menor valor presente na imagem da função objetivo f . Otimizar ou resolver um problema de otimização consiste no processo de encontrar este minimizador.

1.2.7.1 Minimizador Global

A concepção de solução apresentada acima remete idealmente ao conceito de minimizador global, uma vez que este minimizador leva ao menor valor da imagem da função objetivo, que é denominado mínimo global. A Definição 1.2.7 expressa o conceito de minimizador global.

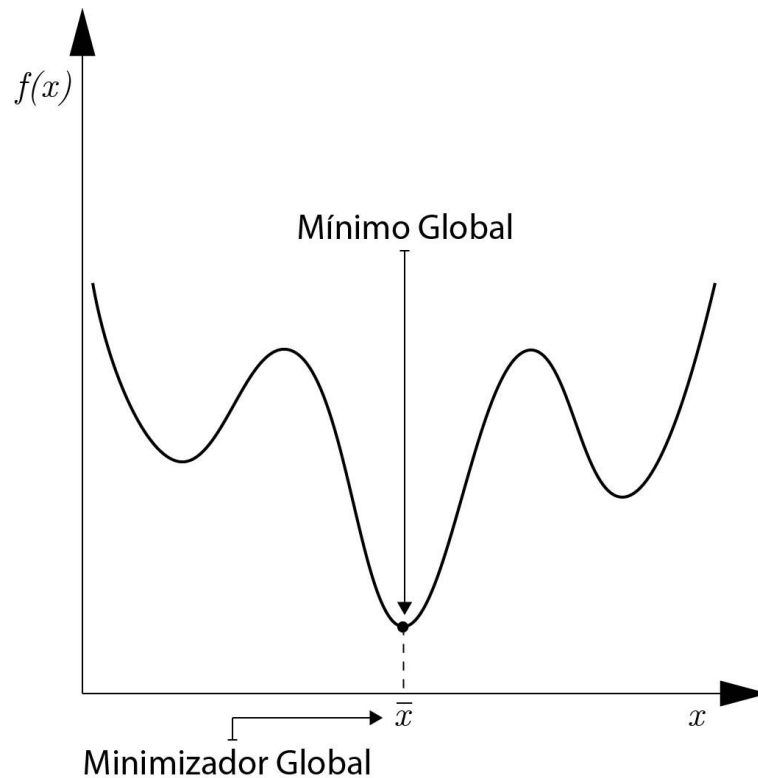
Definição 1.2.7 (Minimizador Global) *Um elemento $\bar{x} \in \Omega$ é dito minimizador global se $f(\bar{x}) \leq f(x)$ para todo $x \in \Omega$.*

A Figura 2 apresenta um exemplo gráfico de minimizador global considerando uma função objetivo $f : \mathbb{R} \rightarrow \mathbb{R}$.

1.2.7.2 Minimizador Local

Idealmente, almeja-se sempre encontrar o minimizador global de um problema de otimização, uma vez que isso significa encontrar o elemento para o qual o valor da função

Figura 2 - Minimizador global e mínimo global.



Fonte: O autor, 2023.

objetivo seja o menor possível. No entanto, encontrar um minimizador global pode ser difícil ou inviável.

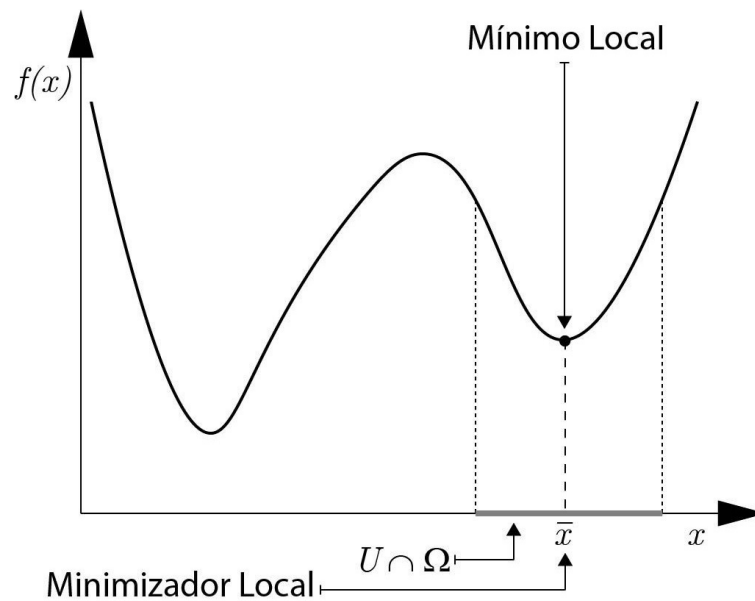
Um dos motivos que tornam inviável o encontro da solução global é que em problemas reais, geralmente não conhecemos o comportamento de toda a função objetivo, tendo apenas o conhecimento dela em um dado intervalo. Outro motivo que provoca essa dificuldade é o alto custo computacional para encontrar a solução global em problemas de alta complexidade. Existe também o fato da maioria dos algoritmos não apresentar garantia de obter a solução global. Pode ser ainda, que exista o interesse em apenas uma determinada região da função, não fazendo sentido a busca pelo minimizador global.

Em decorrência dos fatores apresentados, nestes casos, deseja-se obter soluções que levem ao menor valor da função objetivo em uma determinada vizinhança. Esse tipo de solução é chamada de minimizador local, sendo formalmente expressa pela Definição 1.2.8.

Definição 1.2.8 (Minimizador Local) *Um elemento $\bar{x} \in \Omega$ é um minimizador local se existe uma vizinhança U de \bar{x} tal que $f(\bar{x}) \leq f(x)$ para todo $x \in \{U \cap \Omega\}$.*

A Figura 3 apresenta um exemplo gráfico de minimizador local considerando uma função objetivo $f : \mathbb{R} \rightarrow \mathbb{R}$.

Figura 3 - Um minimizador local e seu mínimo local correspondente.



Legenda: Embora o mínimo global seja um ponto à esquerda de \bar{x} , se considerarmos apenas a vinhança U de \bar{x} dentro do conjunto viável Ω , o minimizador local é o ponto \bar{x} .

Fonte: O autor, 2023.

1.2.8 Minimizador Local Estrito

Existem problemas nos quais pode-se encontrar diversos minimizadores locais que levam ao mesmo valor de mínimo da função objetivo. Diversas vezes, esses minimizadores não são relevantes. Um exemplo clássico de uma situação deste tipo é quando se tem um problema de otimização cuja função objetivo é constante em um determinado intervalo. Neste intervalo, todos os elementos são minimizadores locais, mas que não apresentam relevância para o problema em questão.

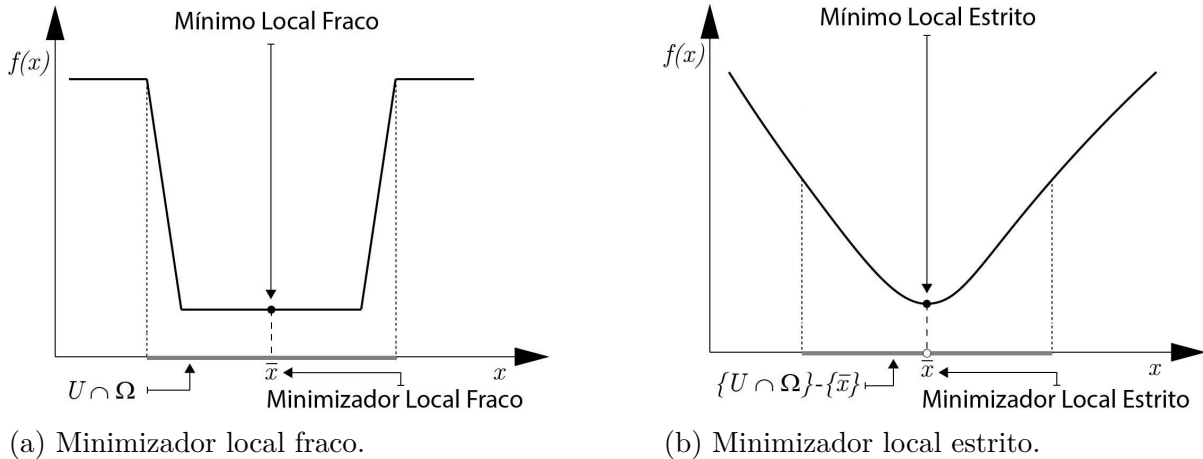
Quando num problema de otimização, temos, em uma vizinhança, mais de um minimizador local que leva a um mesmo valor de mínimo local da função objetivo, denominamos esse tipo de minimizador de **Minimizador Local Fraco**.

Por outro lado, quando em um problema de otimização, em uma vizinhança, existe um único minimizador local que leva a um valor de mínimo local da função objetivo, chamamos este minimizador de **Minimizador Local Estrito**. Esse conceito é formalmente expresso pela Definição 1.2.9.

Definição 1.2.9 (Minimizador Local Estrito) *Um elemento $\bar{x} \in \Omega$ é um minimizador local estrito se existe uma vizinhança U de \bar{x} tal que $f(\bar{x}) < f(x)$ para todo $x \in \{U \cap \Omega\} - \{\bar{x}\}$.*

A Figura 4 ilustra um exemplo de minimizador local fraco e um exemplo de minimizador local estrito, ambos em funções de uma variável.

Figura 4 - Minimizador local fraco e minimizador local estrito.



Legenda: Na função de (a), \bar{x} é um minimizador local fraco devido à existência de diversos pontos em sua vizinhança U pertencentes ao conjunto viável Ω que levam ao mesmo valor de mínimo que ele conduz. Por outro lado, na função mostrada em (b), não existem outros pontos pertencentes à vizinhança U de \bar{x} no conjunto viável Ω que levam a um ponto menor ou igual ao que \bar{x} leva.

Fonte: O autor, 2023.

1.2.8.1 Minimizador em Problemas de Otimização Convexa

Os problemas de otimização convexa possuem uma importante propriedade que garante que todo minimizador deste tipo de problema é um minimizador global. O Corolário 1.2.1 expressa formalmente essa propriedade.

Corolário 1.2.1 *Se o problema de minimização dado pela Equação (10) for um problema de otimização convexa, todo minimizador local deste problema é minimizador global.*

1.3 Otimização Multiobjetivo

No decorrer desta seção, serão abordados os conceitos e características dos problemas de otimização que envolvem mais de um objetivo. Este tipo de problema será suportado pela ferramenta computacional desenvolvida neste trabalho e os problemas abordados nesta tese serão desta natureza. Por esta razão, a compreensão do assunto por parte do leitor é de suma importância para o entendimento desta produção.

Quando um problema de otimização possui mais de um objetivo a ser otimizado, denominados este problema como Problema de Otimização Multiobjetivo. Formalmente, um problema de otimização multiobjetivo é enunciado pela Definição 1.3.1.

Definição 1.3.1 (Problema de Otimização Multiobjetivo) *Seja um conjunto $\Omega \subseteq \mathbb{R}^n$, um vetor x de forma $x = (x_1, x_2, x_3, \dots, x_n)$ e uma função $f : \Omega \rightarrow \mathbb{R}^m$ cujas componentes têm a forma $f(x) = \langle f_1(x), f_2(x), f_3(x), \dots, f_m(x) \rangle$ com $f_k : \Omega \rightarrow \mathbb{R}, \forall k \in \{1, 2, 3, \dots, m\}$, um problema de otimização multiobjetivo pode ser denotado pela Equação (14), onde o conjunto Ω é denominado **conjunto viável**, $f(x)$ configura o conjunto de **funções objetivo**, onde cada uma de suas componentes é uma função objetivo, e x é o vetor de **variáveis** a serem manipuladas a fim de otimizar as funções objetivo, sendo $n \in \mathbb{N}$ o número de variáveis e $m \in \mathbb{N}$ o número de funções objetivo do problema.*

$$\begin{aligned} \underset{x}{\text{Minimizar}} \quad & f(x) = \langle f_1(x), f_2(x), f_3(x), \dots, f_m(x) \rangle \\ \text{Sujeito a} \quad & x \in \Omega \end{aligned} \tag{14}$$

1.3.1 Problema de Minimização e Problema de Maximização

O problema de otimização definido na Equação (14) é um **Problema de Minimização**. Da mesma forma que na otimização mono-objetivo, um problema de otimização multiobjetivo também pode ser um **Problema de Maximização**, sendo denotado pela forma expressa pela Equação (15).

$$\begin{aligned} \underset{x}{\text{Maximizar}} \quad & f(x) = \langle f_1(x), f_2(x), f_3(x), \dots, f_m(x) \rangle \\ \text{Sujeito a} \quad & x \in \Omega \end{aligned} \tag{15}$$

1.3.1.1 Problema Misto de Minimização e Maximização

Como um problema de otimização multiobjetivo possui diversas funções objetivo, pode-se desejar minimizar um grupo de funções $f_1(x), f_2(x), f_3(x), \dots, f_p(x)$, onde $p \in \mathbb{N}, 1 \leq p \leq m$, e maximizar um outro grupo de funções $f_{p+1}(x), f_{p+2}(x), f_{p+3}(x), \dots, f_m(x)$, gerando um problema misto de minimização e maximização, denotado pela Equação (16), onde busca-se encontrar para quais valores das variáveis de x , as funções $f_1(x), f_2(x), f_3(x), \dots, f_p(x)$ apresentam os menores valores e as funções $f_{p+1}(x), f_{p+2}(x), f_{p+3}(x), \dots, f_m(x)$

apresentam os maiores valores, simultaneamente.

$$\begin{aligned}
 \text{Minimizar}_x & f_1(x), f_2(x), f_3(x), \dots, f_p(x) \\
 \text{Maximizar}_x & f_{p+1}(x), f_{p+2}(x), f_{p+3}(x), \dots, f_m(x) \\
 \text{Sujeito a} & x \in \Omega
 \end{aligned} \tag{16}$$

1.3.1.2 Conversão entre Problemas de Minimização e Maximização

Do mesmo modo que nos problemas de otimização mono-objetivo, pode-se transformar uma maximização em minimização através da inversão do sinal da função. Em um problema onde se deseja maximizar todas as funções objetivo, para modelá-lo como um problema de minimização deve-se inverter o sinal das funções objetivo, configurando o problema expresso pela Equação (17).

$$\begin{aligned}
 \text{Minimizar}_x & -f(x) = \langle -f_1(x), -f_2(x), -f_3(x), \dots, -f_m(x) \rangle \\
 \text{Sujeito a} & x \in \Omega
 \end{aligned} \tag{17}$$

No caso de um problema onde deseja-se minimizar um grupo de funções objetivo $f_1(x), f_2(x), f_3(x), \dots, f_p(x)$ e maximizar um outro grupo $f_{p+1}(x), f_{p+2}(x), f_{p+3}(x), \dots, f_m(x)$, para modelá-lo puramente como um problema de minimização, deve-se inverter apenas sinal das funções que se deseja maximizar, configurando o problema dado pela Equação (18).

$$\begin{aligned}
 \text{Minimizar}_x & f_1(x), f_2(x), f_3(x), \dots, f_p(x), -f_{p+1}(x), -f_{p+2}(x), -f_{p+3}(x), \dots, -f_m(x) \\
 \text{Sujeito a} & x \in \Omega
 \end{aligned} \tag{18}$$

Nota ao Leitor 6 *Assim como nos problemas de otimização mono-objetivo, por não existirem diferenças relevantes nos conceitos e solução de problemas de minimização e maximização multiobjetivo, consideraremos aqui apenas problemas de minimização, ressaltado que os conceitos se aplicam também aos de maximização.*

1.3.2 Validade dos Conceitos da Otimização Mono-Objetivo na Multiobjetivo

A maioria dos conceitos, propriedades e características dos problemas de otimização mono-objetivo se aplicam também à otimização multiobjetivo. Desse modo, os conceitos de função custo e utilidade, otimização restrita e irrestrita, forma do conjunto viável, conjunto objetivo, restrições de igualdade e desigualdade, restrições diretas e funcionais

e classificação segundo às restrições também são válidos na otimização multiobjetivo. Por este motivo, esses conceitos não serão definidos novamente para os problemas de otimização multiobjetivo.

No entanto, os conceitos de solução, condições de existência e condições de otimalidade dos problemas de otimização mono-objetivo não se aplicam nos problemas de otimização multiobjetivo. Dessa forma, a otimização multiobjetivo possui conceitos próprios e exclusivos, além de outros que possuem definição diferente de seus equivalentes na otimização mono-objetivo.

1.3.3 Representação de um Problema de Otimização Multiobjetivo

Assim como na otimização mono-objetivo, a notação dada pela Equação (14) representa canonicamente um problema de otimização multiobjetivo, mas do mesmo modo, essa notação não é predominantemente adotada na literatura, que opta de maneira ampla pelo uso de uma notação mais explícita, tal como a denotada pela Equação (19), que apresenta o problema e suas restrições explicitamente na própria notação.

$$\begin{aligned}
 &\underset{x}{\text{Minimizar}} && f(x) \\
 &\text{Sujeito a} && h(x) = (0, \dots, 0) \\
 &&& g(x) \leq (0, \dots, 0) \\
 &&& x \in \Upsilon
 \end{aligned} \tag{19}$$

A notação que apresenta as restrições de intervalo de forma explícita também é bastante presente na literatura no que tange a problemas de otimização multiobjetivo, estando esta denotada na Equação (20).

$$\begin{aligned}
 &\underset{x}{\text{Minimizar}} && f(x) \\
 &\text{Sujeito a} && h(x) = (0, \dots, 0) \\
 &&& g(x) \leq (0, \dots, 0) \\
 &&& \epsilon \leq x \leq \omega \\
 &&& x \in \Upsilon
 \end{aligned} \tag{20}$$

Finalmente, temos a notação dada pela Equação (21), que apresenta os elementos vetoriais explicitados diretamente na mesma, sendo esta a notação que apresenta todos

os elementos do problema de maneira unificada, evitando a definição prévia dos mesmos.

$$\begin{aligned}
 & \underset{x_1, x_2, \dots, x_n}{\text{Minimizar}} && f_1(x_1, x_2, \dots, x_n) \\
 & && f_2(x_1, x_2, \dots, x_n) \\
 & && \vdots \\
 & && f_m(x_1, x_2, \dots, x_n) \\
 \text{Sujeito a} & && h_1(x_1, x_2, \dots, x_n) = 0 \\
 & && h_2(x_1, x_2, \dots, x_n) = 0 \\
 & && \vdots \\
 & && h_k(x_1, x_2, \dots, x_n) = 0 \\
 & && g_1(x_1, x_2, \dots, x_n) \leq 0 \\
 & && g_2(x_1, x_2, \dots, x_n) \leq 0 \\
 & && \vdots \\
 & && g_l(x_1, x_2, \dots, x_n) \leq 0 \\
 & && \epsilon_1 \leq x_1 \leq \omega_1 \\
 & && \epsilon_2 \leq x_2 \leq \omega_2 \\
 & && \vdots \\
 & && \epsilon_n \leq x_n \leq \omega_n \\
 & && x \in \Upsilon
 \end{aligned} \tag{21}$$

1.3.4 Solução de um Problema de Otimização Multiobjetivo

O conceito de solução de um problema multiobjetivo difere do de solução de problema de otimização mono-objetivo. Neste último, busca-se determinar para qual conjunto de variáveis, a função objetivo apresenta seu valor mínimo. Dessa forma, manipula-se os valores das variáveis dentro do conjunto viável com intuito de encontrar qual o conjunto de valores traz o menor custo para a função objetivo, retornando apenas o melhor minimizador encontrado no processo (STORN; PRICE, 1997a).

Em um problema de otimização multiobjetivo, por sua vez, procura-se conjuntos de variáveis que levem simultaneamente várias funções objetivo aos seus menores valores, respectivamente. No entanto, um mesmo conjunto de variáveis pode fazer com que uma função objetivo tenha o valor mínimo, mas outra tenha o valor máximo, além de não se ter garantias de que exista uma mesma solução que minimize todos os objetivos. Essa problemática acarreta no fato de um problema de otimização multiobjetivo não possuir uma única solução, mas sim um conjunto de várias soluções factíveis (DEB, 2001).

1.3.4.1 Conflitos entre Objetivos

Idealmente, se todos os objetivos de um problema de otimização não apresentarem conflitos entre si, é possível encontrar uma única solução os minimize simultaneamente.

No entanto, os problemas de otimização multiobjetivo, predominantemente, apresentam valores de custos concorrentes, gerando um conflito entre seus objetivos. Isso significa que a melhoria na otimização de um objetivo pode acarretar na piora de outro. Essa problemática dificulta ou até mesmo impossibilita encontrar soluções que sejam simultaneamente satisfatórias para todos os objetivos do problema.

Desse modo, uma solução pode otimizar bem alguns objetivos, mas não outros, enquanto outras podem fazer o contrário. Por esta razão, normalmente não existe um único vetor de variáveis capaz de minimizar todas as funções objetivo ao mesmo tempo.

Esse fato nos leva a concluir que um único vetor de variáveis não compõe totalmente a solução de um problema de otimização multiobjetivo, mas sim vários vetores, onde cada um otimiza pelo menos uma função objetivo melhor que todas as demais.

Logo, a otimização multiobjetivo tem o intuito de encontrar um conjunto de vetores que contenham as soluções que melhor otimizem uma função objetivo, de forma que este seja um conjunto equilibrado de soluções no que tange à otimização de todas as funções objetivo do problema.

1.3.4.2 Dominância

O conceito de dominância de uma solução está relacionado à análise comparativa entre duas soluções diferentes de um problema de otimização multiobjetivo. Ela indica a superioridade ou não de uma solução sobre outra. Trata-se de um conceito sumamente importante para a otimização multiobjetivo (ABRAHAM; JAIN; GOLDBERG, 2005; MIETTINEN, 1999).

Considerando um problema de otimização multiobjetivo, podemos relacionar duas possíveis soluções v_1 e v_2 segundo a quatro tipos de relacionamento de dominância, a saber:

1. v_1 domina fracamente v_2 ou v_2 é dominada fracamente por v_1 ;
2. v_1 domina v_2 ou v_2 é dominada por v_1 ;
3. v_1 domina estritamente v_2 ou v_2 é dominada estritamente por v_1 ;
4. Não existe relação de dominância entre v_1 e v_2 .

Cada um dos conceitos que estabelecem as relações descritas acima serão apresentados formalmente a seguir.

1.3.4.2.1 Dominância Fraca

A dominância fraca é uma relação que garante que uma possível solução não é pior que outra, sendo pelo menos igual em todos os objetivos. A Definição 1.3.2 estabelece o conceito de dominância fraca.

Definição 1.3.2 (Dominância Fraca) *Seja o problema de minimização dado pela Equação (14) e os vetores $v_1, v_2 \in \Omega$, dizemos que v_1 domina fracamente v_2 , denotando por $v_1 \preceq v_2$, se v_1 não for pior que v_2 em nenhum objetivo, isto é, se a Equação (22) se verificar.*

$$\forall l \in \{1, 2, 3, \dots, m\} : f_l(v_1) \leq f_l(v_2) \quad (22)$$

1.3.4.2.2 Dominância

O conceito de dominância é mais forte que o de dominância fraca. Ele garante que uma possível solução é ligeiramente melhor que outra, uma vez que em pelo menos um objetivo, ela é superior. A Definição 1.3.3 determina o conceito de dominância.

Definição 1.3.3 (Dominância) *Seja o problema de minimização dado pela Equação (14) e os vetores $v_1, v_2 \in \Omega$, dizemos que v_1 domina v_2 , denotando por $v_1 \prec v_2$, se v_1 for melhor que v_2 em, ao menos, um objetivo e v_1 não for pior que v_2 em nenhum objetivo, isto é, se a Equação (23a) e a Equação (23b) forem satisfeitas.*

$$\forall l \in \{1, 2, 3, \dots, m\} : f_l(v_1) \leq f_l(v_2) \quad (23a)$$

$$\exists l \in \{1, 2, 3, \dots, m\} : f_l(v_1) < f_l(v_2) \quad (23b)$$

1.3.4.2.3 Dominância Forte

A dominância forte ou dominância estrita é a mais forte relação de dominância, garantindo que uma possível solução é completamente melhor que outra, pois esta será superior em todos os objetivos em relação à outra. A Equação (24) especifica o conceito de dominância forte.

Definição 1.3.4 (Dominância Forte) *Seja o problema de minimização dado pela Equação (14) e os vetores $v_1, v_2 \in \Omega$, dizemos que v_1 domina fortemente ou estritamente v_2 ,*

denotando por $v_1 \preceq v_2$, se v_1 for melhor que v_2 em todos os objetivos, isto é, se a Equação (24) for atendida.

$$\forall l \in \{1, 2, 3, \dots, m\} : f_l(v_1) < f_l(v_2) \quad (24)$$

1.3.4.2.4 Inexistência de Relação de Dominância

Pode ser que entre duas possíveis soluções não exista uma relação de dominância. Essa ausência de dominância se dá quando consideradas duas possíveis soluções, cada uma delas será pior em algum objetivo com relação a outra. A Equação (25) caracteriza a inexistência de relação de dominância.

Definição 1.3.5 (Inexistência de Relação de Dominância) *Seja o problema de minimização dado pela Equação (14) e os vetores $v_1, v_2 \in \Omega$, dizemos que não existe relação de dominância entre v_1 e v_2 , denotando por $v_1 \asymp v_2$, se v_1 for pior que v_2 em pelo menos um objetivo e v_2 for pior que v_1 em pelo menos um objetivo, isto é, se Equação (25a) e a Equação (25b) se verificarem.*

$$\exists l \in \{1, 2, 3, \dots, m\} : f_l(v_1) > f_l(v_2) \quad (25a)$$

$$\exists l \in \{1, 2, 3, \dots, m\} : f_l(v_2) > f_l(v_1) \quad (25b)$$

Nota ao Leitor 7 *Para problemas de maximização, as definições de dominância apresentadas são correspondentemente válidas.*

1.3.4.2.5 Considerações sobre Dominância e Otimização Multiobjetivo

A dominância fraca é a mais comum das relações de dominância encontrada nos problemas de otimização multiobjetivo. No entanto, ela não dá garantia alguma de que uma possível solução é melhor que outra.

Por outro lado, a dominância forte é uma relação que garante total superioridade de uma possível solução sobre outra. Porém, esse tipo de dominância é bem menos frequente nos problemas de otimização multiobjetivo.

Na otimização multiobjetivo, quando se versa sobre solução, o conceito de dominância é o predominantemente empregado, uma vez que ele garante que uma solução é melhor que outra, diferentemente da dominância fraca, e é mais frequente que a domi-

nância forte, o que confere maior diversidade de soluções para o problema de otimização multiobjetivo.

Nota ao Leitor 8 *Nas seções que se sucederão, neste capítulo, as referências à dominância remeterão à relação de dominância. As menções às relações de dominância fraca e dominância forte serão feitas de forma explícita.*

1.3.4.3 Otimalidade de Pareto

O conceito de otimalidade em problemas de otimização foi proposto inicialmente por Edgeworth (1881), sendo generalizada posteriormente através de Pareto (1896), consolidando o conceito na área.

Vilfredo Pareto foi um cientista político, sociólogo e economista franco-italiano que prestou relevantes contribuições teóricas para a sociologia e economia, tais como a Lei de Pareto, o Equilíbrio de Pareto e a Eficiência de Pareto (MCLURE, 2001).

Embora as contribuições de Pareto supracitadas tenham sido inicialmente direcionadas à economia, elas foram estendidas a diversas outras áreas, incluindo a otimização, onde tem papel de destaque nos problemas multiobjetivos.

Por conseguinte, neste trabalho, nos interessam as contribuições de Pareto que colaboram diretamente com os problemas de otimização multiobjetivo. Dessa forma, os conceitos que fazem referência a Vilfredo Pareto serão apresentados e descritos na contextualização da otimização multiobjetivo.

1.3.4.3.1 Equilíbrio e Eficiência de Pareto

Em problemas com múltiplos objetivos, onde existem interesses que são conflitantes entre si, objetiva-se alocar os recursos da forma mais eficiente possível. Quando se atinge o cenário onde todos os recursos estão dispostos da forma mais eficiente, diz-se que se atingiu o Equilíbrio de Pareto (PARETO, 1896; COELLO, 2006).

Uma vez atingido o equilíbrio de Pareto, qualquer esforço para realocar recursos a fim de melhorar um objetivo, vai necessariamente, deteriorar pelo menos um outro. Quando isso ocorre, dizemos que a Eficiência de Pareto ou Ótimo de Pareto foi atingido, sendo este o estado onde os recursos estão alocados da forma mais eficiente possível (PARETO, 1896; COELLO, 2006).

Com base nestes princípios, serão definidos formalmente a seguir, os conceitos inerentes a eles no contexto dos problemas de otimização multiobjetivo.

1.3.4.3.2 Otimizador de Pareto

Um otimizador de Pareto consiste em um vetor de variáveis do conjunto viável que não é dominado por nenhum outro vetor deste mesmo conjunto. A Definição 1.3.6 expressa categoricamente o conceito de otimizador de Pareto.

Definição 1.3.6 (Otimizador de Pareto) *Seja o problema de minimização dado pela Equação (14), um vetor de variáveis de decisão $\bar{x} \in \Omega$ é dito um otimizador de Pareto ou minimizador de Pareto se, e somente se, não existir um vetor $x \in \Omega$ tal que $x \prec \bar{x}$.*

1.3.4.3.3 Ótimo de Pareto

Um ótimo de Pareto é um elemento do conjunto objetivo correspondente a um otimizador de Pareto do conjunto viável. Ele expressa um dos melhores resultados do problema de otimização multiobjetivo. A Definição 1.3.7 expressa precisamente o conceito de ótimo de Pareto.

Definição 1.3.7 (Ótimo de Pareto) *Seja o problema de minimização dado pela Equação (14) e seu conjunto objetivo \mathcal{U} , um vetor de objetivos $\bar{o} \in \mathcal{U}$ é dito um ótimo de Pareto ou mínimo de Pareto se não existir um vetor $o \in \mathcal{U}$ tal que a Equação (26a) e a Equação (26b) sejam satisfeitas.*

$$\forall l \in \{1, 2, 3, \dots, m\} : o_l \leq \bar{o}_l \quad (26a)$$

$$\exists l \in \{1, 2, 3, \dots, m\} : o_l < \bar{o}_l \quad (26b)$$

1.3.4.3.4 Conjunto Ótimo de Pareto

Quando se agrupa todos os otimizadores de Pareto do conjunto viável de um problema de otimização, o resultado é o que chamamos de conjunto ótimo de Pareto. Este conjunto contém os melhores vetores de variáveis viáveis, não sendo estes dominados por nenhum outro. A Definição 1.3.8 apresenta essencialmente o conceito de conjunto ótimo de Pareto.

Definição 1.3.8 (Conjunto Ótimo de Pareto) *Seja o problema de minimização dado pela Equação (14), o conjunto Θ dado pela Equação (27) é denominado conjunto ótimo de Pareto.*

$$\Theta = \{\bar{x} \in \Omega \mid \nexists x \in \Omega : x \prec \bar{x}\} \quad (27)$$

1.3.4.3.5 Fronteira de Pareto

O conjunto de todos os ótimos de Pareto do espaço objetivo de um problema de otimização constitui o que chamamos de fronteira de Pareto, frente de Pareto ou curva de Pareto. Ela contém os melhores resultados do problema de otimização. A Definição 1.3.9 enuncia rigorosamente o conceito de fronteira de Pareto.

Definição 1.3.9 (Fronteira de Pareto) *Seja o problema de minimização dado pela Equação (14), sua fronteira de Pareto \mathcal{F} é o conjunto dado pela Equação (28).*

$$\mathcal{F} = \{f(\bar{x}) \in \mathbb{R}^m \mid \bar{x} \in \Theta\} \quad (28)$$

A Figura 5 ilustra alguns exemplos de fronteira de Pareto referentes a problemas de otimização com dois objetivos.

1.3.4.3.6 Conjunto Solução de um Problema de Otimização

Definidos todos os conceitos necessários, pode-se então, definir o conceito de conjunto solução de um problema de otimização multiobjetivo. A Definição 1.3.10 estabelece formalmente o conceito de conjunto solução de um problema de otimização multiobjetivo, tratando-se do conjunto ótimo de Pareto do problema em questão, isto é, os melhores vetores de variáveis viáveis, uma vez que não são dominados por nenhum outro.

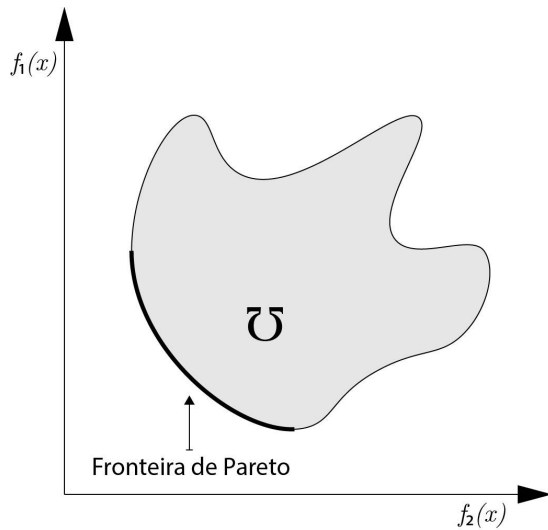
Definição 1.3.10 (Conjunto Solução de um Problema de Otimização Multiobjetivo) *Seja o problema de minimização dado pela Equação (14), seu conjunto solução consiste no conjunto ótimo de Pareto $\Theta \subset \Omega$ do problema de otimização.*

A Figura 6 ilustra os conceitos que envolvem o conjunto solução de um problema de otimização de duas variáveis e dois objetivos.

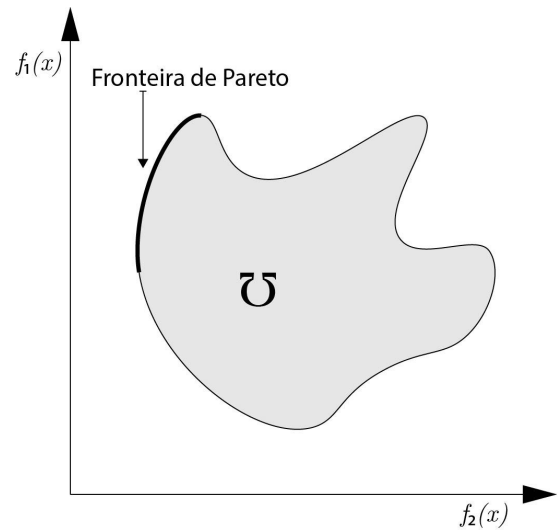
1.3.4.3.7 Ótimo Ideal

O conceito de ótimo ideal de um problema de otimização, embora não seja parte canônica do arcabouço teórico da otimização multiobjetivo, se faz importante em alguns métodos de otimização multiobjetivo.

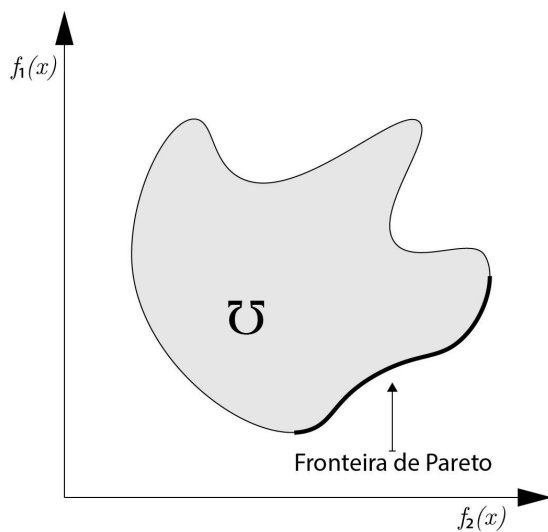
Figura 5 - Fronteira de Pareto.



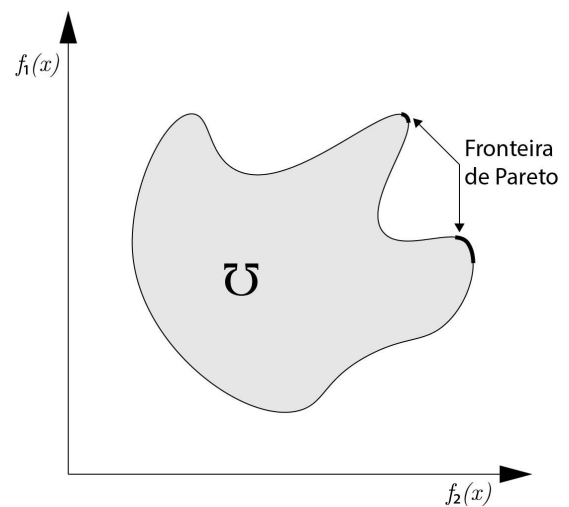
(a) Minimizar $f_1(x), f_2(x)$
 Sujeito a $x \in \Omega$.



(b) Maximizar $f_1(x)$ Minimizar $f_2(x)$
 Sujeito a $x \in \Omega$.



(c) Minimizar $f_1(x)$ Maximizar $f_2(x)$
 Sujeito a $x \in \Omega$.

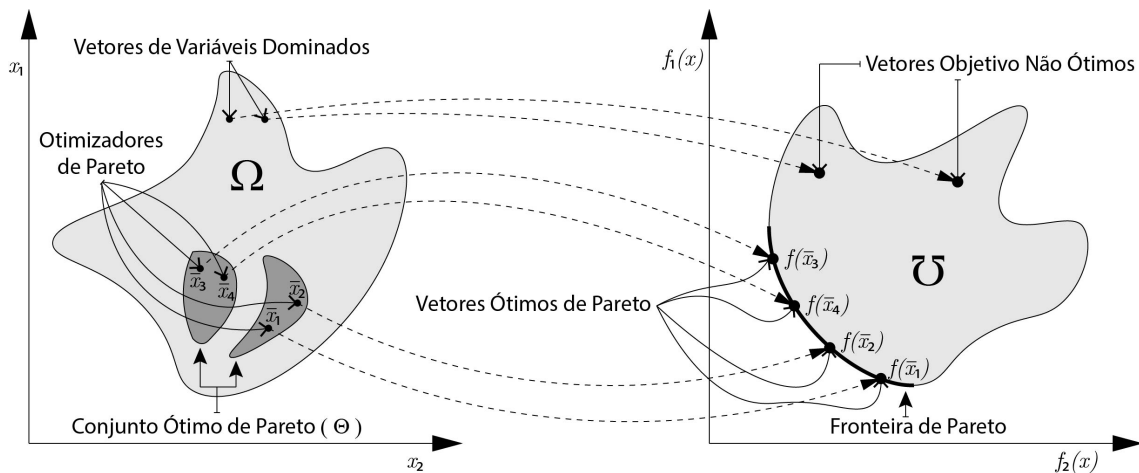


(d) Maximizar $f_1(x), f_2(x)$
 Sujeito a $x \in \Omega$.

Legenda: Os diagramas (a), (b), (c) e (d) mostram as fronteiras de Pareto referentes a diferentes problemas de otimização envolvendo as funções objetivo f_1 e f_2 , considerando o conjunto objetivo \mathcal{U} .

Fonte: O autor, 2023.

Figura 6 - Conjunto solução de um problema de otimização multiobjetivo e conceitos correlatos.



Legenda: Relação entre o conjunto ótimo de Pareto e a fronteira de Pareto do problema

Minimizar $f_1(x_1, x_2), f_2(x_1, x_2)$ Sujeito a $x_1, x_2 \in \Omega$, realçados os otimizadores de Pareto e ótimos de Pareto, estando mostrados também vetores de decisão dominados e vetores objetivo não ótimos relacionados a estes, considerando o conjunto viável Ω e o conjunto objetivo \mathcal{U} .

Fonte: O autor, 2023.

O ponto ótimo ideal de um problema de otimização multiobjetivo é um único ponto de mínimo que consiste no menor valor de todos os objetivos do problema de otimização simultaneamente.

Uma outra denominação para este conceito é ótimo utópico de um problema de otimização. Essa nomenclatura se deve ao fato de que raramente este ponto pertencerá ao conjunto objetivo \mathcal{U} , isto é, dificilmente existirá uma solução pertencente ao conjunto viável Ω que leve a este ponto.

Uma situação em que pode existir uma possibilidade deste ponto ser obtido é quando os objetivos do problema de otimização não forem conflitantes. Porém, na maioria das vezes, esse tipo de problema não possui muito interesse como objeto de estudo. Além de que problemas deste tipo constituem uma fração ínfima dos problemas de otimização.

As componentes do vetor ótimo ideal consistem nos pontos de mínimo de cada um dos objetivos individualmente. A Definição 1.3.11 enuncia formalmente o conceito de ótimo ideal de um problema de otimização multiobjetivo.

Definição 1.3.11 (Ótimo Ideal) *Seja o problema de minimização dado pela Equação (14), seu ponto ótimo ideal $\check{o} \in \mathbb{R}^m$ é o vetor $\check{o} = (\check{o}_1, \check{o}_2, \check{o}_3, \dots, \check{o}_m)$ em que cada componente $\check{o}_i = f_i(\check{x}_i)$, sendo f_i a i -ésima componente da função objetivo f e $\check{x}_i \in \mathbb{R}^n$ a solução do problema de otimização dado pela Equação (29).*

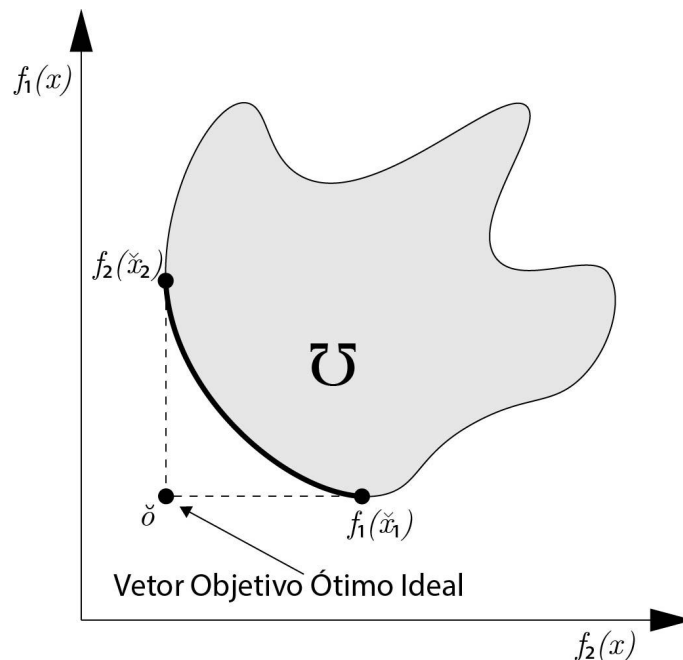
$$\tilde{x}_i = \begin{cases} \text{Minimizar}_x & f_i(x) \\ \text{Sujeito a} & x \in \Omega \end{cases} \quad (29)$$

Uma forma alternativa de definir o ponto ótimo ideal de um problema de otimização multiobjetivo \check{o} é estabelecer cada uma de suas componentes \check{o}_i através da componente o_i que possua o menor valor dentre todos os ótimos $o \in \mathcal{U}$, conforme expressa a Equação (30).

$$\check{o}_i = \min_{o \in \mathcal{U}}(o_i) \quad (30)$$

A Figura 7 ilustra um exemplo de ponto ideal referente a um problema de otimização envolvendo dois objetivos.

Figura 7 - Ótimo ideal.



Legenda: Ponto ótimo ideal referente ao problema de otimização

Minimizar $f_1(x), f_2(x)$ Sujeito a $x \in \Omega$.

Fonte: O autor, 2023.

1.3.4.3.8 Solução Local de um Problema de Otimização Multiobjetivo

Todos os conceitos vistos até aqui, nesta seção, são globais, isto é, são definidos para todo o conjunto viável ou todo o conjunto objetivo. Encontrar a solução global é o que se idealmente deseja. Porém, existem diversas razões que podem inviabilizar ou desinteressar o encontro da solução global, sendo estas já descritas na Seção 1.2.7.2.

Um dos principais fatores que inviabilizam encontrar a solução global é o fato dos algoritmos computacionais não garantirem a obtenção da solução global, exceto nos casos onde existir uma condição teórica, como convexidade, por exemplo, que garanta que a solução encontrada é global.

Por consequência dos motivos mostrados acima, fica-se satisfeito em obter uma solução local, que contém os melhores vetores de variáveis viáveis que correspondem aos melhores resultados dos objetivos em uma dada vizinhança.

Dessa forma, estabeleceremos formalmente, a seguir, os conceitos locais correlatos à solução local de um problema de otimização multiobjetivo.

1.3.4.3.9 Otimizador Local de Pareto

Um otimizador local de Pareto é constituído por um vetor de variáveis do conjunto viável que não é dominado por nenhum outro vetor deste conjunto que corresponda a um ótimo de Pareto presente na vizinhança do ótimo resultante do primeiro vetor. A Definição 1.3.12 expressa essencialmente o conceito de otimizador local de Pareto.

Definição 1.3.12 (Otimizador Local de Pareto) *Seja o problema de minimização dado pela Equação (14) e um escalar $\delta \in \mathbb{R}$, um vetor de variáveis de decisão $\bar{x} \in \Omega$ é dito um otimizador local de Pareto ou minimizador local de Pareto se, e somente se, não existir um vetor $x \in \Omega'$, tal que $x \prec \bar{x}$, onde $\Omega' = \{x \in \Omega \mid \|f(x) - f(\bar{x})\| < \delta\}$, sendo $\delta > 0$.*

O estabelecido pela Definição 1.3.12 equivale a dizer que um vetor $\bar{x} \in \Omega$ é um otimizador local de Pareto se não existir um vetor x que domine \bar{x} , tal que $f(x)$ esteja contida em uma bola aberta, de raio δ , centrada em $f(\bar{x})$, onde $\delta > 0$, isto é, $f(x) \in B(f(\bar{x}), \delta)$, onde $B(f(\bar{x}), \delta) = \{f(x) \in \mathbb{R}^m \mid \|f(x) - f(\bar{x})\| < \delta\}$.

1.3.4.3.10 Ótimo Local de Pareto

Um ótimo local de Pareto é um elemento do conjunto objetivo cujas componentes são todas iguais ou melhores que as dos demais elementos em sua vizinhança, sendo pelo menos uma estritamente melhor. Trata-se de um dos melhores resultados do problema de otimização multiobjetivo em sua vizinhança. A Definição 1.3.13 apresenta rigorosamente o conceito de ótimo local de Pareto.

Definição 1.3.13 (Ótimo local de Pareto) *Seja o problema de minimização dado pela Equação (14), seu conjunto objetivo \mathcal{U} e o escalar $\delta \in \mathbb{R}$, um vetor de objetivos $\bar{o} \in \mathcal{U}$ é dito um ótimo local de Pareto ou mínimo local de Pareto se não existir um vetor $o \in \mathcal{U}'$ tal*

que a Equação (31a) e a Equação (31b) sejam satisfeitas, onde $\mathcal{U} = \{o \in \mathcal{U} \mid \|o - \bar{o}\| < \delta\}$, sendo $\delta > 0$.

$$\forall l \in \{1, 2, 3, \dots, m\} : o_l \leq \bar{o}_l \quad (31a)$$

$$\exists l \in \{1, 2, 3, \dots, m\} : o_l < \bar{o}_l \quad (31b)$$

1.3.4.3.11 Conjunto Ótimo Local de Pareto

Um conjunto ótimo local de Pareto é constituído por todos os otimizadores locais de Pareto correspondentes aos ótimos locais de Pareto contidos em uma determinada região do conjunto objetivo. Este conjunto possui as soluções que correspondem aos melhores resultados presentes em uma dada região do conjunto objetivo, dessa forma, considerada esta região, estas soluções não são dominadas por nenhuma outra. A Definição 1.3.14 determina estritamente o conceito de conjunto ótimo local de Pareto.

Definição 1.3.14 (Conjunto Ótimo Local de Pareto) *Seja o problema de minimização dado pela Equação (14) e o escalar $\delta \in \mathbb{R}$, um conjunto $\Theta' \subset \Omega$ é dito conjunto ótimo local de Pareto se $\Theta' \neq \emptyset$ e se a Equação (32) for satisfeita, onde o conjunto $\Omega' = \{x \in \Omega \mid \|f(x) - f(\bar{x})\| < \delta\}$, sendo $\delta > 0$.*

$$\forall \bar{x} \in \Theta', \nexists x \in \Omega' : x \prec \bar{x} \quad (32)$$

1.3.4.3.12 Fronteira Local de Pareto

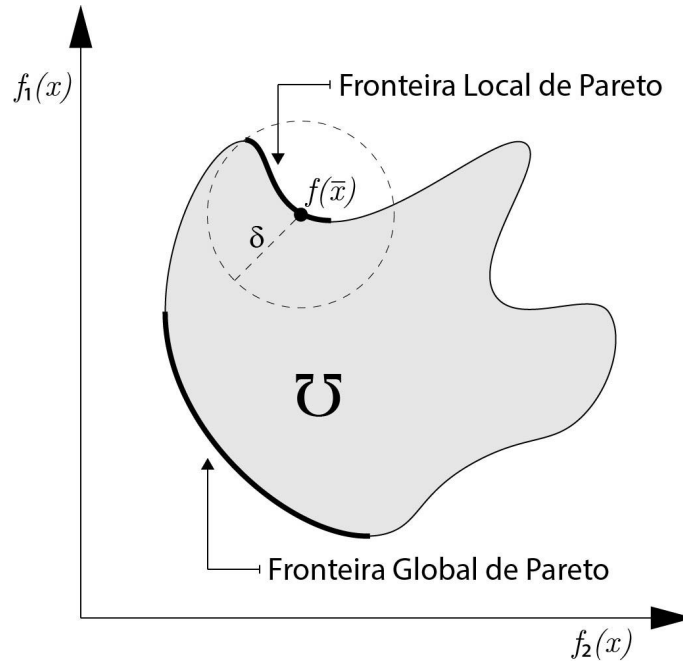
Uma fronteira local de Pareto consiste de todos os ótimos locais de Pareto correspondentes aos otimizadores de Pareto pertencentes a um dado conjunto ótimo local de Pareto, contendo assim todos os melhores resultados presentes em uma dada região do conjunto objetivo. A Definição 1.3.15 expressa categoricamente o conceito de fronteira local de Pareto.

Definição 1.3.15 (Fronteira Local de Pareto) *Seja o problema de minimização dado pela Equação (14) e um conjunto ótimo local de Pareto Θ' , denomina-se fronteira local de Pareto \mathcal{F}' , o conjunto dado pela Equação (33).*

$$\mathcal{F}' = \{f(\bar{x}) \in \mathbb{R}^m \mid \bar{x} \in \Theta'\} \quad (33)$$

A Figura 8 ilustra uma fronteira local de Pareto de um problema de otimização com dois objetivos.

Figura 8 - Fronteira local de Pareto.



Legenda: Fronteira local de Pareto em uma bola aberta centrada em $f(\bar{x})$ com raio δ do conjunto viável \bar{U} em contraste com a fronteira global de Pareto, referente ao problema de otimização $\underset{x}{\text{Minimizar}} f_1(x), f_2(x)$ Sujeito a $x \in \Omega$.

Fonte: O autor, 2023.

1.3.4.3.13 Conjunto Solução Local de um Problema de Otimização Multiobjetivo

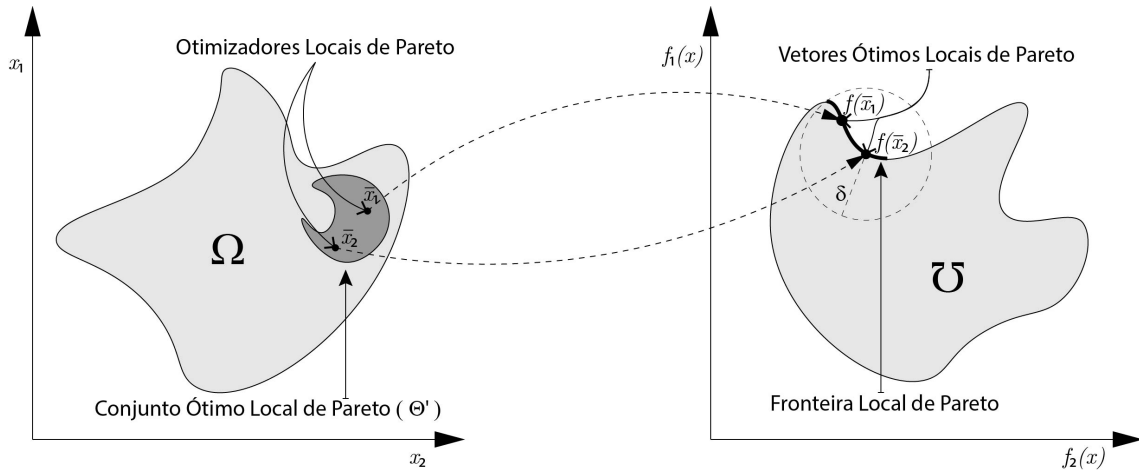
Uma vez definidos localmente todos os conceitos necessários, é possível finalmente, estabelecer o conceito de conjunto solução local de um problema de otimização multiobjetivo. A Definição 1.3.16 enuncia formalmente o conceito de solução local de um problema de otimização multiobjetivo, que se trata do de um conjunto ótimo local de Pareto do problema em questão, isto é, os vetores que remetem aos melhores resultados de uma dada região do conjunto objetivo.

Definição 1.3.16 (Conjunto Solução Local de um Problema de Otimização Multiobjetivo)

Seja o problema de minimização dado pela Equação (14), um conjunto solução local para este problema consiste em um conjunto ótimo local de Pareto $\Theta' \subset \Omega$ do problema de otimização.

A Figura 9 demonstra os conceitos que envolvem o conjunto solução local de um problema de otimização de duas variáveis e dois objetivos.

Figura 9 - Conjunto solução local de um problema de otimização multiobjetivo e conceitos correlatos.



Legenda: Relação entre um conjunto ótimo local de Pareto e a fronteira local de Pareto correspondente a ele, ambos do problema de otimização Minimizar $f_1(x_1, x_2), f_2(x_1, x_2)$ Sujeito a $x_1, x_2 \in \Omega$, realçados os otimizadores locais de Pareto e ótimos locais de Pareto, considerando o conjunto viável Ω e o conjunto objetivo \mathcal{U} .

Fonte: O autor, 2023.

1.3.4.4 Qualidade do Conjunto Solução

Conforme visto anteriormente, um problema de otimização multiobjetivo não possui apenas uma única solução, mas sim, um conjunto de soluções ótimas. Esse fato, levanta a questão acerca de qual das soluções presentes no conjunto solução deve ser adotada.

Essa questão não pode ser respondida de forma generalizada, pois a escolha da solução vai variar de caso para caso, sendo muitas vezes subjetiva ou levando em consideração outros aspectos do problema. Dessa forma, escolher uma solução implica em conhecer informações específicas do problema em particular, que permitam determinar a solução a ser adotada.

Na falta das referidas informações, todas as soluções do conjunto solução do problema são tratadas com igual relevância. Nesse contexto, é desejável encontrar o maior número possível de soluções ótimas, distribuídas da melhor forma possível.

Segundo Deb (2001), existem duas metas elementares para o processo de otimização multiobjetivo, através das quais é possível avaliar a qualidade do conjunto solução de um problema de otimização multiobjetivo, a saber, a convergência e a diversidade (KHARE, 2002; LI *et al.*, 2012).

1.3.4.4.1 Convergência

A convergência indica o quão próximos os valores ótimos correspondentes às soluções estão da fronteira de Pareto. Quanto mais próximos eles estiverem da fronteira de Pareto, maior a convergência e quanto mais distantes da frente de Pareto, maior a divergência. Idealmente, todos os valores ótimos devem pertencer à fronteira de Pareto.

A convergência é uma exigência presente em qualquer problema de otimização, inclusive em outros tipos de problemas, sendo definida especificamente para cada tipo de problema. Soluções que apresentam baixa convergência, na maioria das vezes, são pouco relevantes ou mesmo irrelevantes para os problemas considerados.

Dessa forma, buscamos sempre soluções que apresentem a maior convergência possível, considerando as metodologias e recursos empregados para a obtenção das mesmas.

1.3.4.4.2 Diversidade

A diversidade indica o quão bem distribuídos os valores ótimos estão na fronteira de Pareto com relação à extensão e uniformidade. Quanto mais esparsamente uniforme estiverem dispostas as soluções na fronteira de Pareto, maior a diversidade, quanto mais segregadas e assimétricas se mostrarem as soluções sobre a fronteira de Pareto, menor a diversidade. Idealmente, os valores ótimos devem cobrir integralmente a fronteira de Pareto e estarem uniformemente distribuídos sobre ela.

A diversidade é uma exigência específica dos problemas de otimização com vários objetivos. Soluções pouco diversificadas podem inviabilizar a escolha de uma delas na fase posterior à resolução do problema multiobjetivo, considerando os aspectos particulares do caso em questão, tornando as soluções obtidas inviáveis para aplicação.

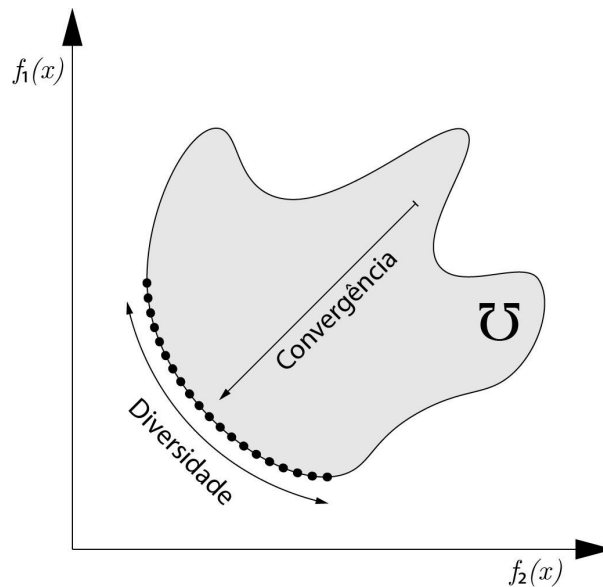
Desse modo, procuramos sempre soluções que apresentem a maior diversidade possível, levando em conta os métodos e recursos usados na obtenção das mesmas.

1.3.4.4.3 Conflito entre Convergência e Diversidade

Idealmente, um conjunto solução deve apresentar máxima convergência e máxima diversidade. Quando isso ocorre, o conjunto de soluções possui a mais alta qualidade. A Figura 10 exemplifica um conjunto de soluções que apresentam a melhor convergência e diversidade possíveis.

No entanto, nem sempre é possível obter um conjunto de soluções com as características supracitadas. Algumas soluções possuem uma convergência muito boa, mas em contrapartida, possuem uma menor diversidade, enquanto outras, possuem menor con-

Figura 10 - Convergência e diversidade.



Legenda: Melhor convergência e diversidade possíveis dos pontos ótimos no problema de otimização
 Minimizar $f_1(x), f_2(x)$ Sujeito a $x \in \Omega$, considerando o conjunto objetivo \mathcal{U} .

Fonte: O autor, 2023.

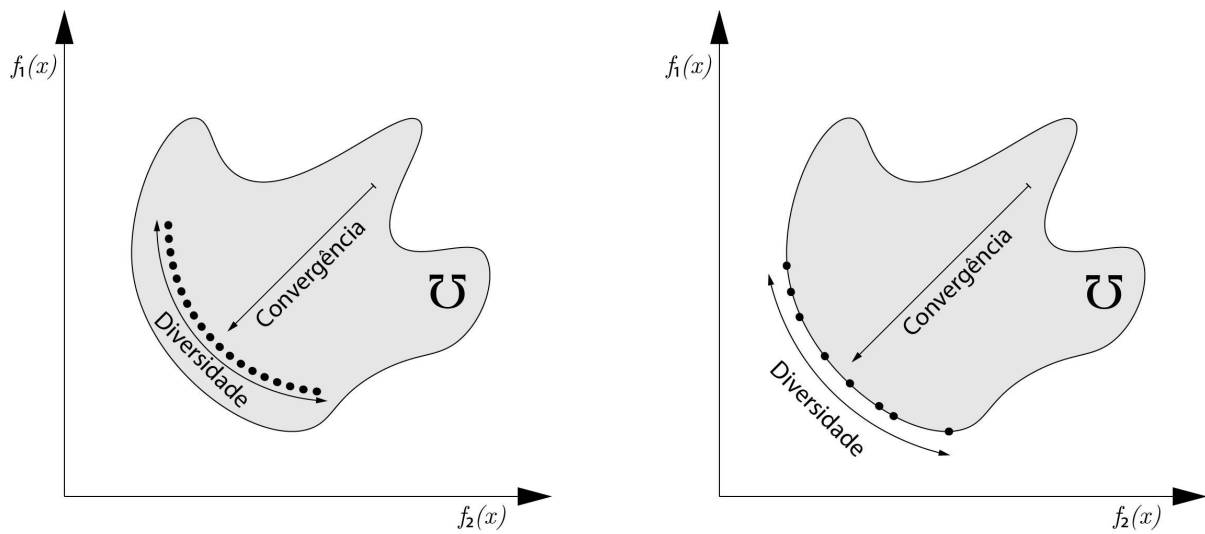
vergência, mas uma diversidade excelente. A Figura 11 ilustra cada um destes casos. Podemos ainda, encontrar conjuntos solução que apresentam tanto convergência, quanto diversidade mais reduzidas

Nestas situações, se faz necessário, escolher um conjunto solução que possua um bom equilíbrio entre convergência e diversidade, exceto nos casos onde se possui informações específicas do caso particular, que mostrem ser vantajoso escolher um conjunto solução que apresente um maior grau de um destes quesitos em detrimento do outro (LAUMANNNS *et al.*, 2002).

1.4 Referências com Informações Complementares

Este capítulo promoveu inicialmente uma introdução a otimização, apresentando a ideia, conceitos e características elementares de um problema de otimização conceitual. Em seguida, foram enunciados os conceitos e características da otimização mono-objetivo e da otimização multiobjetivo, que são muito relevantes para o teor desta tese. Entretanto, exemplificações, casos clássicos e conceitos que não são diretamente relacionados a este trabalho não foram mostrados aqui.

Figura 11 - Conflito entre convergência e diversidade.



(a) Maior diversidade e menor convergência.

(b) Menor diversidade e maior convergência.

Legenda: Considerando o problema de otimização Minimizar $f_1(x), f_2(x)$ Sujeito a $x \in \Omega$ e o conjunto objetivo \bar{U} , a relação mostrada em (a) apresenta uma boa diversidade, tendo uma boa quantidade, esparsidade e uniformidade dos pontos ótimos, mas uma convergência menor, uma vez que estes pontos estão afastados da fronteira de Pareto. Por outro lado, a relação apresentada em (b) apresenta boa convergência, tendo os pontos ótimos sobre a fronteira de Pareto, mas uma diversidade inferior, apresentando uma menor quantidade de pontos ótimos distribuídos de maneira bastante irregular.

Fonte: O autor, 2023.

Aplicações, casos de uso, exemplos e conceitos como robustez, eficiência e precisão, condições necessárias e suficientes em forma primal, relações entre direções viáveis, relações entre direções de Bouligand, direções de descida e problemas de otimização convexa não aparecem neste capítulo.

Existindo interesse em se aprofundar na otimização, a fim de conhecer os assuntos supracitados e outros, ficam recomendadas as seguintes obras e trabalhos da literatura:

- Brasil e Silva (2019);
- Gaspar-Cunha, Takahashi e Antunes (2012);
- Takahashi (2007);
- Ferraz (2021);
- Andretta (2010c);
- Ribeiro e Karas (2010);
- Andretta (2010a);
- Andretta (2010b);
- Andretta (2016);
- Izmailov e Solodov (2014);
- Souza e Ehrhardt (2011);
- Haeser (2015);
- Sabioni (2017);
- Weck (2004);
- Collette e Siarry (2003);
- Abraham, Jain e Goldberg (2005);
- Laumanns *et al.* (2002);

2 ALGORITMOS EVOLUCIONÁRIOS

O que sabemos é uma gota; o que ignoramos é um oceano.

Isaac Newton

Os métodos metaheurísticos de otimização, que serão usados nesta tese, consistem em algoritmos evolucionários com características e funcionamento particulares voltados à otimização. Assim, conhecer os algoritmos evolucionários, que serão abordados neste capítulo, é importante para entender os algoritmos de otimização mencionados.

Os Algoritmos Evolucionários ou Algoritmos Evolutivos são um ramo da computação evolucionária ou computação evolutiva que consiste em uma família de algoritmos heurísticos inspirados na evolução natural ou no comportamento biológico aplicados na resolução de situações modeladas como problemas (FULCHER, 2008; CASTRO, 2006).

Assim, os algoritmos evolucionários, conforme sua própria denominação sugere, são algoritmos inspirados nos princípios e conceitos da evolução da espécie proposta por Darwin (1959).

As seções que se sucedem apresentam as estruturas e operadores que constituem as partes elementares dos algoritmos evolucionários, apresenta-os em seguida.

2.1 Modelagem Computacional dos Conceitos Evolutivos

Entender como os conceitos evolucionários reais são modelados e implementados computacionalmente é sumamente importante para compreender os operadores e os algoritmos evolucionários, que serão abordados nas próximas seções. Assim, conhecer o assunto é fundamental para a compreensão destas, que utilizarão seus conceitos a todo momento.

Muitas áreas da computação foram inspiradas na evolução natural das espécies, contendo em suas técnicas e teorias, conceitos inerentes a esta área. Naturalmente, esses conceitos não estão presentes nas áreas inspiradas na evolução em sua forma real, tal como se dá na natureza. Eles foram simplificados e modelados computacionalmente, de forma a funcionarem e cumprirem seus propósitos no meio computacional.

Por esta razão, nesta seção, esses conceitos evolutivos serão contextualizados computacionalmente, descrevendo sinteticamente como eles são estruturados, representados e implementados no ambiente computacional.

2.1.1 Cromossomo

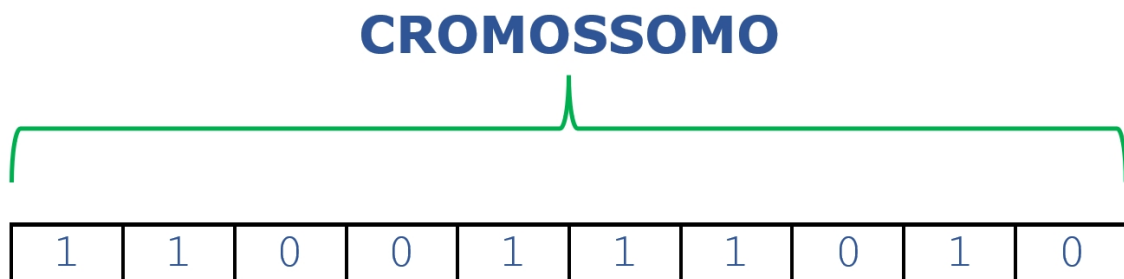
Os cromossomos são as estruturas fundamentais que carregam as características do indivíduo. Ele representa um indivíduo ou solução candidata.

Nota ao Leitor 9 *Devido ao cenário de problema, bem como a representatividade mútua, os termos cromossomo, indivíduo e solução são considerados sinônimos, tratando-se do mesmo conceito, neste contexto.*

Um cromossomo é codificado computacionalmente por uma estrutura de dados de lista ou por um vetor. O tipo de dado depende da codificação do indivíduo usada no problema em questão (LINDEN, 2008).

A Figura 12 ilustra um cromossomo codificado computacionalmente.

Figura 12 - Cromossomo codificado computacionalmente.



Fonte: O autor, 2023.

2.1.2 Gene

Os genes são as estruturas que carregam, unitariamente, o valor de uma determinada característica do cromossomo.

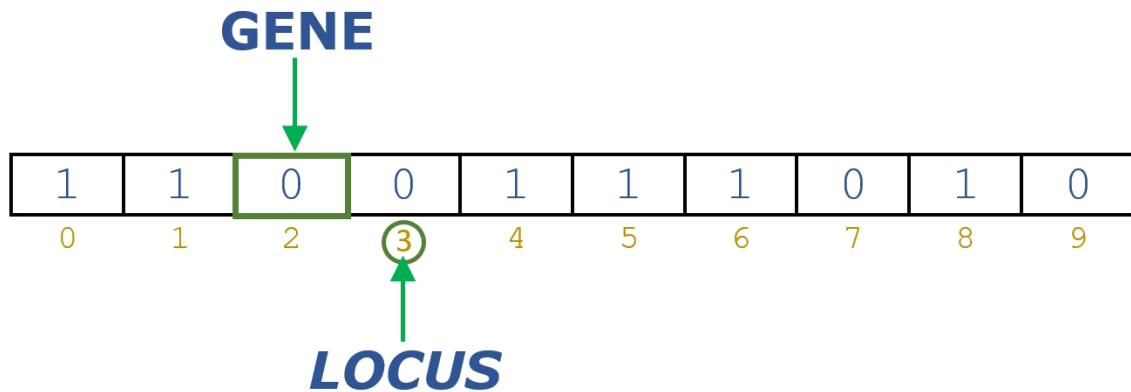
Computacionalmente, o gene consiste em um dos espaços do vetor ou lista que configura o cromossomo. Logo, cada espaço do cromossomo se trata de um gene distinto do mesmo. Uma outra denominação inerente aos genes é a de *locus*, que se trata da posição do gene no cromossomo (COELLO; VELDHUIZEN; LAMONT, 2002).

A Figura 13 indica um gene e um *locus* em um cromossomo codificado.

2.1.3 Alelo

Os alelos consistem no valor da característica presente em um gene de um cromossomo. Cada diferente valor que um gene pode carregar é um alelo distinto.

Figura 13 - Gene e *locus* em um cromossomo codificado computacionalmente.

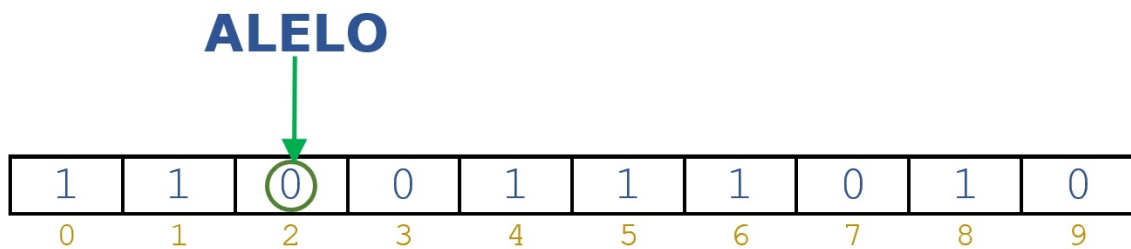


Fonte: O autor, 2023.

O alelo, computacionalmente, se traduz no valor que está contido no espaço do vetor ou lista, que é o gene do cromossomo.

A Figura 14 indica um alelo em um cromossomo codificado.

Figura 14 - Alelo em um cromossomo codificado computacionalmente.



Fonte: O autor, 2023.

2.1.4 Codificação

A codificação ou representação de um indivíduo, cromossomo ou solução candidata é a primeira decisão a ser tomada no processo de desenvolvimento de um algoritmo evolucionário, devendo também ser levada em consideração num eventual processo de escolha de um algoritmo evolucionário existente.

Outra questão importante é a definição do tamanho do cromossomo, consistindo em determinar o tamanho do arranjo ou lista que o representa. Essa decisão deve levar em consideração o tipo de codificação que será utilizada.

2.1.4.1 Codificação Binária

Uma das formas de codificação mais simples é a Representação Binária. Neste tipo de codificação, cada gene pode assumir apenas o alelo zero ou um. De forma que todo o cromossomo será composto apenas destes valores.

Embora traga simplicidade e facilidade na implementação, a representação binária pode apresentar pouco poder de expressividade em problemas do mundo real. Isso pode fazer com que ela seja pouco eficiente na tarefa de representar as possíveis soluções para tais problemas (DEB, 2001).

2.1.4.2 Codificação em Ponto Flutuante

Um outro tipo de codificação existente é a Representação em Ponto Flutuante, também conhecida como Representação Real. Nesta forma de representação, cada gene pode assumir qualquer alelo real e conseqüentemente cada cromossomo será um arranjo ou lista de pontos flutuantes.

A representação em ponto flutuante tem maior adoção pela comunidade, especialmente em problemas em que seja interessante ter uma distribuição dos genes em intervalo contínuo, ao invés de valores discretos (EIBEN; SMITH, 2003).

2.1.5 População

A população é o conjunto de indivíduos sobre os quais atuam todos os processos da evolução natural. Nos algoritmos evolucionários, a população tende a manter o seu tamanho em virtude do processo de seleção.

Computacionalmente, uma população é modelada como um arranjo de arranjos unidimensionais, isto é, um vetor de vetores, onde estes últimos configuram os cromossomos, ou como um arranjo bidimensional, ou seja, uma matriz, onde cada linha consiste em um cromossomo distinto desta população. Também pode-se modelar uma população como uma lista de listas, em que o funcionamento é análogo aos arranjos de arranjos.

A Figura 15 mostra uma população de indivíduos computacionalmente codificada.

2.1.6 Aptidão ou *Fitness*

O valor de Aptidão ou valor de *Fitness* de um indivíduo é um valor que quantifica o quão boa é a solução representada por ele. Através deste valor que os algoritmos

Figura 15 - População de cromossomos codificados computacionalmente.

POPULAÇÃO	0	1	1	0	0	0	1	0	0	0	←CROMOSSOMO
	1	1	0	1	1	1	1	1	1	0	←CROMOSSOMO
	1	0	0	1	0	0	0	0	0	1	←CROMOSSOMO
	0	0	0	0	0	1	0	1	1	1	←CROMOSSOMO
	0	1	0	0	1	1	1	0	1	0	←CROMOSSOMO
	0	1	2	3	4	5	6	7	8	9	

Fonte: O autor, 2023.

comparam as soluções candidatas que compõem uma população, podendo determinar assim quais as melhores e piores soluções presentes.

O valor de aptidão é comumente o resultado fornecido por uma função denominada Função de Aptidão ou de *Fitness*. A implementação dessa função é muito particular, variando consideravelmente de problema para problema, assim como a lista de argumentos que a mesma utilizará.

Nos problemas de otimização, por exemplo, a função de *fitness* pode ser o custo da solução. De forma que, as soluções de menor custo sejam as que terão maior valor de fitness, no caso de um problema de minimização.

2.1.7 Progenitor

Os indivíduos progenitores ou pais são aqueles que pertencem à geração atual, a partir dos quais novos indivíduos serão gerados através do processo reprodutivo. Eles são quem fornecem a base para os cromossomos da próxima geração.

2.1.8 Reprodução

A reprodução é o processo que visa gerar novos indivíduos a partir dos indivíduos existentes na geração atual. A ideia por trás do processo de reprodução é gerar indivíduos potencialmente melhores que seus pais.

O processo se dá por meio dos operadores que atuam sobre os cromossomos, denominados de operadores de reprodução, sendo eles, os operadores de mutação e de cruzamento, que serão detalhados na Seção 2.2.2 e na Seção 2.2.3, respectivamente.. Estes operadores têm o papel de realizar o processo de reprodução analogamente aos mecanismos da reprodução natural.

2.1.9 Descendentes

Os indivíduos descendentes ou filhos são aqueles que são gerados através dos operadores de reprodução após atuarem completamente sobre os indivíduos pais. Estes são os indivíduos que passarão pela seleção para se tornarem os pais da próxima geração.

2.1.10 Geração

A geração remete à época em que um determinado conjunto de indivíduos foi gerado. Computacionalmente, a geração remete a uma das iterações do algoritmo evolucionário em questão. Dessa forma, na n -ésima iteração do algoritmo evolucionário, teremos a n -ésima geração dos indivíduos.

2.1.11 Seleção

A seleção é o processo responsável por escolher os indivíduos que permanecerão na próxima geração, com intuito de refinar as soluções até atingir as que sejam mais adequadas para o problema em questão.

Os indivíduos da população da geração atual passam pelo processo de reprodução, gerando descendentes. Consequentemente, o tamanho da população cresce, pois além dos indivíduos da população atual, existem também os descendentes. Para manter o número de indivíduos controlado, o mecanismo de seleção atua de modo a eliminar os indivíduos que sejam menos adequados.

O conceito de seleção é o expresso na Definição 2.1.1.

Definição 2.1.1 (Seleção) *Seja uma população inicial de n indivíduos e uma prole de m indivíduos, totalizando uma população de $n + m$ indivíduos, para a próxima geração, são selecionados n indivíduos desta população total.*

Embora após a seleção, o tamanho da população voltará a ser de n indivíduos, o tamanho da prole m pode ser maior ou menor que n .

Cada algoritmo evolucionário implementa esse mecanismo de seleção de forma particular, utilizando uma estratégia própria.

2.1.12 Máquinas de Estados Finitos

Embora não sejam próprias da computação evolucionária, as máquinas de estados finitos são um conceito de compreensão necessária para o entendimento de alguns tipos de algoritmos evolucionários que serão apresentados nesta tese.

Uma Máquina de Estados Finitos, Autômato de Estados Finitos, Autômato Finito ou simplesmente, Máquina de Estados é um modelo matemático e computacional usado especialmente para representar softwares e circuitos. Trata-se de uma máquina abstrata que em um determinado instante, pode estar em apenas um dentre um número finito de estados definidos. Esta máquina pode mudar de um estado para outro em resposta a um determinado conjunto de entradas. Essa mudança é denominada transição de estados (BACK; FOGEL; MICHALEWICZ, 2000; WANG; TEPFENHART, 2019).

Uma máquina de estados é definida pela lista de seus estados, seu estado inicial, seus estados finais (opcionalmente), e as entradas que configuram os gatilhos para as transições de estados.

2.1.12.1 Formalismo Matemático

O conceito formal de máquina de estados finitos está expresso na Definição 2.1.2.

Definição 2.1.2 (Máquina de Estados Finitos) *Uma máquina de estados finitos ou aceitador de estados finitos é uma quintupla $(\sigma, S, s_0, \delta, F)$, onde σ é o alfabeto ou conjunto de entradas, S é o conjunto finito não vazio de estados possíveis, s_0 é o estado inicial, sendo este um elemento de S , δ é a função de transição de estados, de forma $\delta : S \times \sigma \rightarrow S$ e F é o conjunto finito de estados finais, que pode ser vazio, e deve ser um subconjunto de S .*

Convencionalmente, δ é uma função parcial de duas variáveis, na forma $\delta(s, \varsigma)$, onde s é um estado pertencente a S e ς é uma entrada pertencente a σ . A parcialidade de δ advém do fato de que dado um certo estado s , geralmente, não existe uma transição deste estado para outro definida para todas as entradas existentes em σ .

2.1.12.2 Funcionamento das Máquinas de Estados

Diversos dispositivos presentes no cotidiano das pessoas podem ser modelados como máquinas de estados, uma vez que estes seguem uma sequência predeterminada de ações, conforme uma série de eventos disparados. Alguns exemplos de dispositivos com comportamento de máquina de estados são semáforos, elevadores e máquinas de venda.

Um exemplo clássico de equipamento modelável por uma máquina de estados finitos é uma catraca de acesso. Esse equipamento é muito utilizado em algumas rodoviárias e em estações de trem e metrô para controlar o acesso de pessoas e garantir o pagamento das tarifas cobradas pelas concessionárias.

A catraca de acesso é uma espécie de portão de três braços giratórios. Inicialmente, os braços da catraca ficam travados, impedindo o acesso dos clientes. Caso o cliente tente girar os braços, estes permanecerão travados, bloqueando sua passagem. Ao inserir o cartão na catraca, os braços são desbloqueados e permitem um único giro, concedendo passagem a um cliente. Após o cliente girar os braços e passar, estes são novamente travados até que um novo cartão seja inserido.

Considerando uma catraca com o comportamento acima descrito, podemos modelá-la como uma máquina de dois estados possíveis, bloqueado e liberado. Existem dois tipos de entradas que promovem a transição de estados, girar os braços e inserir o cartão.

Estando no estado bloqueado, girar os braços manterá a catraca neste estado, ao passo que, inserir o cartão, fará a transição da catraca para o estado liberado. Encontrando-se no estado liberado, tentar inserir outro cartão manterá a catraca no estado liberado, uma vez que ela não aceitará outro cartão até estar bloqueada. No entanto, girar os braços da catraca fará a transição dela para o estado bloqueado.

2.1.12.3 Representação Estruturada de Máquinas de Estados

Embora, uma máquina de estados possa ser descrita por um texto como o apresentado acima, a linguagem natural pode trazer ambiguidades e requerer muito esforço para descrever máquinas de estado mais complexas. Devido a estes inconvenientes, existem formas estruturadas para descrever as máquinas de estados. As formas mais populares são as Tabelas de Transição de Estados e os Diagramas de Estados (HOPCROFT; MOTWANI; ULLMAN, 2008).

A Tabela de Transição de Estados apresenta de forma tabular, os estados possíveis, suas transições e as entradas que disparam as transições. O Quadro 1 apresenta a tabela de transição de estados da máquina de estados que modela a catraca descrita anteriormente.

Quadro 1 - Tabela de transição de estados do modelo da catraca.

Estado Atual	Entrada(s)	Estado Seguinte	Descrição
Bloqueado	Cartão Inserido	Liberado	Os braços são destravados.
Liberado	Braços Girados	Bloqueado	Os braços são travados.

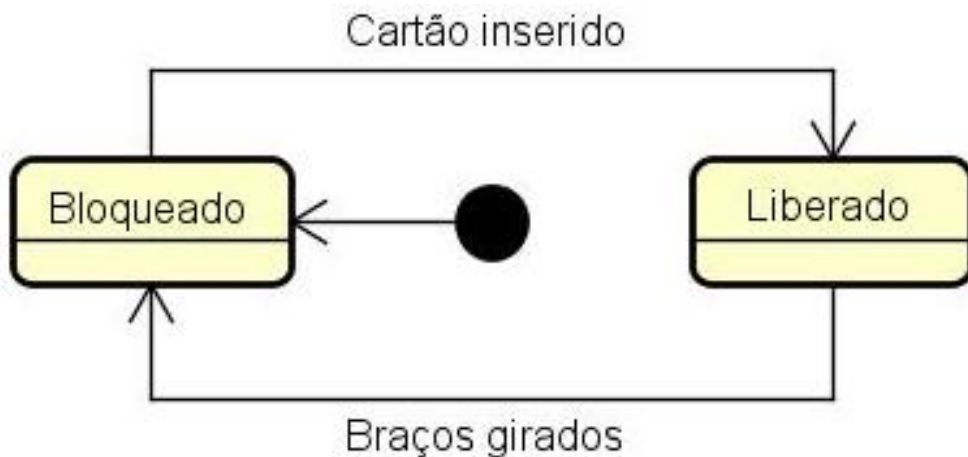
Fonte: O autor, 2023.

O Diagrama de Estados apresenta de forma análoga a um fluxograma, os estados,

suas transições e as entradas que promovem-nas. Existem diversas notações usadas para elaborar um diagrama de estados, mas a notação com maior poder de expressividade e popularidade, atualmente, é a da *Unified Modeling Language* [UML] (HOPCROFT; MOTWANI; ULLMAN, 2008).

A Figura 16 mostra o diagrama de estados referente à catraca descrita anteriormente, onde os retângulos ilustram os estados, as setas representam as transições e os rótulos sobre as setas indicam as entradas que disparam a transição denotada pela seta.

Figura 16 - Diagrama de estados do modelo da catraca.



Fonte: O autor, 2023.

2.2 Operadores Evolucionários de Reprodução

Os operadores evolutivos de reprodução são as principais partes constituintes dos algoritmos evolucionários, sendo imprescindível a sua compreensão para o pleno entendimento do funcionamento e das características destes algoritmos, que serão apresentados na próxima seção.

A reprodução é um processo de vital importância para a evolução, sendo um dos seus principais processos. Ela é a responsável pela geração dos novos indivíduos e pela formação de suas características. De modo que sem ela, as novas gerações inexistiriam, estando a espécie fadada a extinção.

Com os modelos computacionais inspirados na evolução não é diferente, a reprodução tem papel fundamental em seu funcionamento e por isso, estão presentes em todos, sem exceção. Chamamos as formas de reprodução dos indivíduos no ambiente computacional de operadores de reprodução. São eles que serão apresentados e detalhados nesta seção.

2.2.1 Tipos de Reprodução

Existem dois tipos elementares de reprodução que promovem parte da evolução dos indivíduos dentro de uma população que está em processo evolutivo, a saber, a reprodução assexuada e reprodução sexuada.

A **reprodução assexuada** consiste na geração de um descendente a partir de um único indivíduo através do processo de mutação. A **reprodução sexuada**, por sua vez, consiste na participação de pelo menos dois indivíduos para gerar um ou mais descendentes a partir do processo de cruzamento ou recombinação.

As seções a seguir, apresentam os principais operadores de reprodução e suas variantes mais populares que implementam computacionalmente os processos de mutação e cruzamento nos algoritmos evolucionários.

2.2.2 Operador de Mutação

O operador de mutação altera aleatoriamente um ou mais genes de um cromossomo, tornando-o diferente de sua configuração original.

Normalmente, durante o processo de reprodução assexuada, um indivíduo gera uma cópia fiel de si mesmo como descendente, que por sua vez, é submetido ao operador de mutação, sofrendo alterações em seus genes.

A probabilidade de uma mutação ocorrer em um gene de um indivíduo é denominada **taxa de mutação**. Usualmente, se usa um valor percentual bem pequeno para a taxa de mutação, uma vez que esse operador pode gerar um indivíduo potencialmente pior que seu progenitor.

2.2.2.1 Mutação na Representação Binária

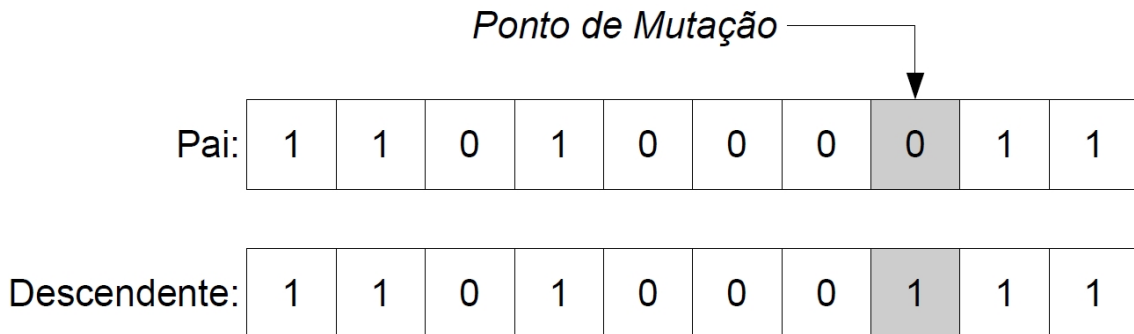
No caso de um indivíduo representado em codificação binária, o operador predominantemente utilizado é o denominado Operador de Mutação Padrão (BACK; FOGEL; MICHALEWICZ, 2000; DEB, 2001).

2.2.2.1.1 Operador de Mutação Padrão

O Operador de Mutação Padrão, quando aplicado a um gene de um cromossomo, inverte o valor do mesmo. Por exemplo, caso o alelo de um gene for zero, após a mutação, seu alelo passará a ser o um e vice versa (GOLDBERG, 1989).

A Figura 17 ilustra a dinâmica de funcionamento do operador de mutação padrão atuando sobre um gene do cromossomo.

Figura 17 - Representação esquemática do funcionamento do operador de mutação padrão.



Fonte: Adaptado de Eiben e Smith, 2003.

2.2.2.2 Mutação na Representação em Ponto Flutuante

Considerando a representação dos indivíduos por codificação em ponto flutuante, os operadores de mutação mais usados são o Operador de Mutação Uniforme e o Operador de Mutação Gaussiana (BACK; FOGEL; MICHALEWICZ, 2000; DEB, 2001).

2.2.2.2.1 Operador de Mutação Uniforme

O operador de mutação uniforme escolhe aleatoriamente um dos genes do cromossomo $x = [x_1, \dots, x_k, \dots, x_n]$, em que $n \in \mathbb{N}$ é a quantidade de genes presente no indivíduo e $k \in \{1, 2, 3, \dots, n\}$ indica o gene escolhido aleatoriamente, e gera um descendente $x' = [x_1, \dots, x'_k, \dots, x_n]$, onde x'_k é um número aleatório com distribuição de probabilidade uniforme amostrado no intervalo $[L_I, L_S]$, sendo L_I e L_S , respectivamente, os limites inferior e superior para o valor do alelo x_k .

2.2.2.2.2 Operador de Mutação Gaussiana

O Operador de Mutação Gaussiana, gera um descendente x' do cromossomo x , com todos os genes alterados segundo a expressão dada na Equação (34), onde $N(0, \sigma)$ é um vetor de variáveis aleatórias gaussianas independentes, com média zero e desvio padrão

σ .

$$x' = x + N(0, \sigma) \quad (34)$$

2.2.3 Operador de Cruzamento

O operador de cruzamento ou operador de recombinação realiza a troca e combinação dos alelos de dois ou mais cromossomos, gerando um novo indivíduo, sendo este o principal operador de reprodução dos algoritmos genéticos (GOLDBERG, 1989).

Neste operador, partes de um cromossomo são trocadas pelas suas equivalentes oriundas de outros cromossomos, gerando um novo indivíduo (cromossomo) a partir da combinação das características potencialmente melhores advindas de seus progenitores.

2.2.3.1 Cruzamento na Representação Binária

Em cromossomos codificados em binário, os operadores de recombinação mais populares são o Operador de Recombinação de Um Ponto, o Operador de Recombinação de Múltiplos Pontos e o Operador de Recombinação Uniforme (BACK; FOGEL; MICHALEWICZ, 2000; DEB, 2001).

2.2.3.2 Operador de Recombinação de Um Ponto

Um operador usual de cruzamento é o Operador de Recombinação de Um Ponto. Neste operador, define-se aleatoriamente um mesmo ponto de corte em dois cromossomos, dividindo-os em duas partes, cada um. Cada descendente é gerado através da combinação entre um dos lados do ponto de corte de um dos ascendentes e o outro lado do ponto de corte do outro ascendente.

Por exemplo, considerando dois indivíduos pais e um ponto de corte definido, o primeiro filho pode ser gerado pela junção da parte à esquerda do ponto de corte do primeiro pai com a parte à direita do ponto de corte do segundo pai. O segundo filho é gerado através da junção das partes não utilizadas, sendo estas, a parte à direita ao ponto de corte do primeiro pai e a parte à esquerda do ponto de corte do segundo pai.

A Figura 18 ilustra o funcionamento do operador de recombinação de um ponto.

Figura 18 - Representação esquemática do funcionamento do operador de recombinação de um ponto.



Fonte: Adaptado de Eiben e Smith, 2003.

2.2.3.3 Operador de Recombinação de Múltiplos Pontos

O Operador de Recombinação de Múltiplos Pontos é uma generalização do operador de recombinação de um ponto. Ao invés de um, ele estabelece aleatoriamente n pontos de corte nos dois cromossomos, dividindo-os em $n + 1$ partes, cada um. Cada filho é gerado por meio da mistura de partes do primeiro e do segundo pai.

A escolha de quais partes de que pais serão usadas para compor o novo filho não segue um modelo pré-determinado, podendo ser realizada de diversas formas, sendo uma forma comum a escolha aleatória das mesmas, respeitadas algumas restrições.

A Figura 19 mostra um exemplo do funcionamento do operador de recombinação de múltiplos pontos com dois pontos de cortes, onde o primeiro filho é gerado pela junção da parte situada entre os dois cortes do primeiro pai com a parte à esquerda do primeiro corte e da direita do segundo corte do segundo pai. O segundo filho é gerado pela junção da parte situada entre os dois cortes do segundo pai com a parte à esquerda do primeiro corte e da direita do segundo corte do primeiro pai.

Figura 19 - Representação esquemática do funcionamento do operador de recombinação de múltiplos pontos.



Legenda: Operador de recombinação de múltiplos pontos com $n = 2$.

Fonte: Adaptado de Eiben e Smith, 2003.

2.2.3.4 Operador de Recombinação Uniforme

O Operador de Recombinação Uniforme é outro tipo comum de cruzamento. Este operador considera para gerar os descendentes, não partes dos cromossomos pais, mas sim cada um de seus genes, de forma independente. Dessa forma, o filho herdará cada gene individualmente de um dos pais.

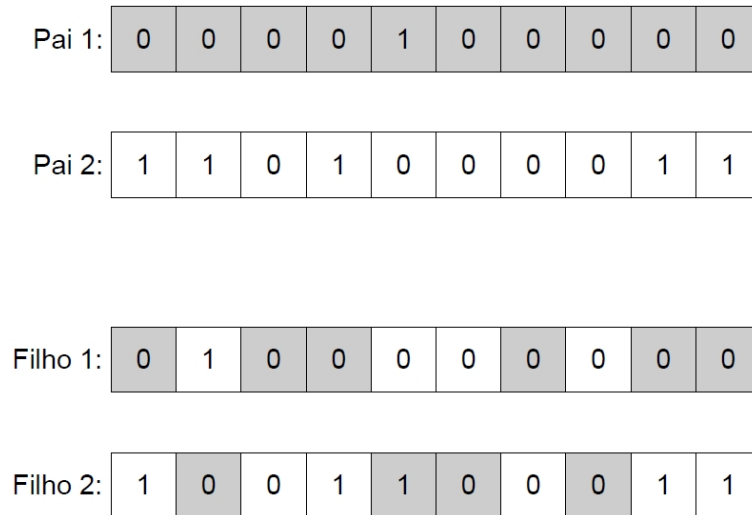
Para definir de qual dos pais, o primeiro descendente herdará um determinado gene, usa-se para cada gene, uma variável aleatória com distribuição de probabilidade uniforme entre zero e um. Caso o valor da variável for inferior a um determinado limiar l , que usualmente é fixado em 0,5, o gene será herdado do primeiro pai, caso seja superior a l , gene será oriundo do segundo pai. O segundo descendente é gerado a partir os genes não utilizados pelo primeiro, através do uso de mapeamento inverso.

A Figura 20 mostra o funcionamento do operador de recombinação uniforme.

2.2.3.5 Cruzamento na Representação em Ponto Flutuante

Para cromossomos representados em ponto flutuante, os operadores de recombinação mais comumente utilizados são o Operador de Recombinação Discreta e os Operadores

Figura 20 - Representação esquemática do funcionamento do operador de recombinação uniforme.



Fonte: Adaptado de Eiben e Smith, 2003.

de Recombinação Aritmética (BACK; FOGEL; MICHALEWICZ, 2000; DEB, 2001).

2.2.3.6 Recombinação Discreta

Os Operadores de Recombinação Discreta pertencem à categoria de operadores de recombinação voltada para problemas com indivíduos codificados em ponto flutuante.

Esses operadores trabalham de forma análoga aos operadores de recombinação para indivíduos com representação binária, tais como o operador de recombinação de um ponto, de múltiplos pontos e de recombinação uniforme. A diferença é que ao invés de vetores de binários, teremos vetores de pontos flutuantes (EIBEN; SMITH, 2003).

A desvantagem dos operadores de recombinação discreta é que não existe surgimento de novos valores nos genes indivíduo filho, uma vez que este está limitado a possuir os alelos presentes em seus pais, limitando a variabilidade genética do mesmo.

A inserção de novos valores, que não estejam nos pais, ficará a cargo apenas do operador de mutação. No entanto, comumente, as mutações possuem baixa probabilidade de ocorrência, significando que geralmente esses novos valores não surgirão no descendente.

2.2.3.7 Recombinação Aritmética

Os Operadores de Recombinação Aritmética ou de Recombinação Intermediária são outro grupo que pertence à categoria de operadores de recombinação voltada para problemas com indivíduos codificados em ponto flutuante. O princípio fundamental destes algoritmos foi proposto por Michalewicz e Janikow (1991).

A recombinação aritmética define os alelos dos descendentes através de valores intermediários aos equivalentes presentes nos pais. O descendente z é gerado através da combinação linear entre os dois cromossomos pais x e y definida pela Equação (35), onde α é um número pertencente ao intervalo $[0, 1]$, que geralmente tem um valor fixo predefinido, mas pode também ser um número aleatório.

$$z = \chi(x, y, \alpha) = x \cdot \alpha + y \cdot (1 - \alpha) \quad (35)$$

Usualmente, adota-se o valor 0,5 para α . Quando isso ocorre, denominamos a operação de **Recombinação Aritmética Uniforme**.

Esse tipo de operador tem aplicação interessante em problemas de otimização numérica com restrições, onde a região factível é convexa, uma vez que a combinação linear entre cromossomos x e y que pertencem à região factível resultará sempre em elementos que também pertencem a esta região.

2.2.3.7.1 Operador de Recombinação Aritmética Simples

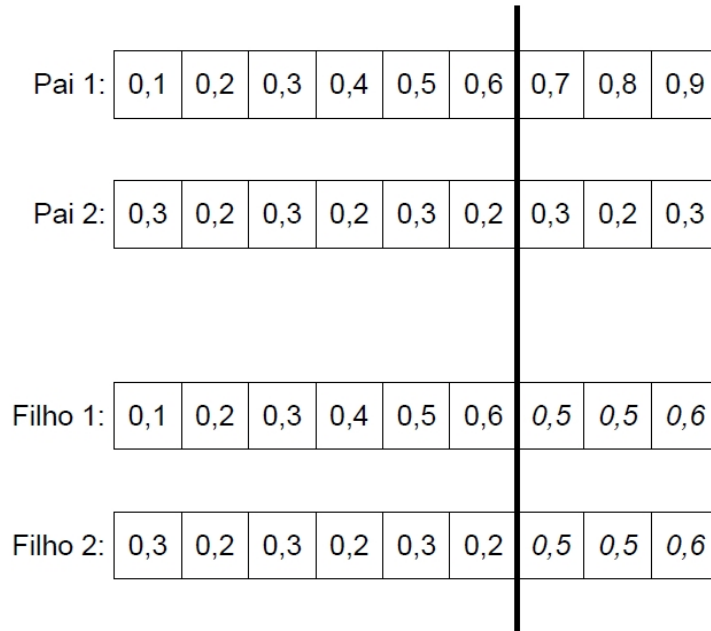
O Operador de Recombinação Aritmética Simples é um dos operadores de recombinação aritmética. Nele, define-se aleatoriamente um ponto de corte nos cromossomos pais e o primeiro descendente é gerado pela junção da parte à esquerda do corte, composta pelos k primeiros genes, do primeiro pai com a combinação linear da parte à direita, composta pelos $n - k$ últimos genes, do corte dos dois pais, onde n é o número total de genes do cromossomo. O segundo descendente é gerado pela junção da parte à esquerda do corte do segundo pai com a combinação linear da parte à direita do corte dos dois pais.

A Equação (36) define a lei de formação dos descendentes z e h considerando os pais x e y , onde \oplus é um operador de concatenação.

$$\begin{aligned} z &= [x_1, x_2, \dots, x_k] \oplus \chi([x_{k+1}, x_{k+2}, \dots, x_n], [y_{k+1}, y_{k+2}, \dots, y_n], \alpha) \\ h &= [y_1, y_2, \dots, y_k] \oplus \chi([y_{k+1}, y_{k+2}, \dots, y_n], [x_{k+1}, x_{k+2}, \dots, x_n], \alpha) \end{aligned} \quad (36)$$

A Figura 21 mostra a dinâmica deste operador realizando uma recombinação uniforme.

Figura 21 - Representação esquemática do funcionamento do operador de recombinação aritmética simples.



Legenda: Recombinação aritmética simples com $\alpha = 0,5$

Fonte: Adaptado de Eiben e Smith, 2003.

2.2.3.7.2 Operador de Recombinação Aritmética Individual

O Operador de Recombinação Aritmética Individual atua na mesma linha do Operador de Recombinação Aritmética Simples, mas neste caso, o primeiro descendente é uma cópia do primeiro pai, tendo apenas um único gene k escolhido aleatoriamente, cujo alelo será alterado para o valor da combinação linear entre os alelos de mesma posição de ambos os pais. O segundo descendente será uma cópia do segundo pai, tendo o gene da posição k com alelo alterado para o valor da combinação linear entre os alelos da mesma posição de ambos os pais.

A Equação (37) define a lei de formação dos descendentes z e h considerando os pais x e y .

$$\begin{aligned} z &= [x_1, x_2, \dots, x_{k-1}, \chi(x_k, y_k, \alpha), x_{k+1}, x_{k+2}, \dots, x_n] \\ h &= [y_1, y_2, \dots, y_{k-1}, \chi(y_k, x_k, \alpha), y_{k+1}, y_{k+2}, \dots, y_n] \end{aligned} \quad (37)$$

A Figura 22 ilustra o funcionamento deste operador efetuando uma recombinação uniforme.

Figura 22 - Representação esquemática do funcionamento do operador de recombinação aritmética individual.

Pai 1:	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
Pai 2:	0,3	0,2	0,3	0,2	0,3	0,2	0,3	0,2	0,3
Filho 1:	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,5	0,9
Filho 2:	0,3	0,2	0,3	0,2	0,3	0,2	0,3	0,5	0,3

Legenda: Recombinação aritmética individual com $\alpha = 0,5$

Fonte: Adaptado de Eiben e Smith, 2003.

2.2.3.7.3 Operador de Recombinação Aritmética Total

O Operador de Recombinação Aritmética Total tem um funcionamento próximo ao operador de recombinação aritmética individual. A diferença é que nele, todos os alelos, tanto do primeiro quanto do segundo descendente, resultam da combinação linear dos respectivos genes dos pais.

A Equação (38) define a lei de formação dos descendentes z e h considerando os pais x e y .

$$\begin{aligned} z &= \chi(x, y, \alpha) = x \cdot \alpha + y \cdot (1 - \alpha) \\ h &= \chi(y, x, \alpha) = y \cdot \alpha + x \cdot (1 - \alpha) \end{aligned} \quad (38)$$

A Figura 23 apresenta a dinâmica deste operador fazendo uma recombinação uniforme.

Este é um dos operadores mais utilizados pela comunidade (GABRIEL; DELBEM, 2008a).

Figura 23 - Representação esquemática do funcionamento do operador de recombinação aritmética total.

Pai 1:	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
Pai 2:	0,3	0,2	0,3	0,2	0,3	0,2	0,3	0,2	0,3
Filho 1:	0,2	0,2	0,3	0,3	0,4	0,4	0,5	0,5	0,6
Filho 2:	0,2	0,2	0,3	0,3	0,4	0,4	0,5	0,5	0,6

Legenda: Recombinação aritmética total com $\alpha = 0,5$

Fonte: Adaptado de Eiben e Smith, 2003.

2.2.3.8 Cruzamento com Múltiplos Pais

O conceito de cruzamento pode ser generalizado em estratégias que envolvam o uso não de apenas dois pais, mas de n pais, sendo $n > 2$. Esse esquema reprodutivo inexistente em sistemas biológicos reais de reprodução.

Esse conceito foi proposto na década de 1960, logo no início das pesquisas na área de algoritmos evolucionários. A ideia por trás dessa estratégia era que a presença de mais progenitores no processo de cruzamento amplificaria os efeitos da recombinação (EIBEN; KEMENADE; KOK, 1995).

Os operadores de cruzamento envolvendo múltiplos pais podem ser agrupados segundo sua estratégia. Conforme Eiben, Kemenade e Kok (1995), Eiben e Smith (2003), podemos classificar esses operadores em três grupos fundamentais, a saber, os Operadores de Eleição P-Sexual, os Operadores de Recombinação Diagonal e os Operadores de Recombinação Massiva.

2.2.3.8.1 Operadores de Eleição P-Sexual

O primeiro grupo é denominado Eleição p-Sexual. Os operadores deste grupo utilizam o cruzamento baseado na sequência de alelos, podendo ser compreendidos

como generalização do operador de recombinação uniforme.

2.2.3.8.2 Operadores de Recombinação Diagonal

O segundo grupo é chamado de Recombinação Diagonal. Os operadores deste grupo usam o cruzamento baseado em segmentação e recombinação dos pais, podendo serem entendidos como generalização do operador de recombinação de múltiplos pontos.

2.2.3.8.3 Operadores de Recombinação Massiva

O terceiro grupo é conhecido como Recombinação Massiva. Os operadores deste grupo fazem uso do cruzamento baseado em operações numéricas envolvendo alelos em ponto flutuante, podendo serem encarados como generalização dos operadores de recombinação aritmética.

2.2.3.8.4 Vantagens do Cruzamento com Múltiplos Pais

Os estudos teóricos promovidos até o momento foram inconclusivos no que tange as vantagens dos operadores de cruzamento que utilizam vários pais no processo de recombinação. Apesar disso, resultados empíricos mostraram que em muitos casos, esses operadores se mostraram superiores aos clássicos que envolvem apenas dois pais (EIBEN; SCHIPPERS, 1996; EIBEN; SMITH, 2003).

Nota ao Leitor 10 *Os algoritmos de cruzamento com múltiplos pais não serão aprofundados nesta tese por se tratarem apenas de uma generalização dos algoritmos de recombinação com dois pais e por ainda não terem um uso tão amplo quanto estes.*

2.3 Algoritmos Evolutivos

Os algoritmos evolucionários ou evolutivos são algoritmos de busca e otimização inspirados na natureza, através dos quais busca-se encontrar uma boa solução para um determinado problema por meio dos passos evolutivos no ambiente computacional, envolvendo os processos cruzamento, mutação e seleção natural, que foram apresentados anteriormente. Dessa forma parte-se de um conjunto de soluções aleatórias que vão evoluindo até que se chegue a um conjunto de soluções que resolve efetivamente o problema.

Diferentes linhas de pesquisa desenvolveram técnicas computacionais que vieram

a compor os Algoritmos Evolucionários. As áreas de Programação Evolucionária, Estratégias Evolutivas e de Algoritmos Genéticos foram fundamentais para o desenvolvimento da área de Algoritmos Evolutivos, fornecendo as bases e relevantes contribuições a ela, passando estas a serem denominadas Algoritmos Evolucionários Canônicos.

Ao longo do tempo, outros tipos de algoritmos evolucionários foram se desenvolvendo com intuito de alcançar melhor desempenho, menor consumo de memória, menor custo computacional e maior velocidade, além de melhorias em outras características algorítmicas que venham a ser interessantes para um determinado problema ou determinadas condições.

A seções que se sucedem apresentam os tipos mais populares de algoritmos evolucionários, classificados em algoritmos evolucionários canônicos e algoritmos evolucionários não canônicos.

2.3.1 Algoritmos Evolucionários Canônicos

Os algoritmos evolucionários canônicos foram técnicas primordiais que fundamentaram os Algoritmos Evolutivos. Compreender as características e comportamentos dos algoritmos evolucionários canônicos é essencial para entender a própria natureza dos algoritmos evolutivos, bem como as atuais soluções baseadas em conceitos evolutivos propostas para resolver problemas e ainda, para possuir a base para o desenvolvimento de novas técnicas e ferramentas evolucionárias.

As subseções abaixo, mostram os algoritmos evolutivos canônicos e suas características fundamentais.

2.3.1.1 Programação Evolucionária

A programação evolucionária foi uma técnica proposta por Fogel (1962), cujo objetivo fundamental era utilizar os conceitos de evolução das espécies para o desenvolvimento da área de inteligência artificial.

Na programação evolucionária, cada indivíduo é representado por uma determinada máquina de estados finitos que processa uma sequência de símbolos.

Durante o processo de seleção, os indivíduos são submetidos a uma função de *fitness* que considera a saída da máquina de estados em que consiste o indivíduo em questão, tendo por referência uma saída esperada que configura a solução para um problema conhecido.

O processo de reprodução é caracterizado apenas pela presença de operadores de mutação, trata-se de um processo de clonagem dos indivíduos que sofre a incidência de

mutações, configurando um processo de reprodução puramente assexuada.

A população da próxima geração compreenderá não apenas os novos indivíduos, mas também os da geração atual. Dessa forma, os filhos competem também com seus próprios pais e apenas os indivíduos com maior *fitness* sobrevivem.

A programação evolucionária garante que todos os indivíduos de uma população gerarão novos indivíduos e que apenas os melhores dentre os atuais e os descendentes sobrevivam. O domínio total dos melhores indivíduos é denominado Elitismo total. O elitismo total garante apenas a sobrevivência dos k melhores indivíduos, onde $k < n$ e n é o tamanho da população (MORALES, 2004; MORALES; GARCIA, 2001).

No entanto, o elitismo total pode reduzir significativamente a variabilidade dos indivíduos, tornando-os muito semelhantes entre si. Esse fator pode acarretar na condução da solução do problema em questão para ótimos locais, além de poder também provocar um aumento substancial no tempo de convergência do algoritmo (JONG, 2006).

2.3.1.2 Estratégias Evolutivas

As Estratégias Evolutivas foram técnicas propostas, originalmente, por Rechenberg (1965), Schwefel (1975) com intuito de resolver problemas de otimização de parâmetros. Eles desenvolveram uma técnica chamada de $(1 + 1)$ -EE, que consiste em um indivíduo pai gerar apenas um único filho por reprodução assexuada e ambos competem pela sobrevivência.

Em uma estratégia evolutiva, cada gene do cromossomo representa uma dimensão do problema em questão e seu alelo é representado em ponto flutuante. Os cromossomos são compostos por dois arranjos (*arrays*), em que um deles armazena os valores para cada dimensão do problema e o outro guarda o desvio padrão de cada um desses valores.

A geração de um novo indivíduo se dá por meio da aplicação de um operador de mutação que apresenta distribuição de probabilidade gaussiana com média zero e desvio padrão do gene correspondente no indivíduo pai. O elitismo total também foi adotado nas estratégias evolutivas e conseqüentemente suas implicações valem também para esta.

Esse modelo primordial tinha um problema de lentidão em sua convergência. Por isso, ocorreu o desenvolvimento de dois novos modelos denominados (μ, λ) -EE e $(\mu + \lambda)$ -EE (JONG, 2006).

No modelo, (μ, λ) -EE, durante a seleção, μ indivíduos pais morrem e λ filhos sobrevivem, não existindo competição entre os μ indivíduos já existentes e os λ novos indivíduos.

Com relação ao modelo $(\mu + \lambda)$ -EE, apenas μ indivíduos são selecionados e sobrevivem dentre os μ indivíduos da geração atual e os λ indivíduos da nova geração, caracterizando assim uma competição entre indivíduos já existentes e os novos indivi-

duos.

2.3.1.3 Algoritmos Genéticos

O conceito de Algoritmo Genético foi proposto Holland (1975), que estudou a evolução natural das espécie e a considerou um processo simples, robusto e poderoso, vislumbrando que ele poderia ser adaptado e utilizado computacionalmente na obtenção de soluções eficientes para problemas de otimização.

O conceito de robustez mencionado tange ao fato dos algoritmos genéticos convergirem para soluções satisfatórias independentemente da escolha do conjunto de parâmetros iniciais para o problema (GOLDBERG, 1989; GOLDBERG, 2002).

A grande novidade que os algoritmos genéticos trazem em relação à programação evolucionária e as estratégias evolutivas é a chamada reprodução sexuada. Diferentemente dos anteriores, nos algoritmos genéticos, os indivíduos de uma nova geração não eram gerados apenas usando operadores de mutação, mas incluíam também o uso dos operadores de cruzamento, também chamados de operadores de recombinação ou *crossover*, aplicando assim um misto dos dois operadores (JONG, 2006).

O uso conjunto dos operadores de mutação e cruzamento, ao invés do uso de apenas um deles, proporciona o equilíbrio entre dois objetivos aparentemente conflitantes, sendo o primeiro, a maior exploração do espaço de busca, evitando a estagnação em ótimos locais, e o segundo, a seleção das melhores soluções obtidas, evitando a presença de soluções não ótimas.

Devido ao uso de dois operadores, o processo de busca se torna multidimensional, de modo a preservar soluções candidatas e promover a troca de informação entre as soluções exploradas (MICHALEWICZ, 1996; ZUBEN, 2000).

Segundo Michalewicz (1996), os algoritmos genéticos têm como principais etapas, as descritas a seguir.

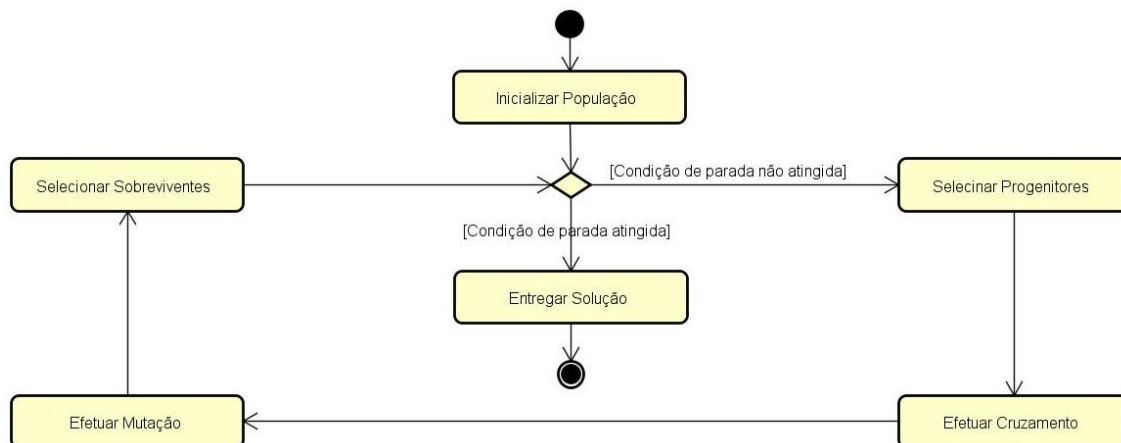
- No início de uma geração ou iteração g , existe uma população de indivíduos $P(g) = x_1^g, x_2^g, \dots, x_n^g$;
- Cada indivíduo $x_i^g, i = 1, 2, \dots, n$ é avaliado considerando a sua aptidão ou *fitness*;
- Da população atual, são escolhidos os indivíduos que serão os progenitores dos novos indivíduos no processo de reprodução;
- São gerados λ novos indivíduos a partir dos μ indivíduos da geração atual, através do cruzamento e da mutação, passando a serem considerados novas soluções potenciais;
- Dos indivíduos atuais e novos ($\mu + \lambda$), são selecionados μ indivíduos para a próxima

geração ($g + 1$), priorizando a escolha daqueles que apresentam uma maior aptidão ou *fitness*;

- Este processo se repete até que uma determinada condição de parada seja atingida;

A Figura 24 mostra um fluxograma dos passos elementares de um algoritmo genético.

Figura 24 - Fluxograma genérico de um algoritmo genético.



Fonte: O autor, 2023.

2.3.1.3.1 Codificação dos Indivíduos nos Algoritmos Genéticos

Conforme Goldberg (1989), nos algoritmos genéticos canônicos, os indivíduos que configuram as soluções candidatas são definidos por arranjos (*arrays*) binários de tamanho fixo.

No entanto, em certas aplicações, o uso da codificação binária apresenta desempenho insatisfatório. Por exemplo, em problemas de otimização numérica com parâmetros reais, algoritmos genéticos que trabalham com representação inteira ou em ponto flutuante comumente têm desempenho superior aos que trabalham com representação binária (DEB, 2001; MICHALEWICZ, 1996).

Representar os cromossomos por dígitos binários, segundo Michalewicz (1996), Deb (2001), acarreta em uma considerável perda de desempenho em problemas numéricos com muitas dimensões ou naqueles em que se deseja obter alta precisão numérica.

Michalewicz (1996), Deb (2001) também descrevem simulações computacionais que compararam alguns algoritmos genéticos que usavam codificação binária com outros que usavam codificação em ponto flutuante. Os resultados dessas simulações apontaram uma clara superioridade no desempenho da representação em ponto flutuante sobre a binária, nos casos considerados.

Fogel (1962) dissocia a eficiência de um AG da dimensão do espaço de busca do problema em questão. Ele argumenta que, desconsiderando a escolha da forma de representação dos cromossomos, o espaço de busca não define por si só a eficiência do algoritmo. Espaços de busca de grande dimensão podem ser eficientemente explorados, ao passo que espaços de dimensão reduzida podem apresentar considerável dificuldade em sua exploração.

2.3.1.3.2 Método de Seleção da Roleta

Os algoritmos genéticos canônicos usam um sistema de seleção denominado Algoritmo da Roleta para escolher os indivíduos que sobreviverão e passarão para a próxima geração. Uma implementação comum deste algoritmo consiste em definir uma lista $[a_1, a_2, a_3, \dots, a_\mu]$ de modo que a Equação (39) seja satisfeita, em que $P(i)$ é a função de probabilidade de um indivíduo passar para a próxima geração considerando seu *fitness* (GOLDBERG, 1989).

$$a_i = \sum_{c=1}^i P(i) \quad (39)$$

O Algoritmo 1 apresenta a dinâmica de funcionamento deste método de seleção.

Algoritmo 1: Algoritmo de seleção da roleta.

Entrada: $\mu, populacao$
Saída: $individuos_selecionados$

```

1 ROLETA( $\mu, populacao$ )
2 início
3   para  $i = 1$  até  $\mu$  faça
4     para  $c = 1$  até  $i$  faça
5        $a[i] \leftarrow a[i] + P(i)$ ;
6     fim
7   fim
8   para  $c = 1$  até  $\mu$  faça
9      $r \leftarrow$  (valor aleatório de probabilidade uniforme em  $[0, 1]$ );
10     $i \leftarrow 1$ ;
11    enquanto  $a[i] < r$  faça
12       $i \leftarrow i + 1$ ;
13       $a[i] \leftarrow a[i] + P(i)$ ;
14    fim
15     $individuos\_selecionados[c] \leftarrow populacao[i]$ ;
16  fim
17  retorna  $individuos\_selecionados$ ;
18 fim
```

2.3.1.3.3 Seleção Elitista

Em alguns casos, o desempenho dos algoritmos genéticos pode ser aprimorado através da imposição da escolha do melhor indivíduo encontrado em todas as gerações durante o processo de seleção.

Outra opção que pode ser adotada para melhorar o desempenho de um AG é sempre manter, na próxima geração, o indivíduo com o melhor *fitness* da geração atual. Essa estratégia é chamada de seleção elitista (FOGEL, 1994; MICHALEWICZ, 1996).

2.3.1.3.4 Seleção Baseada em Rank

Um outro mecanismo de seleção é a Seleção Baseada em Rank. Esta estratégia consiste em dispor os indivíduos ordenados (ranqueados) segundo o seu *fitness*. Quanto melhor o ranque, maior a probabilidade de seleção (MICHALEWICZ, 1996). Uma forma simples de implementar esse método de seleção é meramente escolher os μ melhores indivíduos e passá-los para a próxima geração (ZUBEN, 2000).

2.3.1.3.5 Seleção por Torneio

Existe ainda, uma outra estratégia chamada Seleção por Torneio. Ela consiste em definir um subgrupo da população atual contendo k indivíduos que são escolhidos aleatoriamente e em seguida, os melhores indivíduos deste subgrupo são selecionados para serem os novos pais. Esse processo se repete até serem designados todos os μ indivíduos que comporão o grupo dos novos progenitores.

Comumente, se trabalha com o torneio de 2 indivíduos ($k = 2$). Dessa forma, um grupo de dois indivíduos sorteados dentre toda a população competem entre si e o vencedor, que é o que apresenta maior *fitness* torna-se um dos pais. Esse mecanismo é repetido até que sejam selecionados μ indivíduos.

Duas características relevantes da seleção por torneio são o fato de não depender do conhecimento global da população e de não considerar o ranque do indivíduo perante a população, uma vez que ela não avalia a população inteira, mas sim um subgrupo desta. A presença dessas características permitem uma seleção menos tendenciosa que as outras alternativas apresentadas (EIBEN; SMITH, 2003).

O Algoritmo 2 ilustra o processo de Seleção por Torneio.

Algoritmo 2: Algoritmo de seleção por torneio.

Entrada: $\mu, populacao$
Saída: $individuos_selecionados$
1 *TORNEIO*($\mu, populacao$)
2 **início**
3 **para** $c = 1$ até μ **faça**
4 $populacao_selecionada \leftarrow (k$ indivíduos aleatórios da $populacao)$;
5 $melhor_individo \leftarrow populacao_selecionada[1]$;
6 **para** $l = 1$ até k **faça**
7 **se** $melhor_individo.aptidao < populacao_selecionada[l].aptidao$
8 **então**
9 $melhor_individo \leftarrow populacao_selecionada[l]$;
10 **fim**
11 **fim**
12 $individuos_selecionados[c] \leftarrow melhor_individo$;
13 **retorna** $individuos_selecionados$;
14 **fim**

2.3.2 Algoritmos Evolucionários Não Canônicos

Com o passar do tempo, surgiram diversas frentes de pesquisa sobre problemas de alta complexidade nas áreas de otimização e aprendizagem de máquina. Os estudos promovidos nestas pesquisas originaram novos tipos de algoritmos evolucionários com melhor eficiência em certas situações. Como estes algoritmos apresentam características distintas dos algoritmos primordiais apresentados, eles são denominados algoritmos evolucionários não canônicos.

Embora os algoritmos evolutivos não canônicos apresentem características distintas dos canônicos, seus conceitos são fortemente baseados nos algoritmos evolutivos canônicos, tendo alterações nas características fundamentais destes com intuito de aprimorá-los para certos tipos de problemas e situações.

As subseções abaixo, mostram os algoritmos evolutivos não canônicos e suas características fundamentais.

2.3.2.1 Programação Genética

Proposta por Koza (1989), a programação genética (PG) pode ser entendida como uma extensão dos algoritmos genéticos. O que difere a programação genética dos algoritmos genéticos é a representação dos indivíduos, os operadores de reprodução e os métodos de seleção (KOZA, 1992; KOZA, 1994; KOZA *et al.*, 1999; KOZA *et al.*, 2003).

A Programação genética foi desenvolvida com intuito de solucionar problemas de

aprendizagem de máquina. Seu objetivo primordial é o desenvolvimento automatizado de softwares computacionais (REZENDE, 2003).

Na PG, os indivíduos são representados como árvores da teoria dos grafos. Os nós internos dessas árvores possuem operações, procedimentos ou funções, enquanto os nós folha contêm variáveis, constantes ou argumentos para procedimentos ou funções.

Os operadores usados na PG, são operadores de mutação e cruzamento específicos para trabalhar com a representação de indivíduos por árvores.

Na operação de cruzamento, existe a troca de partes da árvore de um indivíduo pelas correspondentes de seus progenitores, sendo que os pontos onde as árvores são cortadas são definidos de modo a não originar operações inválidas a serem transferidas aos descendentes.

A operação de mutação consiste em alterar um nó ou uma subárvore. Caso um nó interno seja escolhido para ser mutado, este passará a possuir uma nova operação, função ou procedimento. Em caso de escolha de um nó folha, o valor presente no mesmo será aleatoriamente alterado. No caso de uma subárvore ser escolhida para sofrer a mutação, esta será substituída por uma nova subárvore compatível gerada aleatoriamente.

Alguns algoritmos de programação genética não adotam o operador de mutação no processo de busca de soluções, limitando-se à utilizar operações de cruzamento a fim de encontrar as melhores soluções dos problemas para os quais estão sendo aplicados.

O processo de avaliação consiste em executar o software algorítmico representado pela árvore em que consiste o indivíduo. Quanto melhor este aproximar a resposta certa, melhor será o valor de aptidão (*fitness*) a ele atribuído, ou quanto menos ele se aproximar da resposta procurada, pior será o seu *fitness*.

2.3.2.2 Algoritmos Micro Genéticos

Os algoritmos micro genéticos ou Micro-AGs são variantes dos algoritmos genéticos canônicos desenvolvidos com enfoque em evoluir eficientemente soluções trabalhando com populações de poucos indivíduos. Krishnakumar (1989) propôs, o que se acredita ser, o primeiro trabalho envolvendo algoritmos micro genéticos.

No modelo proposto por Krishnakumar (1989), uma população pequena é gerada e evoluída até que todos os indivíduos tenham genótipos muito similares, ou mesmo, iguais. O melhor indivíduo da população é selecionado e transferido para uma nova população, onde os demais indivíduos são gerados aleatoriamente. Comumente, nos Micro-AGs, trabalha-se com uma alta taxa de recombinação, estando próxima de um (100%), enquanto utiliza uma baixa taxa de mutação, que fica próxima de zero (0%).

Os algoritmos micro genéticos possuem grande relevância em diversas aplicações, principalmente em projetos com recursos computacionais limitados ou que necessitam de

alto desempenho, característica comum nos projetos de robótica e em aplicações de tempo real, respectivamente (CARVALHO *et al.*, 2004).

Nos problemas evolucionários em robótica em que cada indivíduo é representado por um robô, pode ser inviável construir um grande número de robôs em decorrência dos custos, requerendo que a população seja pequena.

Nas aplicações de tempo real, trabalhar com populações reduzidas confere maior desempenho, uma vez que necessitam de menos tempo computacional para serem geradas e evoluídas quando comparadas principalmente às populações de maiores tamanhos usadas nos algoritmos genéticos canônicos (DELBEM; CARVALHO; BRETAS, 2005).

2.3.2.3 Algoritmos de Estimação e Distribuição

Os algoritmos de estimação e distribuição (AEDs) foram propostos por Mhlenbein e Paa (1996) e vêm sendo intensamente estudados nos últimos anos. Eles incorporaram ao conceito de evolução, técnicas de aprendizagem capazes de correlacionarem variáveis de um problema de probabilidades, devidamente codificadas (LARRAENAGA; LOZANO, 2001; PAUL; IBA, 2003).

Neste modelo, os indivíduos são tratados como amostras, a partir das quais, um modelo probabilístico de distribuição é ajustado. Após o ajuste, determina-se os melhores indivíduos da população conforme o modelo ajustado.

A vantagem dos algoritmos de estimação e distribuição está no fato de dispensarem a especificação de parâmetros como taxa de cruzamento e mutação, não estando o modelo atrelado a este tipo de informação que é predefinida pelo usuário. Este tipo de definição é indispensável nos algoritmos canônicos, onde as taxas previamente definidas podem interferir diretamente no desempenho do algoritmo.

A desvantagem trazida pelos algoritmos de estimação e distribuição reside na alta complexidade e alto custo computacional que a maioria dos modelos probabilísticos utilizados para guiar o processo de busca pela solução demandam.

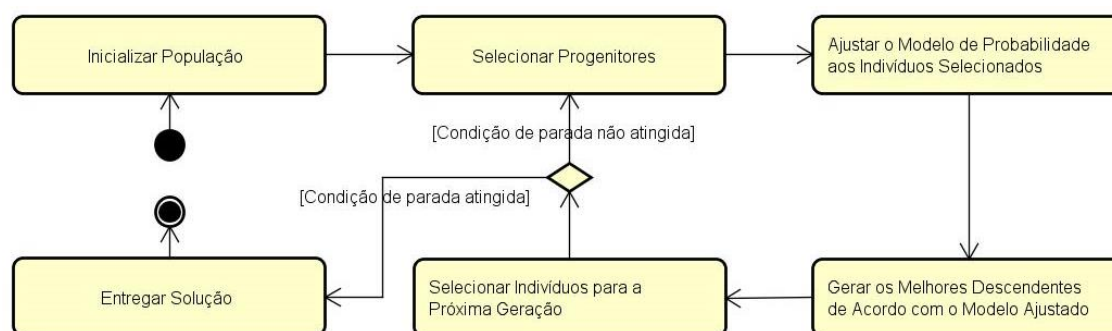
Com intuito de reduzir o problema de complexidade e custo computacional dos AEDs, novas abordagens foram propostas, como o *Univariate Marginal Distribution Algorithm* [UMDA] (MHLENBEIN, 1998), o *Bayesian Optimization Algorithm* [BOA] (PELIKAN; GOLDBERG; LOBO, 1999) e, mais recentemente, um algoritmo proposto por Melo *et al.* (2007) que trabalha com modelos simplificados e apresentam um desempenho satisfatório para uma certa gama de problemas.

Paul e Iba (2003) defendem que é razoável utilizar os algoritmos de estimação e distribuição ao invés de outros algoritmos evolucionários. Entretanto, estimar a distribuição de probabilidade associada à população de indivíduos selecionados é um grande gargalo desta abordagem, inexistindo, até o momento, métodos simples para trabalhar

com modelos complexos como os dos AEDs.

a Figura 25 apresenta o fluxograma de um algoritmo de estimação e distribuição clássico.

Figura 25 - Fluxograma simplificado de um algoritmo de estimação e distribuição.



Fonte: O autor, 2023.

2.3.2.4 Algoritmos Evolutivos para Otimização Multiobjetivo

Nota ao Leitor 11 *Alguns dos conceitos citados nesta subseção serão melhor detalhados e definidos mais adiante, no Capítulo 3, que trata especificamente de algoritmos de otimização.*

Os problemas de otimização multiobjetivo vêm ganhando relevante interesse na área de otimização. Neste tipo de problema, uma solução é qualificada com base em sua adequação a diversos objetivos, na maioria das vezes, conflitantes (DEB, 2001; EIBEN; SMITH, 2003).

2.3.2.4.1 Métodos Baseados em Pesos

Existem alguns métodos de solução que transformam o problema de otimização multiobjetivo em outro de apenas um objetivo, em que os demais objetivos são configurados como restrições, e a partir deste, procuram uma solução para o problema transformado.

Um conjunto de métodos que funciona da forma descrita acima são os Métodos Baseados em Pesos ou Métodos de Pesos. Nestes métodos, são atribuídos diferentes pesos aos objetivos, de forma que o processo é feito através da otimização de uma função que considera estes pesos em sua implementação (ARENALES *et al.*, 2007).

O problema dos métodos baseados em pesos está na sua dependência da escolha

adequada dos pesos, uma vez que a escolha incorreta dos pesos pode levar a soluções insatisfatórias. Para escolher mais adequadamente os pesos, pode ser necessário conhecer previamente os intervalos correspondentes a estes. Informação esta que na maioria das vezes, não é conhecida.

2.3.2.4.2 Métodos de Múltiplas Soluções

Devido ao problema acima apresentado, iniciou-se a busca por métodos que solucionem problemas de otimização multiobjetivo sem o uso de pesos (EIBEN; SMITH, 2003). A ideia é que estes métodos não mais reduziram um problema multiobjetivo em um mono objetivo e forneceriam uma solução, mas sim resolveriam o problema multiobjetivo em sua formulação original, fornecendo não apenas uma, mas um conjunto de soluções ótimas, chamado de conjunto ótimo de Pareto ou fronteira de Pareto (DEB, 2001).

Schaffer (1985) propôs, até onde se sabe, o primeiro Algoritmo Evolucionário para Otimização Multiobjetivo (AEOM) nos moldes descritos acima. Este algoritmo foi intitulado *Vector Evaluated Genetic Algorithm* (VEGA), sendo um algoritmo genético alterado que avalia cada objetivo de maneira separada.

O algoritmo de Schaffer apresenta uma característica problemática no que tange a resolução dos problemas de otimização multiobjetivo. As soluções da fronteira de Pareto obtidas através do algoritmo VEGA apresentam baixa diversidade, configurando indivíduos muito semelhantes, o que empobrece o conjunto de soluções.

2.3.2.4.3 Dominância

Posteriormente, Goldberg (1989) estabeleceu o conceito de dominância, que fornece um valor de aptidão (*fitness*) para uma determinada solução que é diretamente proporcional ao número de soluções dominadas por esta. Nesse contexto, dominar uma solução significa não ser pior que ela em nenhum dos objetivos e ser melhor em ao menos um deles. Dessa forma, quanto mais soluções uma determinada solução dominar, maior será sua aptidão e, assim, terá uma maior quantidade de aparições no conjunto de soluções.

Para a manutenção da diversidade das soluções, Goldberg (1989) propôs o uso de um método de compartilhamento que calcula o nicho de cada uma das soluções dentro da fronteira à qual esta pertence. Vários algoritmos evolutivos para otimização multiobjetivo foram propostos com embasamento nos conceitos estabelecidos por Goldberg.

2.3.2.4.4 Seleção Baseada em Dominância

Os algoritmos evolutivos para otimização multiobjetivo e os algoritmos evolucionários tradicionais têm por diferença principal o operador de seleção. Nos AEOMs, o operador de seleção usa o conceito de dominância para comparar e escolher as soluções, ao invés da proximidade a uma solução correta conhecida, como ocorre nos algoritmos evolutivos tradicionais.

Algoritmos como o *Multi-objective Genetic Algorithm* [MOGA] (FONSECA; FLEMING, 1993) e o *Strength Pareto Evolutionary Algorithm* [SPEA] (ZITZLER; THIELE, 1998) calculam o valor de aptidão (*fitness*) proporcionalmente à dominância da solução. Outros algoritmos como o *Niched Pareto Genetic Algorithm for Multi-Objective Optimization* [NPGA] (HORN; NAFPLIOTIS; GOLDBERG, 1994) usam a dominância, mas não trabalham com o cálculo de um valor de aptidão.

2.3.2.4.5 Algoritmos Elitistas e Não Elitistas

Deb (2001), Eiben e Smith (2003) classificam os algoritmos evolutivos para otimização multiobjetivo em dois tipos, os elitistas e os não elitistas.

Os algoritmos elitistas são aqueles que, em seu processo de seleção, escolhem de algum modo um percentual alto dos melhores indivíduos presentes na população a cada iteração, sendo estes modos denominados formas de elitismo. Os algoritmos *Pareto Envelope-based Selection Algorithm* [PESA] e o *Strength Pareto Evolutionary Algorithm* [SPEA] usam uma população externa para armazenar as soluções dominantes (elite) encontradas até a iteração atual (CORNE; KNOWLES; OATES, 2000). O Non-dominated Sorting Genetic Algorithm II (NSGA-II) realiza uma união entre a população atual e a nova; e desta união, mantém as melhores soluções de ambas (DEB *et al.*, 2000).

Os algoritmos não elitistas por sua vez, não utilizam nenhuma forma de elitismo durante as iterações. O algoritmo Non-Dominated Sorting Genetic Algorithm (NSGA), predecessor do algoritmo elitista Non-dominated Sorting Genetic Algorithm II (NSGA-II), é um exemplo de algoritmo não elitista.

Zitzler, Deb e Thiele (2000) procederam com um estudo que levou a conclusão de que o usar o elitismo melhora significativamente as soluções encontradas por um algoritmo evolutivo para otimização multiobjetivo.

2.4 Referências com Informações Complementares

No curso do presente capítulo, foram enunciadas as estruturas que representam computacionalmente os conceitos evolucionários, descritos os operadores evolutivos de reprodução e suas características, e apresentados os principais tipos de algoritmos evolucionários pertinentes e relacionados aos assuntos abordados nesta tese, acompanhados de conceitos fundamentais.

No entanto, o tema não foi explanado com total profundidade, de forma que nem todos os operadores, algoritmos e outros temas não foram tratados aqui, uma vez que não serão utilizados diretamente nesta tese.

Temas como reprodução por permutação, recombinação por mistura, recombinação de ordem, recombinação cíclica, cruzamento baseado em posição, cruzamento baseado em ordem, cruzamento parcialmente mapeado, inteligência de enxame e algoritmos imuno-inspirados não foram percorridos aqui.

Se desejar conhecer ou se aprofundar mais sobre esses e outros conceitos, orienta-se a leitura das seguintes obras da literatura:

- Ávila (2020);
- Santos (2019);
- Itaborahy Filho (2018);
- Petroli Neto (2011);
- Zuben (2000);
- Back, Fogel e Michalewicz (2000);
- Eiben e Smith (2003);
- Goldberg (1989);
- Gabriel e Delbem (2008a);
- Eiben, Kemenade e Kok (1995);
- Castro (2006);
- Jong (2006);
- Gabriel e Delbem (2008b);
- Paiva (2018);

3 MÉTODOS DE OTIMIZAÇÃO

Diante da vastidão do tempo e da imensidão do espaço, é um imenso prazer para mim dividir um planeta e uma época com você.

Carl Sagan

Os métodos de otimização compõem parte importante deste trabalho, estando relacionados às soluções dos problemas formulados nesta tese, bem como ao algoritmo e à ferramenta computacional, que foram desenvolvidos nesta tese. Por esta razão, conhecer os métodos de otimização é imprescindível para o pleno entendimento deste trabalho.

De nada adianta, formular um problema de otimização com intuito de fazer com que um processo produza melhores resultados, consuma menos tempo e utilize menos recursos, se não dispormos de meios para solucionar este problema.

O papel dos métodos de otimização é justamente promover a resolução de problemas de otimização, fornecendo uma ou mais soluções, dependendo do método, que permitem a correta tomada de decisões dentro do respectivo contexto em que o problema está inserido.

O presente capítulo apresentará metaheurísticas voltadas tanto à resolução de problemas de otimização mono-objetivo quanto a problemas de otimização multibjetivo.

3.1 Métodos Metaheurísticos de Otimização Mono-Objetivo

Os métodos metaheurísticos de otimização mono-objetivo são voltados à solução de problemas de otimização com apenas um objetivo, usando técnicas heurísticas guiadas. Métodos deste tipo também estão disponíveis na biblioteca de código desenvolvida nesta tese. Por este motivo, é interessante que o leitor esteja a par destas técnicas.

Os métodos metaheurísticos ou simplesmente metaheurísticas são procedimentos de alto nível que guiam processos heurísticos, geralmente de busca local, com intuito de explorar soluções além de um ótimo local, procurando encontrar um ótimo global (LUKE, 2016).

Esse tipo de algoritmo se baseia nas soluções encontradas para explorar novas regiões do espaço de busca com intuito de encontrar soluções mais promissoras nele.

Algoritmos metaheurísticos têm a vantagem de requerer pouco conhecimento acerca do problema a ser resolvido, não exigindo que se suponha muitas características sobre ele. Além disso, eles conseguem solucionar problemas cujo espaço de busca seja extenso, uma vez que, neste caso, os métodos clássicos podem levar um tempo consideravelmente alto, inviabilizando, muitas vezes, seu uso neste tipo de problema (LUKE, 2016).

A desvantagem deste tipo de método é que não existe qualquer garantia de que uma solução que leve a um ótimo local ou global do problema de otimização em questão será encontrada (LUKE, 2016).

Uma importante classe de algoritmos metaheurísticos são os algoritmos evolucionários voltados para resolução de problemas de otimização. Estes constituem uma parte considerável dos algoritmos metaheurísticos.

Nesta seção, serão apresentados os métodos metaheurísticos de otimização mono-objetivo Evolução Diferencial e Algoritmos Genéticos.

3.1.1 Evolução Diferencial (ED)

A Evolução Diferencial (ED) é um método de otimização proposto inicialmente por Storn e Price (1995) destinado a solucionar problemas de otimização mono-objetivo. Trata-se de um algoritmo evolucionário de otimização, uma vez que se baseia nos mecanismos de seleção natural e na genética, utilizando operadores de mutação, cruzamento e seleção para gerar novos indivíduos em busca da melhor solução.

Embora seja considerado um algoritmo evolutivo, diversas referências da literatura não consideram a ED um algoritmo bio inspirado, devido ao fato de seus operadores de mutação, recombinação e seleção serem matematicamente definidos e não inspirados na natureza, sendo denominados operadores sintéticos (DAS; SUGANTHAN, 2011).

A ED é um algoritmo dotado de simplicidade, requerendo apenas quatro parâmetros, sendo estes, a constante de ponderação de diferença F , a probabilidade da ocorrência de recombinação ℓ , o tamanho da população q e o número de gerações g_{max} (SILVA, 2010).

Além disso, a ED é um algoritmo que possui baixa dependência de seus parâmetros, sendo tolerante a escolhas de valores menos adequados, e tem uma boa convergência, obtendo comumente resultados satisfatórios (ROCHA; SARAMAGO, 2011).

3.1.1.1 Mutação Diferencial

Na evolução diferencial, o operador de mutação é empregado para gerar uma população de vetores mutantes a partir dos vetores presentes na população de indivíduos da geração atual.

O operador de mutação consiste em gerar um vetor mutante v^i , chamado de vetor doador, a partir de um vetor aleatório r_0 da população perturbado pela diferença entre outros dois vetores aleatórios r_1^i e r_2^i desta multiplicados por um fator de ponderação de diferença F .

A Equação (40) define o operador de mutação diferencial, onde i é o índice do

indivíduo em questão e o fator de ponderação de diferença F deve possuir um valor que esteja no intervalo $[0, 2]$.

$$v^i = r_0^i + F(r_1^i - r_2^i) \quad (40)$$

O Algoritmo 3 apresenta a operação de mutação empregada fim de gerar q vetores doadores.

Algoritmo 3: Mutação diferencial.

Entrada: \mathcal{P}, q, F
Saída: v

- 1 *MUTACAO_DIFERENCIAL*(\mathcal{P}, q, F)
- 2 **início**
- 3 **para** $i = 1$ **até** q **faça**
- 4 $r_0^i \leftarrow$ (um vetor aleatório de \mathcal{P});
- 5 $r_1^i \leftarrow$ (um vetor aleatório de \mathcal{P} tal que $r_1^i \neq r_0^i$);
- 6 $r_2^i \leftarrow$ (um vetor aleatório de \mathcal{P} tal que $r_2^i \neq r_0^i$ e $r_2^i \neq r_1^i$);
- 7 $v^i \leftarrow r_0^i + F(r_1^i - r_2^i)$;
- 8 **fim**
- 9 **retorna** v ;
- 10 **fim**

Na literatura, existem diversas abordagens para definir o valor do parâmetro F , não existindo consenso acerca deste assunto (DAS; SUGANTHAN, 2011). Segundo Storn e Price (1997b), um valor inicial apropriado para F é $\frac{1}{2}$.

3.1.1.2 Cruzamento Diferencial

O algoritmo da ED emprega o operador de recombinação para gerar uma população de vetores recombinados, denominados vetores experimentais, a partir dos vetores doadores criados pela operação de mutação e dos vetores da população original da geração atual, chamados de vetores alvos.

O operador de cruzamento consiste em gerar um vetor recombinado u^i , em que cada uma de suas componentes u_j^i possui uma probabilidade ℓ de herdar o valor da respectiva componente v_j^i , do respectivo vetor doador v^i . Caso esta atribuição não se concretize, u_j^i assumirá o valor da respectiva componente χ_j^i do respectivo vetor alvo χ^i .

A Equação (41) define o operador de cruzamento diferencial, onde b_j é um valor aleatório entre 0 e 1, que definirá se uma determinada componente de u^i herdará o valor

da respectiva componente de v^i , segundo a probabilidade de ocorrência ℓ .

$$u_j^i = \begin{cases} v_j^i, & b_j \leq \ell \\ \chi_j^i, & b_j > \ell \end{cases}, \quad \forall j \in \{1, 2, 3, \dots, n\} \quad (41)$$

O Algoritmo 4 mostra a operação de cruzamento usada para constituir uma população de q indivíduos experimentais.

Algoritmo 4: Cruzamento diferencial.

Entrada: χ, q, v, ℓ

Saída: u

```

1 CRUZAMENTO_DIFERENCIAL( $\chi, q, v, \ell$ )
2 início
3   para  $i = 1$  até  $q$  faça
4     para  $j = 1$  até  $n$  faça
5        $b_j \leftarrow$  (um número real aleatório do intervalo  $[0, 1]$ );
6       se  $b_j \leq \ell$  então
7          $u_j^i \leftarrow v_j^i$ ;
8       senão
9          $u_j^i \leftarrow \chi_j^i$ ;
10      fim
11    fim
12  fim
13  retorna  $u$ ;
14 fim
```

3.1.1.3 Seleção Diferencial

O processo de seleção dos indivíduos se baseia no valor resultante da função objetivo aplicada a cada um deles, uma vez que a função objetivo é a função de avaliação e o valor mencionado se traduz na aptidão de cada indivíduo.

O operador de seleção adotará como melhor indivíduo \bar{x}^i , o vetor u^i , caso a função objetivo aplicada a u^i apresente um valor menor que se aplicada ao vetor original χ^i da geração atual.

A Equação (42) define o operador de seleção diferencial, onde u^i é o i -ésimo vetor recombinado e χ^i é o i -ésimo vetor original da geração atual.

$$\bar{x}^i = \begin{cases} u^i, & f(u^i) \leq f(\chi^i) \\ \chi^i, & f(u^i) > f(\chi^i) \end{cases} \quad (42)$$

O Algoritmo 5 apresenta a operação de seleção usada para escolher todos os q indivíduos da próxima geração.

Algoritmo 5: Seleção diferencial.

Entrada: χ, q, u
Saída: $\bar{\chi}$

```

1 SELECAO_DIFERENCIAL( $\chi, q, u$ )
2 início
3   para  $i = 1$  até  $q$  faça
4     se  $f(u^i) \leq f(\chi^i)$  então
5       |  $\bar{\chi}^i \leftarrow u^i$ ;
6     senão
7       |  $\bar{\chi}^i \leftarrow \chi^i$ ;
8     fim
9   fim
10  retorna  $\bar{\chi}$ ;
11 fim
```

3.1.1.4 Algoritmo Geral da Evolução Diferencial

Enunciados os operadores e conceitos fundamentais, pode-se então enunciar o algoritmo evolução diferencial, que consiste, inicialmente, em produzir uma população \mathcal{P}_0 de q indivíduos aleatórios com distribuição de probabilidade uniforme.

Subsequentemente, desenvolve-se q vetores doadores através do processo de mutação e em seguida, aplicam-se operações de cruzamento a fim de gerar q vetores experimentais.

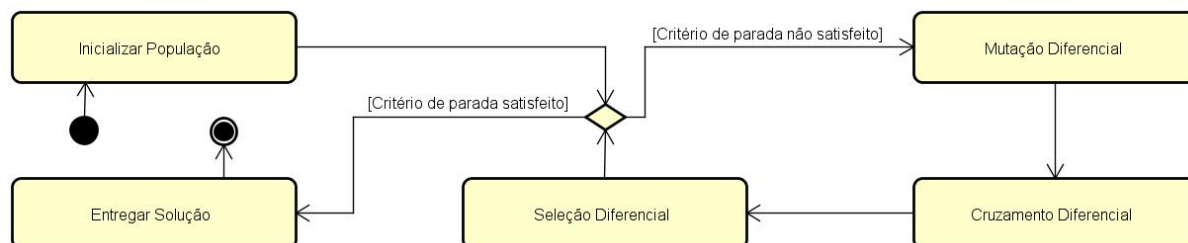
A seguir, são selecionados q vetores dentre os vetores alvos e os vetores experimentais, onde se escolhe o vetor experimental caso seu custo for menor que o custo do respectivo vetor alvo. Caso contrário, o vetor alvo é escolhido.

Os vetores selecionados constituem os vetores alvos da próxima geração que compõem a população \mathcal{P}_{g+1} . Este processo se repete a partir da etapa de mutação considerando a nova população gerada até que o número de gerações pretendida seja atingida.

A Figura 26 apresenta um diagrama que ilustra as etapas da evolução diferencial.

O Algoritmo 6 sintetiza o método evolução diferencial completo, que foi descrito acima.

Figura 26 - Etapas da evolução diferencial.



Fonte: Adaptado de Rosario, 2011.

Algoritmo 6: Evolução diferencial.

Entrada: P, u, ℓ

Saída: \mathcal{P}_g

1 *EVOLUCAO_DIFERENCIAL*(P, u, ℓ)

2 **início**

3 $\mathcal{P}_0 \leftarrow (q \text{ indivíduos gerados aleatoriamente com distribuição de probabilidade uniforme});$

4 **para** $g = 0$ **até** g_{max} **faça**

5 $\iota \leftarrow 0;$

6 **para todo** $x \in \mathcal{P}_g$ **faça**

7 $\chi^\iota \leftarrow x;$

8 $\iota \leftarrow \iota + 1;$

9 **fim**

10 $v \leftarrow \text{MUTACAO_DIFERENCIAL}(\mathcal{P}_g, q, F);$

11 $u \leftarrow \text{CRUZAMENTO_DIFERENCIAL}(\chi, q, v, \ell);$

12 $\bar{\chi} \leftarrow \text{SELECAO_DIFERENCIAL}(\chi, q, u)$

13 $\mathcal{P}_{g+1} \leftarrow \emptyset$

14 **para** $\iota = 1$ **até** q **faça**

15 $\mathcal{P}_{g+1} \leftarrow \mathcal{P}_{g+1} \cup \{\bar{\chi}^\iota\};$

16 **fim**

17 **fim**

18 **retorna** $\mathcal{P}_g;$

19 **fim**

3.1.1.5 Aplicação da Evolução Diferencial em Problemas de Otimização Restrita

Para empregar a evolução diferencial em problemas restritos, pode-se utilizar métodos de otimização restrita, tais como o Método das Barreiras, Método das Penalidades e Método dos Lagrangeanos Aumentados (MLA), a fim de transformá-los em problemas de otimização irrestrita por meio do uso de parâmetros de penalização.

Nota ao Leitor 12 *Os métodos de otimização restrita não serão descritos e aprofundados neste trabalho. Para conhecer ou obter maior aprofundamento no assunto, consultar a literatura sugerida ao final do capítulo.*

3.1.2 Algoritmo Genético (AG)

Os algoritmos genéticos, apresentados na Seção 2.3.1.3, podem ser facilmente empregados para solucionar um problema de otimização mono-objetivo. Não é necessária, nenhuma adaptação ou complemento para que eles desempenhem a tarefa de algoritmo otimizador (ÁVILA, 2020).

3.1.2.1 Características do Algoritmo Genético para Otimização

A única exigência para que esta abordagem funcione é que a função de aptidão deve ser a função objetivo do problema de otimização em questão. As demais definições, tais como escolha de operadores e forma de representação, são livres, uma vez que nenhuma delas inviabilizará a resolução do problema de otimização, muito embora, certas opções podem ser mais vantajosas para determinados tipos de problema.

3.1.2.2 Aplicação de Algoritmos Genéticos em Problemas de Otimização Restrita

Quando se deseja resolver um problema de otimização restrita, existem diversas alternativas para lidar com as restrições. Algumas delas consistem nos métodos de otimização restrita, que transformam os problemas restritos em irrestritos.

No entanto, uma abordagem interessante que pode ser adotada por algoritmos evolucionários de otimização é realizar o tratamento das restrições, durante a seleção por torneio dos indivíduos pertencentes à população (DEB *et al.*, 2002).

Na seleção por torneio binário, onde uma dentre duas soluções é escolhida, quando se está na presença de restrições, existem três cenários possíveis, a saber:

Caso 1: Ambas as soluções são viáveis;

Caso 2: Uma solução é viável e a outra não é viável;

Caso 3: Ambas as soluções são inviáveis.

Na abordagem enunciada, no caso de um problema de otimização mono-objetivo, a escolha da solução para cada um dos casos acima, seguem critérios bastante simples, a saber:

No Caso 1: Escolhe-se a solução cuja função objetivo apresenta o menor valor;

No Caso 2: Escolhe-se a solução viável;

No Caso 3: Escolhe-se a solução que apresenta a menor violação de restrições.

Nesta metodologia, inexistem quaisquer comparações ou mesclas entre restrições e objetivos, dispensando, deste modo, qualquer uso de parâmetro de penalidade. Essas características fazem esta abordagem ser bastante atrativa e utilizada, uma vez que não é complexa, possui implementação simples e não provoca qualquer mal condicionamento numérico (DEB *et al.*, 2002).

3.2 Métodos Metaheurísticos de Otimização Multiobjetivo

Os métodos metaheurísticos de otimização multiobjetivo destinam-se a solucionar problemas de otimização com múltiplos objetivos, usando técnicas que obtêm soluções aproximadas para estes. Métodos desta natureza estão presentes na ferramenta computacional desenvolvida e serão empregados para resolver os problemas de otimização tratados neste trabalho. Por esta razão, deve-se conhecer este tipo de técnica afim de compreender a presente tese.

Os métodos metaheurísticos de otimização multiobjetivo possuem as mesmas características, vantagens e desvantagens dos métodos metaheurísticos de otimização mono-objetivo mostradas no início do Seção 3.1.

Diferentemente dos métodos clássicos de otimização multiobjetivo, os métodos metaheurísticos de otimização multiobjetivo não consistem na transformação do problema multiobjetivo em um problema clássico.

Desse modo, esse tipo de método atua diretamente sobre o problema multiobjetivo, buscando um conjunto de soluções ótimas da curva de Pareto em uma mesma execução do algoritmo. Além desta característica, os métodos metaheurísticos buscam encontrar um conjunto de soluções que apresente uma boa diversidade, objetivando assim atingir um bom número de soluções que apresentem boa convergência e diversidade (DEB, 2001).

Nesta seção, serão apresentados os métodos metaheurísticos de otimização multiobjetivo *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) e o *Adaptive Evolutionary Algorithm Based on Non Euclidean Geometry* (AGE-MOEA).

3.2.1 Non-dominated Sorting Genetic Algorithm II (NSGA-II)

O algoritmo Non-Dominated Sorting Genetic Algorithm (NSGA) foi inicialmente proposto por Srinivas e Deb (1995) com o intuito de implementar por meio da adaptação de um algoritmo genético, os conceitos de dominância e diversidade de soluções, seguindo os conceitos de otimização multiobjetivo indicados por Goldberg (1989).

Posteriormente, Deb *et al.* (2002) desenvolveu como evolução do algoritmo NSGA, o Non-Dominated Sorting Genetic Algorithm II (NSGA-II), que tornou-se muito popular em virtude de seu baixo custo computacional para ordenar as soluções por dominância e por não requerer parâmetros para promover a diversidade das soluções, sendo, atualmente, um dos métodos mais empregados e difundidos na literatura de metaheurísticas de otimização multiobjetivo (LOBATO, 2008; RAMPAZZO, 2012).

3.2.1.1 Ordenação em Frentes Não Dominadas

O processo de ordenação elitista por dominância das soluções usado no NSGA-II é denominado *Fast Non-Dominated Sorting*. Ele consiste na classificação dos indivíduos da população \mathcal{P} em diversos níveis ou frentes de dominância $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \dots, \mathcal{F}_w)$, conforme o grau de dominância de cada indivíduo, sendo w o número de frentes de dominância.

Assim, a frente \mathcal{F}_1 contém todos os indivíduos não dominados de \mathcal{P} . A frente \mathcal{F}_2 , por sua vez, contém todos os indivíduos não dominados de $\mathcal{P} - \mathcal{F}_1$. A frente \mathcal{F}_3 , por conseguinte, contém todos os indivíduos não dominados de $\mathcal{P} - (\mathcal{F}_1 \cup \mathcal{F}_2)$; e assim sucessivamente até a frente \mathcal{F}_w que contém todos os indivíduos de $\mathcal{P} - (\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3 \cup \dots \cup \mathcal{F}_{w-1})$.

Outra perspectiva de enxergar as frentes é perceber que os elementos da frente \mathcal{F}_1 dominam os indivíduos das frentes $\mathcal{F}_2, \mathcal{F}_3, \dots, \mathcal{F}_w$, os elementos da frente \mathcal{F}_2 dominam os indivíduos das frentes $\mathcal{F}_3, \mathcal{F}_4, \dots, \mathcal{F}_w$; e assim por diante, até chegar aos elementos da frente \mathcal{F}_{w-1} que dominam os indivíduos da frente \mathcal{F}_w .

O Algoritmo 7 formaliza o processo de ordenação em frentes não dominadas *Fast Non-Dominated Sorting*, onde \bar{S}_x é o conjunto das soluções dominadas pela solução $x \in \mathcal{P}$, $\hat{\eta}_x$ é o número de soluções que dominam x , $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \dots, \mathcal{F}_w)$ são as frentes de dominância e \mathcal{R}_x é o valor de aptidão ou *fitness*, neste caso, denominado de ranque, da

solução x .

Algoritmo 7: Ordenação em frentes não dominadas.

```

Entrada:  $\mathcal{P}$ 
Saída:  $\mathcal{F}, \mathcal{R}$ 
1 ORDENACAO_EM_FRENTES_NAO_DOMINADAS( $\mathcal{P}$ )
2 início
3   para todo  $x \in \mathcal{P}$  faça
4      $\bar{S}_x \leftarrow \emptyset$ ;
5      $\hat{\eta}_x \leftarrow 0$ ;
6     para cada  $y \in \mathcal{P}$  faça
7       se  $x \prec y$  então
8          $\bar{S}_x \leftarrow \bar{S}_x \cup \{y\}$ ;
9       senão se  $y \prec x$  então
10         $\hat{\eta}_x \leftarrow \hat{\eta}_x + 1$ ;
11      fim
12    fim
13    se  $\hat{\eta}_x = 0$  então
14       $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \{x\}$ ;
15       $\mathcal{R}_x \leftarrow 1$ ;
16    fim
17  fim
18   $\iota \leftarrow 1$ ;
19  enquanto  $\mathcal{F}_\iota \neq \emptyset$  faça
20     $\mathcal{F}_{\iota+1} \leftarrow \emptyset$ ;
21    para todo  $x \in \mathcal{F}_\iota$  faça
22      para cada  $y \in \bar{S}_x$  faça
23         $\hat{\eta}_y \leftarrow \hat{\eta}_y - 1$ ;
24        se  $\hat{\eta}_y = 0$  então
25           $\mathcal{R}_y \leftarrow \iota + 1$ ;
26           $\mathcal{F}_{\iota+1} \leftarrow \mathcal{F}_{\iota+1} \cup \{y\}$ ;
27        fim
28      fim
29    fim
30     $\iota \leftarrow \iota + 1$ ;
31  fim
32  retorna  $(\mathcal{F}, \mathcal{R})$ ;
33 fim

```

3.2.1.2 Diversidade de Soluções

O uso do ranque para ordenar e selecionar as soluções durante o processo de busca contribui para a convergência do método. No entanto, conforme discutido na Seção 1.3.4.4, a convergência não é a única propriedade que importa em um conjunto de soluções de um

problema multiobjetivo. A diversidade das soluções também é um fator muito importante em um conjunto de soluções, uma vez que uma boa diversidade garante que as soluções sejam esparsas e uniformes.

A diversidade da população também é muito importante para um algoritmo evolucionário, pois indivíduos com uma boa diversidade tendem a gerar filhos com maior aptidão. Nas metaheurísticas, a diversidade confere uma maior probabilidade de encontrar soluções ótimas, uma vez que estas permitem uma maior exploração do conjunto viável.

Para existir uma boa diversidade, deve-se elencar soluções que não estejam muito próximas a outras na frente de Pareto. Para isso, é necessário calcular, de algum modo, a distância entre uma solução e as que estão ao seu redor, também chamada de densidade populacional em torno de uma solução.

3.2.1.3 Distância de Aglomeração

O NSGA-II usa como métrica para estimar a densidade populacional em torno de um indivíduo, a chamada Distância de Aglomeração ou *Crowding Distance*, pois seu cálculo não requer que nenhum parâmetro externo seja informado, além de apresentar um baixo custo computacional.

A distância de aglomeração consiste na distância entre as duas soluções mais próximas do indivíduo em questão. A interpretação geométrica da distância de aglomeração de uma solução corresponde ao semi perímetro do hiper retângulo que conecta as duas soluções adjacentes a ela.

O cálculo da distância de aglomeração requer que, para cada objetivo f_i , as soluções $x \in \mathcal{S}$ estejam ordenadas crescentemente pelo valor da função objetivo f_i aplicada a cada uma delas, para $i = 1, 2, 3, \dots, m$, sendo m o número de objetivos do problema e \mathcal{S} um conjunto de soluções.

Desse modo, denomina-se $s = (s_1, s_2, s_3, \dots, s_\mu)$ o vetor das soluções $x \in \mathcal{S}$ ordenadas pelo valor $f_i(x)$, onde μ é número de soluções contidas em \mathcal{S} . Cada s_j corresponde a uma solução $x \in \mathcal{S}$ distinta, para $j \in \{1, 2, 3, \dots, \mu\}$.

Dessa forma, a distância de aglomeração α^* de uma solução s_j se dá pela Equação (43), onde $f_i^{max} = f_i(x_i^{max})$, sendo que $x_i^{max} \in \mathcal{S}$ tal que $f_i(x_i^{max}) \geq f_i(x)$ para todo $x \in \mathcal{S}$, $f_i^{min} = f_i(x_i^{min})$, sendo que $x_i^{min} \in \mathcal{S}$ tal que $f_i(x_i^{min}) \leq f_i(x)$ para todo $x \in \mathcal{S}$ e **ord**, a função que ordena crescentemente as soluções de \mathcal{S} pelo valor de f_i .

$$\alpha_{s_j}^* = \sum_{i=1}^m \frac{f_i(s_{j+1}) - f_i(s_{j-1})}{f_i^{max} - f_i^{min}}, \text{ sendo } s = \mathbf{ord}(\mathcal{S}, f_i) \quad (43)$$

A razão da distância $[f_i(s_{j+1}) - f_i(s_{j-1})]$ por $(f_i^{max} - f_i^{min})$ tem o intuito de

normalizá-la. Essa normalização se faz necessária, pois os objetivos comumente estão em diferentes escalas de grandeza e conseqüentemente suas distâncias também estarão.

Pode-se concluir que, quanto maior a distância de aglomeração de uma solução, menos densa é a região ao seu redor na fronteira de Pareto, indicando que se trata de uma solução mais esparsa, pois existem poucas soluções ao seu redor.

Por outro lado, quanto menor a distância de aglomeração de uma solução, mais densa é a região ao seu redor na curva de Pareto, indicando que a solução é menos esparsa, existindo muitas soluções ao seu redor. Assim, são desejáveis, soluções que apresentem a maior distância de aglomeração possível.

Existem algumas exceções à fórmula de cálculo da distância de aglomeração dada pela Equação (43). Tratam-se de casos particulares de soluções que, devido a sua importância, devem ser selecionadas pelo NSGA-II. Tais casos estão especificados a seguir.

Para cada objetivo f_i , considerando o vetor de soluções ordenadas s pelos valores de f_i , tanto a solução s_1 , quanto a solução s_μ possuem a distância de aglomeração infinita.

Isso se deve ao fato destas soluções serem imprescindíveis para a diversidade das soluções, uma vez que se tratam dos extremos do conjunto de soluções, ou seja, $s_1 = x_i^{min}$ e $s_\mu = x_i^{max}$.

Assim, elas são prioritárias em relação às outras, pois apresentam o melhor e o pior valor para o objetivo f_i , individualmente. A distância de aglomeração infinita atribuída a tais soluções garante que elas serão selecionadas.

A Equação (44) expressa a fórmula final de cálculo da distância de aglomeração α , considerando os casos em que esta é infinita.

$$\alpha_{s_j} = \begin{cases} \alpha_{s_j}^*, & 1 < j < \mu \\ \infty, & j = 1 \vee j = \mu \end{cases} \quad (44)$$

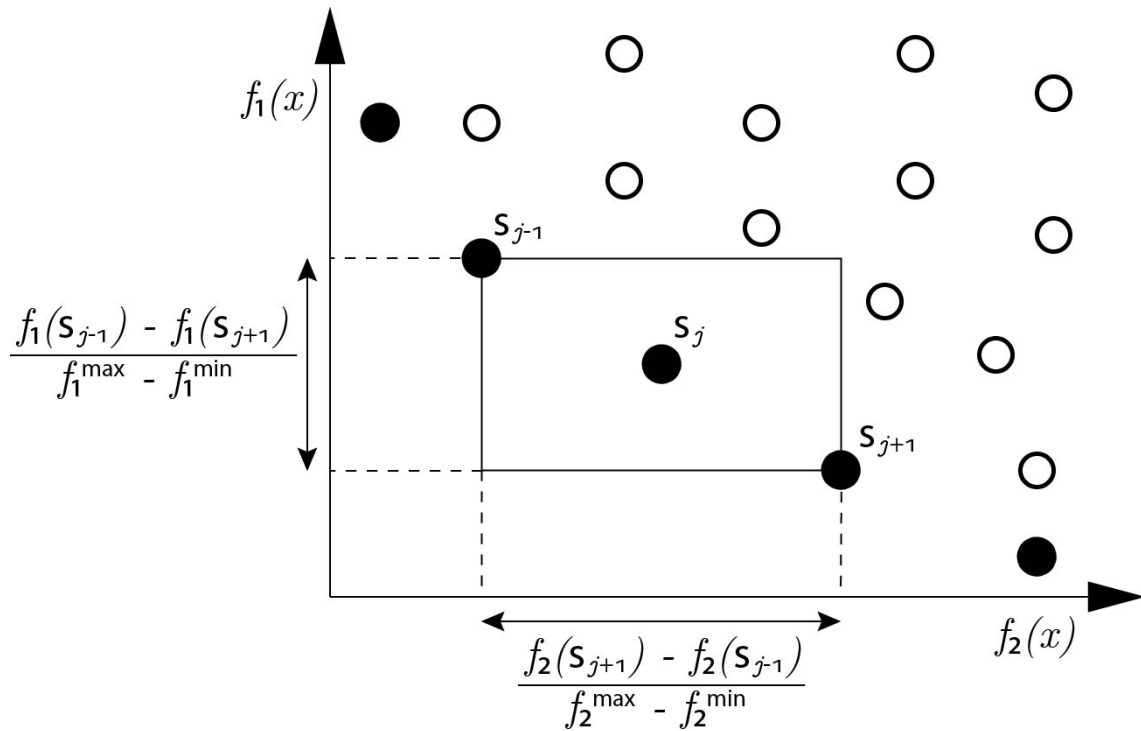
A Figura 27 ilustra geometricamente a distância de aglomeração de uma solução em um problema com dois objetivos.

O Algoritmo 8 formaliza um modo de calcular a distância de aglomeração ou *crowding distance*, onde $\aleph(\mathcal{S})$ é a cardinalidade de \mathcal{S} . Devido as soluções estarem ordenadas pelo valor f_i , pode-se concluir que, considerando todas as soluções em s , a solução s_1 é a que leva f_i ao seu menor valor e s_μ ao seu maior valor. Portanto, pode-se fazer $f_i^{min} = f_i(s_1)$ e $f_i^{max} = f_i(s_\mu)$.

3.2.1.4 Comparação de Aglomeração

Para empregar a distância de aglomeração como requisito de seleção do NSGA-II, Deb *et al.* (2002) propuseram o conceito de comparação de aglomeração ou *crowded com-*

Figura 27 - Distância de aglomeração.



Fonte: O autor, 2023.

Algoritmo 8: Distância de aglomeração.

Entrada: \mathcal{S}

Saída: α

```

1 DISTANCIA_DE_AGLOMERACAO( $\mathcal{S}$ )
2 início
3    $\mu \leftarrow \aleph(\mathcal{S});$ 
4   para todo  $x \in \mathcal{S}$  faça
5      $\alpha_x \leftarrow 0;$ 
6   fim
7   para  $i = 1$  até  $m$  faça
8      $s \leftarrow$  (soluções  $x \in \mathcal{S}$  ordenadas crescentemente pelo valor de  $f_i(x)$ );
9      $\alpha_{s_1} \leftarrow \infty;$ 
10     $\alpha_{s_\mu} \leftarrow \infty;$ 
11     $f_i^{min} \leftarrow f_i(s_1);$ 
12     $f_i^{max} \leftarrow f_i(s_\mu);$ 
13    para  $j = 2$  até  $\mu - 1$  faça
14       $\alpha_{s_j} \leftarrow \alpha_{s_j} + \frac{f_i(s_{j+1}) - f_i(s_{j-1})}{f_i^{max} - f_i^{min}};$ 
15    fim
16  fim
17  retorna  $\alpha;$ 
18 fim
  
```

parison, que introduziu o operador binário de comparação de aglomeração (\prec_η), também conhecido como *crowded comparison operator*, que classifica uma solução como maior ou menor segundo ao seu ranque e distância de aglomeração. A Definição 3.2.1 enuncia o conceito de comparação de aglomeração.

Definição 3.2.1 (Comparação de Aglomeração) *Sejam as soluções τ_1 e τ_2 , seus respectivos ranques \mathcal{R}_{τ_1} e \mathcal{R}_{τ_2} ; e suas respectivas distâncias de aglomeração α_{τ_1} e α_{τ_2} , dizemos que τ_1 é menor no critério de aglomeração que τ_2 , denotado por $\tau_1 \prec_\eta \tau_2$, se o ranque de τ_1 for menor que o ranque de τ_2 ou caso o ranque de τ_1 seja igual ao de τ_2 e a distância de aglomeração de τ_1 seja maior que a de τ_2 , isto é, se a Equação (45) ou a Equação (46) for atendida.*

$$\mathcal{R}_{\tau_1} > \mathcal{R}_{\tau_2} \tag{45}$$

$$\mathcal{R}_{\tau_1} = \mathcal{R}_{\tau_2} \tag{46a}$$

$$\alpha_{\tau_1} > \alpha_{\tau_2} \tag{46b}$$

3.2.1.5 Algoritmo Principal do NSGA-II

Enunciados os conceitos preliminares, é possível, agora, apresentar o algoritmo NSGA-II. A dinâmica do NSGA-II consiste, inicialmente, em gerar um população inicial de progenitores \mathcal{A}_0 com q indivíduos. Depois, esta população é ordenada em frentes não dominadas, a fim de obter o ranque \mathcal{R} de cada um de seus indivíduos. Subsequentemente, aplica-se uma seleção por torneio sobre os indivíduos de \mathcal{A}_0 , considerando o ranque de cada um como valor de aptidão, que definirá o conjunto de pais \mathcal{A}_0^* que participarão do processo reprodutivo. Em seguida, são executados os operadores de cruzamento e de mutação sobre os indivíduos de \mathcal{A}_0^* de modo a gerar uma população \mathcal{D}_0 de q descendentes. Estas populações constituem a geração inicial ou geração zero ($g = 0$) do algoritmo.

Reúne-se a população de pais \mathcal{A}_g e a de filhos \mathcal{D}_g da geração g atual em uma única população \mathcal{P}_g . Depois, ordena-se os indivíduos em frentes não dominadas $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \dots, \mathcal{F}_w$, calculando-se também o ranque \mathcal{R} de cada indivíduo.

Subsequentemente, inicia-se a população de pais da nova geração \mathcal{A}_{g+1} vazia. Esta população deve ter exatamente q indivíduos. Para preenche-la, vão sendo inseridos os indivíduos das frentes não dominadas, começando pela frente \mathcal{F}_1 e indo até se encontrar uma frente \mathcal{F}_z cujo o número de elementos seja superior às vagas restantes para comple-

tar os q indivíduos de \mathcal{A}_{g+1} . Neste momento, também são calculadas as distâncias de aglomeração de todos os indivíduos das frentes $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \dots, \mathcal{F}_4$.

Para preencher as vagas restantes de \mathcal{A}_{g+1} e completar seus q indivíduos, ordena-se os indivíduos da fronteira \mathcal{F}_z do menor para o maior, usando a comparação de aglomeração, e em seguida, esses indivíduos vão sendo acrescentados à \mathcal{A}_{g+1} , partindo do primeiro até chegar àquele que completará os q indivíduos de \mathcal{A}_{g+1} . Os indivíduos restantes da fronteira \mathcal{F}_z , bem como os das demais frentes são desconsiderados.

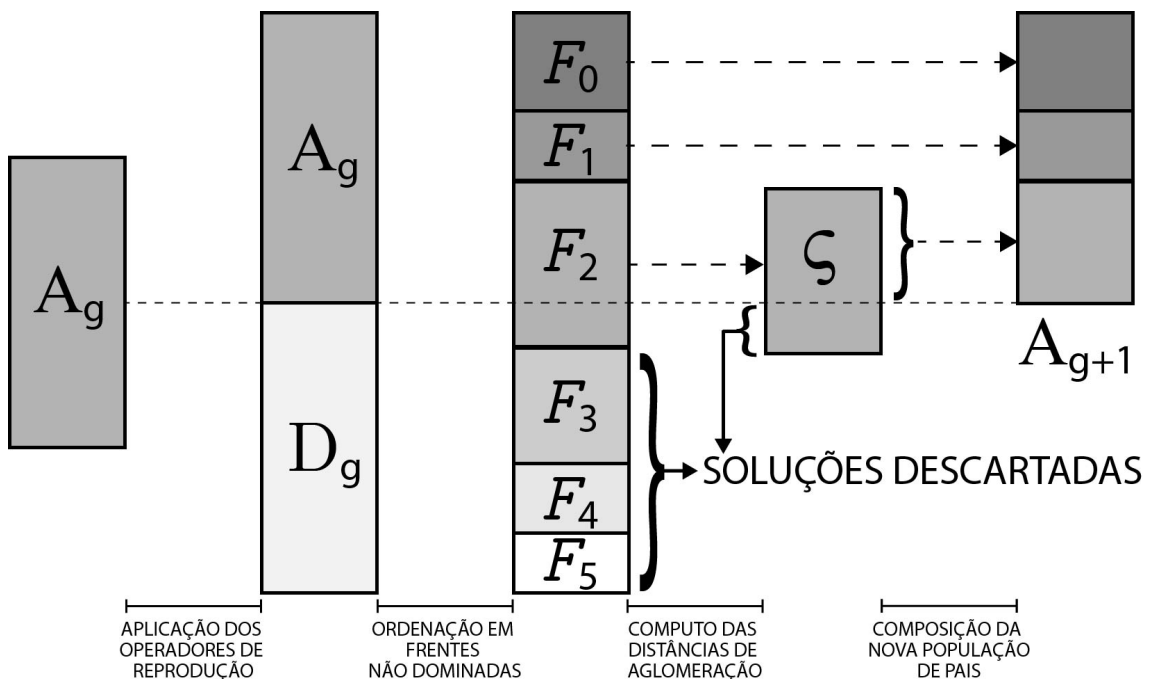
Promove-se então uma seleção por torneio sobre os indivíduos de \mathcal{A}_{g+1} , comparando os indivíduos por meio da comparação de aglomeração, definindo o conjunto de pais \mathcal{A}_{g+1}^* que participarão do processo reprodutivo.

São executados os operadores de cruzamento e de mutação sobre os indivíduos de \mathcal{A}_{g+1}^* de modo a gerar a população de filhos \mathcal{D}_{g+1} . Os indivíduos de \mathcal{A}_{g+1} e \mathcal{D}_{g+1} são mesclados em uma única população e o processo se repete até alcançar o número desejado de gerações (g_{max}).

Importante mencionar que a quantidade de indivíduos q de uma população e o número máximo de gerações g_{max} são parâmetros externos fornecidos pelo usuário do método.

A Figura 28 ilustra, em alto nível, as etapas do processo algorítmico do NSGA-II.

Figura 28 - Etapas do NSGA-II.



Legenda: Nesta figura, ζ ilustra a fronteira F_2 ordenada por meio do operador de comparação de aglomeração \prec_η .

Fonte: Adaptado de Deb *et al.*, 2002.

O Algoritmo 9 sintetiza o NSGA-II, onde $\aleph(\cdot)$ denota o operador de cardinalidade

de conjuntos.

Canonicamente, o NSGA-II adota a representação binária, apresentada na Seção 2.1.4.1, operador de mutação padrão, mostrado na Seção 2.2.2.1.1, o operador de recombinação de um ponto, apresentado na Seção 2.2.3.2, e a seleção por torneio binário, mostrada na seção Seção 2.3.1.3.5.

3.2.1.6 Aplicação do NSGA-II a Problemas de Otimização Restrita

Diante de um problema de otimização restrita, existem diversas abordagens que podem ser empregadas para tratar as restrições, tais como, os métodos de otimização restrita, que convertem os problemas restritos em irrestritos.

Entretanto, por conta de suas vantagens, é possível adotar a abordagem enunciada na Seção 3.1.2.2, que consiste no tratamento das restrições durante a seleção por torneio dos indivíduos, com as devidas alterações para lidar com problemas de otimização multiobjetivo.

No caso de problemas de otimização multiobjetivo, os casos 2 e 3 são tratados da mesma forma que nos problemas de otimização mono-objetivo, que foi descrita acima. A alteração na abordagem ocorre no caso 1, onde será escolhida a solução dominante. Caso não exista relação de dominância entre as soluções, deve-se usar um outro critério para elencar a melhor solução.

Uma forma simples de introduzir essa metodologia no processo de seleção por torneio é substituir o conceito de dominância pelo de dominância restrita, formalizado pela Definição 3.2.2. Dessa forma, a comparação realizada na seleção por torneio levará em consideração as restrições envolvidas no problema de otimização (DEB *et al.*, 2002).

Definição 3.2.2 (Dominância Restrita) *Seja o problema de minimização dado pela Equação (14) e os vetores v_1 e v_2 , dizemos que v_1 domina restritamente v_2 , se ao menos uma das seguintes condições forem satisfeitas.*

1. *A solução v_1 é viável e a solução v_2 não;*
2. *Tanto a solução v_1 quanto a solução v_2 não são viáveis, mas solução v_1 possui menor violação de restrições que v_2 ;*
3. *Tanto a solução v_1 quanto a solução v_2 são viáveis, mas a solução v_1 domina v_2 .*

O NSGA-II adota esta metodologia para resolver eventuais problemas de otimização restrita.

Algoritmo 9: NSGA-II.

Entrada: q, g_{max}
Saída: \mathcal{A}_g

- 1 $NSGA_II(q, g_{max})$
- 2 **início**
- 3 $\mathcal{A}_0 \leftarrow \mathcal{D}_0 \leftarrow \emptyset;$
- 4 $\mathcal{A}_0 \leftarrow (q \text{ indivíduos gerados aleatoriamente});$
- 5 $(_, \mathcal{R}) \leftarrow ORDENACAO_EM_FRENTES_NAO_DOMINADAS(\mathcal{A}_0)$
- 6 $\mathcal{A}_0^* \leftarrow (\text{indivíduos de } \mathcal{A}_0 \text{ selecionados por torneio, cujas aptidões são dadas por } \mathcal{R});$
- 7 $\mathcal{D}'_0 \leftarrow (q \text{ indivíduos obtidos pelo cruzamento entre os elementos de } \mathcal{A}_0^*);$
- 8 $\mathcal{D}_0 \leftarrow (q \text{ indivíduos obtidos pela mutação dos elementos de } \mathcal{D}'_0);$
- 9 **para** $g = 0$ **até** g_{max} **faça**
- 10 $\mathcal{P}_g \leftarrow \mathcal{A}_g \cup \mathcal{D}_g;$
- 11 $(\mathcal{F}, \mathcal{R}) \leftarrow ORDENACAO_EM_FRENTES_NAO_DOMINADAS(\mathcal{P}_g)$
- 12 $\mathcal{A}_{g+1} \leftarrow \emptyset;$
- 13 $\iota \leftarrow 1;$
- 14 **enquanto** $\aleph(\mathcal{A}_{g+1}) \leq q$ **faça**
- 15 $\alpha^\iota \leftarrow DISTANCIA_DE_AGLOMERACAO(\mathcal{F}_\iota);$
- 16 **se** $\aleph(\mathcal{F}_\iota) > [q - \aleph(\mathcal{A}_{g+1})]$ **então**
- 17 $\varsigma \leftarrow (\text{soluções } x \in \mathcal{F}_\iota \text{ ordenadas através da comparação por } \prec_\eta);$
- 18 **para** $j = 1$ **até** $q - \aleph(\mathcal{A}_{g+1})$ **faça**
- 19 $\mathcal{A}_{g+1} \leftarrow \mathcal{A}_{g+1} \cup \{\varsigma_j\}$
- 20 **fim**
- 21 **senão**
- 22 $\mathcal{A}_{g+1} \leftarrow \mathcal{A}_{g+1} \cup \mathcal{F}_\iota;$
- 23 **fim**
- 24 $\iota \leftarrow \iota + 1;$
- 25 **fim**
- 26 **para** $\kappa = 1$ **até** ι **faça**
- 27 **para todo** $x \in \alpha^\kappa$ **faça**
- 28 $\bar{\alpha}_x \leftarrow \alpha_x^\kappa;$
- 29 **fim**
- 30 **fim**
- 31 $\mathcal{A}_{g+1}^* \leftarrow (\text{indivíduos de } \mathcal{A}_{g+1} \text{ selecionados por torneio, cuja comparação é dada por } \prec_\eta);$
- 32 $\mathcal{D}'_{g+1} \leftarrow (q \text{ indivíduos obtidos pelo cruzamento entre os elementos de } \mathcal{A}_{g+1}^*);$
- 33 $\mathcal{D}_{g+1} \leftarrow (q \text{ indivíduos obtidos pela mutação dos elementos de } \mathcal{D}'_{g+1});$
- 34 **fim**
- 35 **retorna** $\mathcal{A}_g;$
- 36 **fim**

3.2.2 Adaptive Evol. Algorithm Based on Non Euclidean Geometry (AGE-MOEA)

O algoritmo Adaptive Evolutionary Algorithm based on Non-Euclidean Geometry (AGE-MOEA) foi proposto por Panichella (2019) com o propósito de resolver problemas de otimização multiobjetivo através de um algoritmo evolucionário que seja efetivo também quando empregado na resolução de problemas cuja fronteira de Pareto apresenta geometria não euclideana.

Observou-se que em métodos de otimização multiobjetivo, como o NSGA-II, o desempenho destes apresenta uma degradação considerável quando o número de objetivos é maior que três, pois o número de soluções não dominadas em cada população cresce exponencialmente conforme o número de objetivos aumenta (PURSHOUSE; FLEMING, 2007; TANABE; OYAMA, 2017).

Na tentativa de contornar o problema acima, surgiram diversos métodos com diferentes abordagens, tais como o NSGA-III (DEB; JAIN, 2014a; DEB; JAIN, 2014b), θ -DEA (YUAN *et al.*, 2016), GrEA (YANG *et al.*, 2013) e AR-MOEA (TIAN *et al.*, 2018a).

Todas estas técnicas supõem implicitamente que a frente de Pareto do problema a ser resolvido possui uma geometria euclideana. No entanto, muitos problemas de otimização multiobjetivo apresentam fronteiras de Pareto com outros tipos de geometria, tais como hiperbólica ou esférica (MARTINEZ *et al.*, 2014; TIAN *et al.*, 2018b).

O método AGE-MOEA não pressupõe inicialmente nenhuma geometria para a frente de Pareto do problema de otimização a ser resolvido, buscando estimá-la em cada uma de suas iterações.

3.2.2.1 Ordenação em Frentes Não Dominadas

O AGE-MOEA, assim como o NSGA-II, utiliza o processo de ordenação em frentes não dominadas, o *Fast Non-Dominated Sorting*, mostrado na Seção 3.2.1.1, para classificar os indivíduos da população em vários níveis de dominância.

3.2.2.2 Normalização de Objetivos

O algoritmo requer que os objetivos do problema de otimização estejam normalizados segundo a Equação (47), onde f_i é o i -ésimo objetivo. f_i^{min} e θ_i serão definidos a

seguir.

$$\bar{f}_i(x) = \frac{f_i(x) - f_i^{min}}{\theta_i} \quad (47)$$

f_i^{min} consiste no valor de $f_i(x_i^{min})$, onde x_i^{min} é a solução que leva f_i ao menor valor, considerando todas as soluções da frente não dominada \mathcal{F}_1 , sendo dado pela Equação (48), sendo que, neste caso, $\mathcal{S} = \mathcal{F}_1$.

$$f_i^{min} = \mathbf{min}_{x \in \mathcal{S}} [f_i(x)] \quad (48)$$

θ_i consiste na interseção do hiperplano m -dimensional $\bar{f}^{max} \in \mathbb{R}^m$ com o eixo do objetivo f_i . O hiperplano \bar{f}^{max} é constituído pelo maior valor de cada objetivo após sua transladação para a origem dos eixos, considerando as soluções presentes em \mathcal{F}_1 . Desse modo, ele tem a forma $\bar{f}^{max} = \{\bar{f}_1^{max}, \bar{f}_2^{max}, \bar{f}_3^{max}, \dots, \bar{f}_m^{max}\}$, sendo que cada \bar{f}_i^{max} é calculado pela Equação (49), onde $\mathcal{S} = \mathcal{F}_1$.

$$\bar{f}_i^{max} = \mathbf{max}_{x \in \mathcal{S}} (f_i(x) - f_i^{min}) \quad (49)$$

As interseções entre \bar{f}^{max} e o eixo de cada um dos objetivos f_i , para $i = 1, 2, 3, \dots, m$, consistem no vetor $\theta \in \mathbb{R}^m$, de forma $\theta = (\theta_1, \theta_2, \theta_3, \dots, \theta_m)$, que pode ser encontrado através da solução do sistema de equações lineares mostrado pela Equação (50).

$$\bar{f}^{max} \theta = 1 \quad (50)$$

Existem situações em que o sistema de equações (50) é impossível ou é possível, mas indeterminado. Nestes casos, as interseções são calculadas individualmente pela normalização Min-Max dada pela Equação (51) (PANICHELLA, 2019).

$$\theta_i = (\bar{f}_i^{max} - f_i^{min}) \quad (51)$$

O Algoritmo 10 sintetiza o processo de normalização dos objetivos, tendo como resultado a função objetivo vetorial normalizada $\bar{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ de forma $\bar{f} = \langle \bar{f}_1, \bar{f}_2, \bar{f}_3, \dots, \bar{f}_m \rangle$, onde n é o número de variáveis e m , o número de objetivos.

3.2.2.3 Estimação da Geometria da Frente de Pareto

O processo de inferência da geometria da frente de Pareto consiste em estipular o parâmetro ρ da norma L_ρ que será usada para calcular a proximidade e diversidade das soluções candidatas, de forma que estas não sejam calculadas retilinearmente, mas sim

Algoritmo 10: Normalizar objetivos.

Entrada: \mathcal{S}
Saída: $\bar{f}(\cdot)$
 1 *NORMALIZAR_OBJETIVOS*(\mathcal{S})
 2 **início**
 3 **para** $i = 1$ **até** m **faça**
 4 $f_i^{min} \leftarrow \mathbf{min}_{x \in \mathcal{S}} [f_i(x)];$
 5 $\bar{f}_i^{max} \leftarrow \mathbf{max}_{x \in \mathcal{S}} (f_i(x) - f_i^{min});$
 6 **fim**
 7 **se** (o sistema de equações lineares $\bar{f}^{max}\theta = 1$ for determinado) **então**
 8 $\theta \leftarrow$ (solução do sistema de equações lineares $\bar{f}^{max}\theta = 1$);
 9 **senão**
 10 **para** $i = 1$ **até** m **faça**
 11 $\theta_i \leftarrow \bar{f}_i^{max} - f_i^{min};$
 12 **fim**
 13 **fim**
 14 $\bar{f}_i(\cdot) \leftarrow \frac{f_i(\cdot) - f_i^{min}}{\theta_i};$
 15 **retorna** $\bar{f}(\cdot);$
 16 **fim**

sobre a geometria da frente.

Desse modo, para alcançar este objetivo, o algoritmo AGE-MOEA aproxima o valor de ρ através da Equação (52), onde m é o número de objetivos e ζ_i é a i -ésima componente do ponto ζ , que será estabelecido a seguir.

$$\rho = \frac{\log(m)}{\log(m) - \log(\sum_{i=1}^m \zeta_i)} \quad (52)$$

O ponto $\zeta \in \mathbb{R}^m$ é o ponto mais ao centro da frente \mathcal{F}_1 . Seu valor é dado por $\bar{f}(\varpi)$, onde ϖ é a solução encontrada por meio da resolução do problema de otimização simples dado pela Equação (53), onde $\mathcal{S} = \mathcal{F}_1$. O vetor $\vec{\chi}$ e a função $\gamma^\perp(\cdot)$ serão definidos a seguir.

$$\varpi = \begin{cases} \text{Minimizar}_x & \gamma^\perp(\bar{f}(x), \vec{\chi}) \\ \text{Sujeito a} & x \in \mathcal{S} \end{cases} \quad (53)$$

O vetor $\vec{\chi}$ é definido por $\overrightarrow{\chi_0\chi_1}$, onde $\chi_0 \in \mathbb{R}^m$ é o menor vetor objetivo ideal normalizado, ou seja, $\chi_0 = (0, 0, 0, \dots, 0)$ e $\chi_1 \in \mathbb{R}^m$ é o maior vetor objetivo ideal normalizado, ou seja, $\chi_1 = (1, 1, 1, \dots, 1)$. Logo, $\vec{\chi}$ é definido conforme a Equação (54).

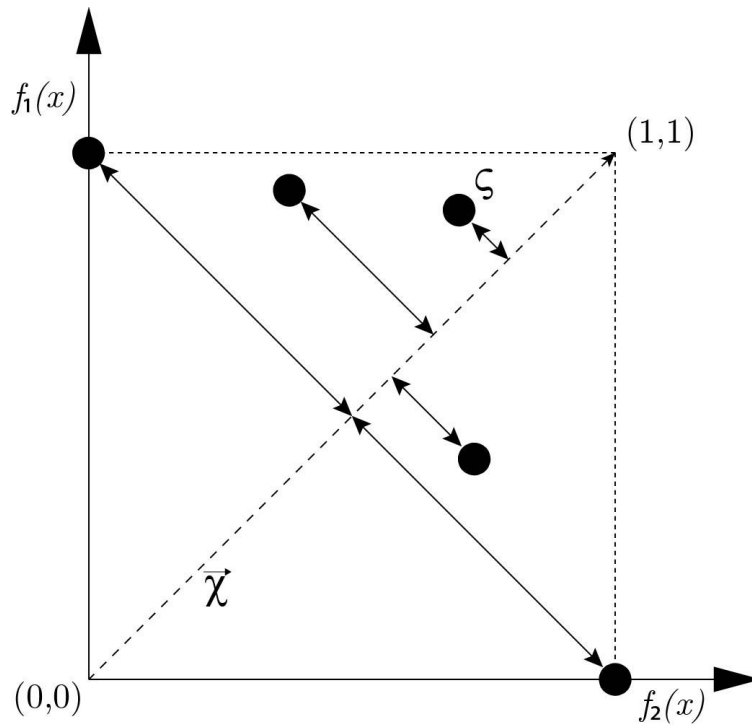
$$\vec{\chi} = \overrightarrow{\chi_1\chi_2} = \chi_2 - \chi_1 = (1, 1, \dots, 1) - (0, 0, \dots, 0) = (1, 1, \dots, 1) \quad (54)$$

A função $\gamma^\perp : \mathbb{R}^{2m} \rightarrow \mathbb{R}$ define a menor distância perpendicular entre um ponto $\varrho \in \mathbb{R}^m$ e a reta dada pelo vetor $\vec{v} \in \mathbb{R}^m$, sendo definida pela Equação (55), onde o vetor $\vec{v} = \overrightarrow{\nu_1 \nu_2} = \nu_2 - \nu_1$, o vetor $\vec{\varrho} = \overrightarrow{\nu_1 \varrho} = \varrho - \nu_1$, o operador \times denota o produto vetorial e $\|\cdot\|_2$ é a norma euclídeana.

$$\gamma^\perp(\varrho, \vec{v}) = \frac{\|\vec{\varrho} \times \vec{v}\|_2}{\|\vec{v}\|_2} \quad (55)$$

A Figura 29 mostra o ponto mais ao centro de uma frente \mathcal{F}_1 em um problema com dois objetivos.

Figura 29 - Ponto mais ao centro.



Legenda: O ponto ζ ilustrado é o ponto mais próximo de $\vec{\chi}$.

Fonte: O autor, 2023.

O Algoritmo 11 formaliza os passos para aproximar o expoente ρ , obtendo assim a geometria da frente.

3.2.2.4 Proximidade e Diversidade de Soluções

No AGE-MOEA, a proximidade e diversidade de soluções são calculadas através da norma L_ρ , tendo ρ , o valor calculado pela Equação (52). A proximidade de uma solução x é dada por $\|\bar{f}(x) - \chi_0\|_\rho$, onde χ_0 é o menor vetor objetivo ideal. Porém, uma vez que os objetivos estão normalizados, $\chi_0 = (0, 0, 0, \dots, 0)$, então pode-se estabelecer que a

Algoritmo 11: Geometria da frente.

Entrada: $\mathcal{S}, \bar{f}(\cdot)$
Saída: ρ
 1 *GEOMETRIA_DA_FRENTE*($\mathcal{S}, \bar{f}(\cdot)$)
 2 **início**
 3 $\varpi \leftarrow \begin{cases} \text{Minimizar } \gamma^{\perp}(\bar{f}(x), \bar{\mathcal{X}}) \\ \text{Sujeito a } x \in \mathcal{S} \end{cases} ;$
 4 $\zeta \leftarrow \bar{f}(\varpi);$
 5 $\rho \leftarrow \frac{\log(m)}{\log(m) - \log\left(\sum_{i=1}^m \zeta_i\right)}$
 6 **retorna** $\rho;$
 7 **fim**

proximidade φ de uma solução x é dada pela Equação (56).

$$\varphi_x = \|\bar{f}(x)\|_{\rho} \quad (56)$$

A diversidade ϑ' de uma solução x consiste na distância dada pela norma L_{ρ} entre ela e a solução mais próxima a ela. A Equação (57) estabelece como se dá o cálculo da diversidade de uma solução x .

$$\vartheta'_x = \min_{\substack{y \in \mathcal{S} \\ y \neq x}} (\|\bar{f}(x) - \bar{f}(y)\|_{\rho}) \quad (57)$$

Pode-se também calcular a diversidade ϑ de uma solução x pela soma da distância entre ela e a solução mais próxima a ela com a distância entre ela e a segunda solução mais próxima a ela. Desse modo, contempla-se duas soluções adjacentes ao invés de uma. A Equação (58) expressa a fórmula de cálculo da diversidade de uma solução x , levando em conta as duas soluções mais próximas, onde y^* é a solução mais próxima de x (PANICHELLA, 2019).

$$\vartheta_x = \min_{\substack{y \in \mathcal{S} \\ y \neq x}} (\|\bar{f}(x) - \bar{f}(y)\|_{\rho}) + \min_{\substack{y \in \mathcal{S} \\ y \neq x \\ y \neq y^*}} (\|\bar{f}(x) - \bar{f}(y)\|_{\rho}) \quad (58)$$

3.2.2.5 Pontuação de Sobrevivência

O valor que será usado tanto como aptidão dos indivíduos quanto para ordenar os indivíduos da última frente a ser inserida é chamado aqui de pontuação de sobrevivência. O cálculo desta pontuação é o mesmo para todas as frentes, diferindo apenas no caso da frente \mathcal{F}_1 .

Para os elementos da frente \mathcal{F}_1 , a pontuação de sobrevivência \mathcal{E} é dada pela razão entre a diversidade e a proximidade do elemento em questão. A Equação (59) expressa a pontuação para indivíduos de \mathcal{F}_1 .

$$\mathcal{E}'_x = \frac{\vartheta_x}{\varphi_x} \quad (59)$$

Para os elementos das demais frentes, a pontuação de sobrevivência \mathcal{E} é dada pelo inverso da proximidade do elemento em questão. A Equação (60) expressa a pontuação para indivíduos de todas as frentes, exceto a frente \mathcal{F}_1 .

$$\mathcal{E}^*_x = \frac{1}{\varphi_x} \quad (60)$$

Generalizando, pode-se definir a fórmula de cálculo da pontuação de sobrevivência pela Equação (61).

$$\mathcal{E}_x = \begin{cases} \frac{\vartheta_x}{\varphi_x}, & x \in \mathcal{F}_1 \\ \frac{1}{\varphi_x}, & x \notin \mathcal{F}_1 \end{cases} \quad (61)$$

As soluções que levam ao extremo de algum objetivo, ou seja, as que, dentre todas, levam ao maior ou menor valor de um objetivo, devido à sua importância, possuem pontuação infinita, sendo exceções à regra dada pela Equação (61) (PANICHELLA, 2019).

O processo para calcular a pontuação de todos os indivíduos da frente \mathcal{F}_1 consiste, inicialmente em obter conjunto E de todas as soluções que levam a valores extremos de objetivos e atribuir a cada uma a pontuação de sobrevivência \mathcal{E} infinita.

Em seguida, coloca-se no conjunto das soluções já pontuadas Λ , as soluções presentes em E e no conjunto das soluções ainda não pontuadas $\bar{\Lambda}$, coloca-se os demais elementos de \mathcal{F}_1 que não estejam contidos em E .

Calcula-se para todas as soluções ainda não pontuadas, sua proximidade φ segundo a Equação (56). Subsequentemente, calculam-se as distâncias γ dadas pela norma L_p entre cada solução de \mathcal{F}_1 e todas as demais soluções em \mathcal{F}_1 .

São então estabelecidas as diversidades ϑ de todas as soluções ainda não pontuadas conforme a Equação (58) com relação apenas às soluções que já foram pontuadas. Em seguida, estipula-se por meio Equação (59), as pontuações de sobrevivência temporárias v de todas as soluções ainda não pontuadas.

A solução que possui a maior pontuação de sobrevivência temporária é considerada pontuada e o valor de sua pontuação passa a ser definitivo \mathcal{E} . Esta solução é então adicionada ao conjunto de soluções pontuadas Λ e removida do conjunto de soluções não pontuadas $\bar{\Lambda}$.

O processo a partir do cálculo das diversidades é repetido até que não exista ne-

nhuma solução não pontuada, ou seja, até que $\bar{A} = \emptyset$.

O método de cálculo da pontuação de sobrevivência dos indivíduos das demais frentes é bastante simples. A pontuação de sobrevivência \mathcal{E} , nestes casos, é dada diretamente pelo inverso da proximidade do elemento em questão, conforme enuncia a Equação (60).

O Algoritmo 12 sintetiza o processo de cálculo da pontuação de sobrevivência dos indivíduos de uma frente, onde \mathcal{S} é a frente da qual os indivíduos serão pontuados, F identifica qual a frente fornecida, $\bar{f}(\cdot)$ é a função objetivo vetorial normalizada, ρ é o parâmetro calculado pelo Algoritmo 11.

3.2.2.6 Algoritmo Principal do AGE-MOEA

Apresentados os conceitos fundamentais, pode-se, agora, definir o algoritmo AGE-MOEA, que consiste, inicialmente, em gerar uma população inicial de progenitores \mathcal{A}_0 com q indivíduos. Depois, esta população é ordenada em frentes não dominadas, a fim de obter o ranque \mathcal{R} de cada um de seus indivíduos. Subsequentemente, aplica-se uma seleção por torneio sobre os indivíduos de \mathcal{A}_0 , considerando o ranque de cada um como valor de aptidão, que definirá o conjunto de pais \mathcal{A}_0^* que participarão do processo reprodutivo. Em seguida, são executados operadores de cruzamento e de mutação sobre os indivíduos de \mathcal{A}_0^* de modo a gerar uma população \mathcal{D}_0 de q descendentes. Estas populações constituem a geração inicial ou geração zero ($g = 0$) do algoritmo.

A população de pais \mathcal{A}_g é unida a de filhos \mathcal{D}_g em uma única população \mathcal{P}_g . Em seguida, ordena-se os indivíduos em frentes não dominadas $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \dots, \mathcal{F}_w$, calculando-se também o ranque \mathcal{R} de cada indivíduo.

Os objetivos do problema de otimização são então normalizados com base na frente \mathcal{F}_1 , resultando na função objetivo normalizada $\bar{f} = \langle \bar{f}_1, \bar{f}_2, \bar{f}_3, \dots, \bar{f}_m \rangle$. Depois, a geometria da curva de Pareto é estimada com base na frente \mathcal{F}_1 através do cálculo do parâmetro ρ .

A seguir, inicia-se a população de progenitores da nova geração \mathcal{A}_{g+1} vazia. Esta população deve ter exatamente q indivíduos. Para preenchê-la, vão sendo inseridos os indivíduos das frentes não dominadas, começando pela frente \mathcal{F}_1 e indo até se encontrar uma frente \mathcal{F}_z cujo o número de elementos seja superior às vagas restantes para completar os q indivíduos de \mathcal{A}_{g+1} . Também é definida a pontuação de sobrevivência para todos indivíduos das frentes $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \dots, \mathcal{F}_z$.

Para completar as vagas restantes em \mathcal{A}_{g+1} , integralizando q indivíduos, ordena-se os indivíduos da fronteira \mathcal{F}_z do maior para o menor, comparando-os pelo valor de sua pontuação de sobrevivência, e em seguida, esses indivíduos vão sendo acrescidos à \mathcal{A}_{g+1} , partindo do primeiro até chegar àquele que completará os q indivíduos de \mathcal{A}_{g+1} . Os indivíduos restantes da fronteira \mathcal{F}_z , bem como os das demais fronteiras são descartados.

Realiza-se então uma seleção por torneio sobre os indivíduos de \mathcal{A}_{g+1} , considerando

Algoritmo 12: Pontuação de sobrevivência.

Entrada: $\mathcal{S}, F, \bar{f}(\cdot), \rho$
Saída: \mathcal{E}

```

1 PONTUACAO_DE_SOBREVIVENCIA( $\mathcal{S}, F, \bar{f}(\cdot), \rho$ )
2 início
3   se  $F = 1$  então
4      $E \leftarrow \emptyset$ ;
5     para  $i = 1$  até  $m$  faça
6        $x_i^{min} \leftarrow \left\{ \text{Minimizar}_x \bar{f}_i(x) \text{ Sujeito a } x \in \mathcal{S}; \right.$ 
7        $x_i^{max} \leftarrow \left\{ \text{Maximizar}_x \bar{f}_i(x) \text{ Sujeito a } x \in \mathcal{S}; \right.$ 
8        $E = E \cup \{x_i^{min}\} \cup \{x_i^{max}\}$ ;
9        $\mathcal{E}_{x_i^{min}} \leftarrow \mathcal{E}_{x_i^{max}} \leftarrow \infty$ ;
10    fim
11     $\Lambda \leftarrow E$ ;
12     $\bar{\Lambda} \leftarrow \mathcal{S} - E$ ;
13    para cada  $x \in \bar{\Lambda}$  faça
14       $\varphi_x \leftarrow \|\bar{f}(x)\|_\rho$ ;
15    fim
16    para todo  $x \in \mathcal{S}$  faça
17      para cada  $y \in \mathcal{S}$  faça
18         $\gamma_x^y \leftarrow \|\bar{f}(x) - \bar{f}(y)\|_\rho$ ;
19      fim
20    fim
21    enquanto  $\aleph(\bar{\Lambda}) > 0$  faça
22      para cada  $x \in \bar{\Lambda}$  faça
23         $\gamma^{\circ min} \leftarrow \min_{y \in \Lambda} (\gamma_x^y)$ ;
24         $y^* \leftarrow \left\{ \text{Minimizar}_y \gamma_x^y \text{ Sujeito a } y \in \Lambda; \right.$ 
25         $\gamma^{\circ min} \leftarrow \min_{\substack{y \in \Lambda \\ y \neq y^*}} (\gamma_x^y)$ ;
26         $\vartheta_x \leftarrow \gamma^{\circ min} + \gamma^{\circ min}$ ;
27         $v_x \leftarrow \frac{\vartheta_x}{\varphi_x}$ ;
28      fim
29       $x^* \leftarrow \left\{ \text{Maximizar}_x v_x \text{ Sujeito a } x \in \bar{\Lambda}; \right.$ 
30       $\mathcal{E}_{x^*} \leftarrow v_{x^*}$ ;
31       $\Lambda \leftarrow \Lambda \cup \{x^*\}$ 
32       $\bar{\Lambda} \leftarrow \bar{\Lambda} - \{x^*\}$ 
33    fim
34  else
35    para cada  $x \in \mathcal{S}$  faça
36       $\mathcal{E}_x \leftarrow \frac{1}{\|\bar{f}(x)\|_\rho}$ 
37    fim
38  end
39  retorna  $\mathcal{E}$ ;
40 fim

```

a pontuação de sobrevivência de cada um como valor de aptidão, estabelecendo o conjunto de pais \mathcal{A}_{g+1}^* que participarão do processo reprodutivo.

Os operadores de cruzamento de mutação são executados sobre os indivíduos de \mathcal{A}_{g+1}^* de modo a gerar a população de filhos \mathcal{D}_{g+1} . Os indivíduos de \mathcal{A}_{g+1} e \mathcal{D}_{g+1} são unidos em uma única população e o processo se repete até alcançar o número desejado de gerações (g_{max}).

O Algoritmo 13 sintetiza o método AGE-MOEA, onde $\aleph(\cdot)$ denota o operador de cardinalidade de conjuntos.

3.2.2.7 Emprego do AGE-MOEA em Problemas de Otimização Restrita

As considerações acerca do emprego do AGE-MOEA em problemas de otimização restrita, bem como a forma de abordagem deste algoritmo para este tipo de problema são as mesmas do método NSGA-II, que foram apresentadas na Seção 3.2.1.6.

3.3 Referências com Informações Complementares

Ao longo deste capítulo, discorreu-se acerca dos métodos metaheurísticos voltados tanto à resolução de problemas de otimização mono-objetivo quanto à solução de problemas de otimização multibjetivo, bem como, apresentou-se alguns métodos de otimização de cada um destes tipos, mostrando seus aspectos, peculiaridades e dinâmica de funcionamento.

Todavia, há ainda uma vasta gama de métodos de otimização contidos na literatura, que não foram enunciados neste trabalho. Cada um destes métodos possuem suas próprias características, vantagens e desvantagens. Além disso, existem variações que podem ser aplicadas aos algoritmos apresentados, alterando as propriedades e aplicabilidade dos mesmos, que não foram mostradas neste capítulo.

Se houver anseio em conhecer outros conceitos concernentes aos métodos de otimização, bem como, explorar outros métodos e obter maior aprofundamento no assunto, ficam recomendados os trabalhos da literatura a seguir:

- Rao (2009);
- Bazaraa, Sherali e Shetty (2006);
- Ferraz (2021);
- Fletcher (2000);
- Correia, Matias e Serodio (2008);

Algoritmo 13: AGE-MOEA.

Entrada: q, g_{max}
Saída: \mathcal{A}_g

- 1 $AGE - MOEA(q, g_{max})$
- 2 **início**
- 3 $\mathcal{A}_0 \leftarrow \mathcal{D}_0 \leftarrow \emptyset;$
- 4 $\mathcal{A}_0 \leftarrow (q \text{ indivíduos gerados aleatoriamente});$
- 5 $(_, \mathcal{R}) \leftarrow ORDENACAO_EM_FRENTE_NAO_DOMINADAS(\mathcal{A}_0)$
- 6 $\mathcal{A}_0^* \leftarrow (\text{indivíduos de } \mathcal{A}_0 \text{ selecionados por torneio, cujas aptidões são dadas por } \mathcal{R});$
- 7 $\mathcal{D}'_0 \leftarrow (q \text{ indivíduos obtidos pelo cruzamento entre os elementos de } \mathcal{A}_0^*);$
- 8 $\mathcal{D}_0 \leftarrow (q \text{ indivíduos obtidos pela mutação dos elementos de } \mathcal{D}'_0);$
- 9 **para** $g = 0$ **até** g_{max} **faça**
- 10 $\mathcal{P}_g \leftarrow \mathcal{A}_g \cup \mathcal{D}_g;$
- 11 $(\mathcal{F}, _) \leftarrow ORDENACAO_EM_FRENTE_NAO_DOMINADAS(\mathcal{P}_g);$
- 12 $\bar{f}(\cdot) \leftarrow NORMALIZAR_OBJETIVOS(\mathcal{F}_1);$
- 13 $\rho \leftarrow GEOMETRIA_DA_FRENTE(\mathcal{F}_1, \bar{f}(\cdot));$
- 14 $\mathcal{A}_{g+1} \leftarrow \emptyset;$
- 15 $\iota \leftarrow 1;$
- 16 **enquanto** $\aleph(\mathcal{A}_{g+1}) + \aleph(\mathcal{F}_\iota) \leq q$ **faça**
- 17 $\mathcal{E}^\iota \leftarrow PONTUACAO_DE_SOBREVIVENCIA(\mathcal{S}, \iota, \bar{f}(\cdot), \rho);$
- 18 $\mathcal{A}_{g+1} \leftarrow \mathcal{A}_{g+1} \cup \mathcal{F}_\iota;$
- 19 $\iota \leftarrow \iota + 1;$
- 20 **fim**
- 21 $\mathcal{E}^\iota \leftarrow PONTUACAO_DE_SOBREVIVENCIA(\mathcal{S}, \iota, \bar{f}(\cdot), \rho);$
- 22 $\varsigma \leftarrow (\text{soluções } x \in \mathcal{F}_\iota \text{ ordenadas através da comparação pelos valores de } \mathcal{E}^\iota_x);$
- 23 $j \leftarrow 1;$
- 24 **enquanto** $\aleph(\mathcal{A}_{g+1}) < q$ **faça**
- 25 $\mathcal{A}_{g+1} \leftarrow \mathcal{A}_{g+1} \cup \{\varsigma_j\};$
- 26 $j \leftarrow j + 1;$
- 27 **fim**
- 28 **para** $\kappa = 1$ **até** ι **faça**
- 29 **para todo** $x \in \mathcal{E}^\kappa$ **faça**
- 30 $\bar{\mathcal{E}}_x \leftarrow \mathcal{E}_x^\kappa;$
- 31 **fim**
- 32 **fim**
- 33 $\mathcal{A}_{g+1}^* \leftarrow (\text{indivíduos de } \mathcal{A}_{g+1} \text{ selecionados por torneio, cujas aptidões são dadas por } \bar{\mathcal{E}});$
- 34 $\mathcal{D}'_{g+1} \leftarrow (q \text{ indivíduos obtidos pelo cruzamento entre os elementos de } \mathcal{A}_{g+1}^*);$
- 35 $\mathcal{D}_{g+1} \leftarrow (q \text{ indivíduos obtidos pela mutação dos elementos de } \mathcal{D}'_{g+1});$
- 36 **fim**
- 37 **retorna** $\mathcal{A}_g;$
- 38 **fim**

- Luke (2016);
- Glover e Kochenberger (2003);
- Das e Suganthan (2011);
- Storn e Price (1997b);
- Ávila (2020);
- Miettinen (1994);
- Andersson (2001);
- Ferreira (1999);
- Collette e Siarry (2003);
- Steuer (1986);
- Evans (1984);
- Hwang *et al.* (1979);
- Deb (2001);
- Coello (1999);
- Abraham, Jain e Goldberg (2005);
- Purshouse e Fleming (2007);
- Coello (2006);
- Srinivas e Deb (1995);
- Deb *et al.* (2002);
- Panichella (2019);

4 FERMENTAÇÃO ALCOÓLICA

Nunca ande por trilhas, pois assim só irá até onde outros já foram.

Alexander Graham Bell

Alguns dos processos tratados nesta tese são processos de fermentação alcoólica. Assim, para que se possa compreender tais problemas, se faz necessário conhecer a fermentação etanólica, sua importância e características, além dos tipos de processo fermentativo, bioquímica e modelagem matemática da fermentação, assuntos que são abordados neste capítulo.

O uso da fermentação remonta aos primórdios da humanidade, sendo usado pelo antigos egípcios, sumérios, babilônicos e assírios para produção de alimentos fermentados. O uso do processo fermentativo para produção de bebidas remonta ao ano 6000 A.C. (VILLEN, 2009).

No século XIX, o cientista francês Louis Pasteur, cujas descobertas foram de extrema importância para a química e a medicina, promoveu estudos voltados para a cultura, desenvolvimento e ação de microrganismos em alimentos lácteos, como queijos e iogurtes, em bebidas alcoólicas, como vinho, e em combustíveis. Por meio destes estudos, identificou-se que o processo de fermentação estava intimamente ligado à ação dos microrganismos (AIBA; HUMPHREY; MILLIS, 1973; BAILEY; OLLIS, 1986).

Descobriu-se também que os substratos, especialmente os açúcares, eram metabolizados pelos microrganismos, gerando os produtos da fermentação. Por meio de tais descobertas, o campo da biotecnologia foi consideravelmente expandido.

Os grandes passos no desenvolvimento das técnicas para fermentação de etanol foram dados, na década de 1930, especialmente pela França e Alemanha, que construíram colunas de destilação alcoólicas voltadas para fins farmacêuticos, produção de bebidas e de etanol para queima, com o intento de gerar energia térmica (LIMA; BASSO; AMORIM, 2001).

4.1 Ascensão e Consolidação da Indústria Alcooleira

Na crise financeira de 1929, a quebra da bolsa de valores de Nova York afetou diretamente a comercialização do açúcar. Assim, havia excedente de cana-de-açúcar e de etanol no mercado, uma vez que não existia demanda suficiente para toda a produção. Por conta disto, no ano de 1931, foi instalada a primeira planta de produção de álcool anidro e a gasolina passou a conter em sua composição 5% de etanol (LIMA; BASSO;

AMORIM, 2001).

No ano de 1973, aconteceu a primeira crise do petróleo em virtude do aumento de 300% no preço do barril pela Organização dos Países Exportadores de Petróleo (OPEP) como forma de protesto contra o apoio dos Estados Unidos à Israel na guerra de Yom Kippur (UNICA, 2012).

Esse aumento prejudicou severamente a economia brasileira, que precisou elevar de 600 milhões para 2,5 bilhões de dólares, o custo com importação de petróleo. Neste cenário, foi criado o Programa Nacional do Álcool (PROÁLCOOL) que promoveu um massivo incentivo para que os combustíveis derivados do petróleo fossem substituídos, em larga escala, pelo etanol (UNICA, 2012).

Em 2003, a introdução dos veículos com motores que operavam tanto com gasolina, quanto com álcool impulsionou consideravelmente a produção alcooleira no Brasil. Esse tipo de veículo chegou a representar 57% da vendas totais de veículos em 2005 (HIRA; OLIVEIRA, 2009).

Toda esta sucessão de eventos corroborou para que, atualmente, o Brasil figurasse como um dos maiores produtores de etanol do mundo (RFA, 2022).

4.2 Etapas do Processo de Produção Alcoólica

De acordo com Nova Cana (2020), O processo de produção alcooleira é constituído de sete etapas elementares, que abarcam desde a chegada da matéria prima até o armazenamento para distribuição. Cada uma destas etapas está citada abaixo.

- Lavagem;
- Moagem;
- Purificação;
- Fermentação;
- Destilação;
- Desidratação;
- Armazenagem.

A seguir, cada uma das etapas acima mencionadas será brevemente apresentada e descrita.

4.2.1 Lavagem

A matéria prima, que pode ser, por exemplo, cana-de-açúcar ou milho, chega às plantas de produção em sua forma bruta, tal como foi colhida. Essa matéria prima é submetida a um processo de lavagem que visa remover poeira, terra, areia e outras impurezas que estejam presentes nela.

A seguir, a matéria prima é fracionada e passa sob um eletroímã, que atrai e retira quaisquer resíduos e materiais metálicos dela. Após esta etapa, a matéria prima segue para a Moagem.

4.2.2 Moagem

Nesta etapa, a matéria prima é submetida a moendas, extraindo-se o caldo que é chamado mosto. Aproximadamente 70% da matéria prima consiste neste caldo e os 30% restantes consistem no bagaço sólido.

O mosto gerado neste processo segue para a etapa de Purificação, enquanto o bagaço é destinado à geração de energia elétrica pela termoelétrica a biomassa anexa à usina, caso esta a possua.

4.2.3 Purificação

Nesta etapa, o mosto passa por uma filtragem para eliminar as impurezas de maiores dimensões e em seguida, segue para um tanque de decantação, onde ficará em repouso a fim de que as impurezas restantes fiquem depositadas no fundo do tanque.

Terminada a decantação, o mosto passa por uma esterilização, onde é aquecido a uma determinada temperatura a fim de eliminar microrganismos presentes nele para reduzir ao máximo a contaminação biológica.

4.2.4 Fermentação

Nesta etapa é onde de fato ocorre a formação do etanol. Por esta razão, de todas as etapas citadas, este é o processo de particular interesse desta tese, sobre o qual discorreremos amplamente, uma vez que é a geração de álcool que se deseja otimizar.

Descrevendo este processo de forma muito breve, temos que, o mosto é colocado em reatores, onde é misturado ao fermento biológico, que consiste em microrganismos, geralmente fungos chamados leveduras da espécie *Saccharomyces Cerevisiae*.

Esses microrganismos metabolizam os açúcares presentes no mosto, quebrando as moléculas de glicose, produzindo etanol e dióxido de carbono. Após este processo, o mosto estará transformado em um vinho fermentado que contém leveduras, açúcares não fermentados e álcool.

4.2.5 Destilação

Nesta etapa, o vinho obtido após a fermentação é posto em colunas de destilação, onde é aquecido até que o etanol evapore, separando-se do vinho, e em seguida o vapor de álcool é submetido a temperaturas mais baixas para que este se condense, voltando a ser líquido.

Neste processo, parte da água também evapora, de forma que, após a condensação, temos o chamado álcool hidratado, que consiste em cerca de 96% de etanol e 4% de água. Este é o álcool usado como combustível.

Se o objetivo da usina de produção for gerar apenas álcool hidratado, este segue diretamente para a fase de armazenamento, caso contrário segue para a desidratação.

4.2.6 Desidratação

Esta etapa objetiva remover o restante da água presente no álcool para que este atinja sua forma pura. Existem diversas técnicas que podem ser empregadas para a realização desta tarefa.

Uma destas técnicas consiste no emprego de solventes que se misturam com a água e ambos evaporam conjuntamente, deixando apenas o etanol. Existe também a técnica de filtração molecular, em que se empregam filtros que retêm apenas as moléculas de H_2O , permitindo a passagem das moléculas de etanol.

O etanol oriundo desta etapa é chamado álcool anidro, cuja composição é de aproximadamente 99,5% de etanol. Este é o álcool usado em mistura com a gasolina.

4.2.7 Armazenagem

Nesta fase, o álcool, tanto o anidro, quanto o hidratado, são estocados em reservatórios devidamente higienizados para posteriormente serem transportados até os distribuidores que, por sua vez, conduzirão o etanol para os próximos agentes da cadeia de consumo.

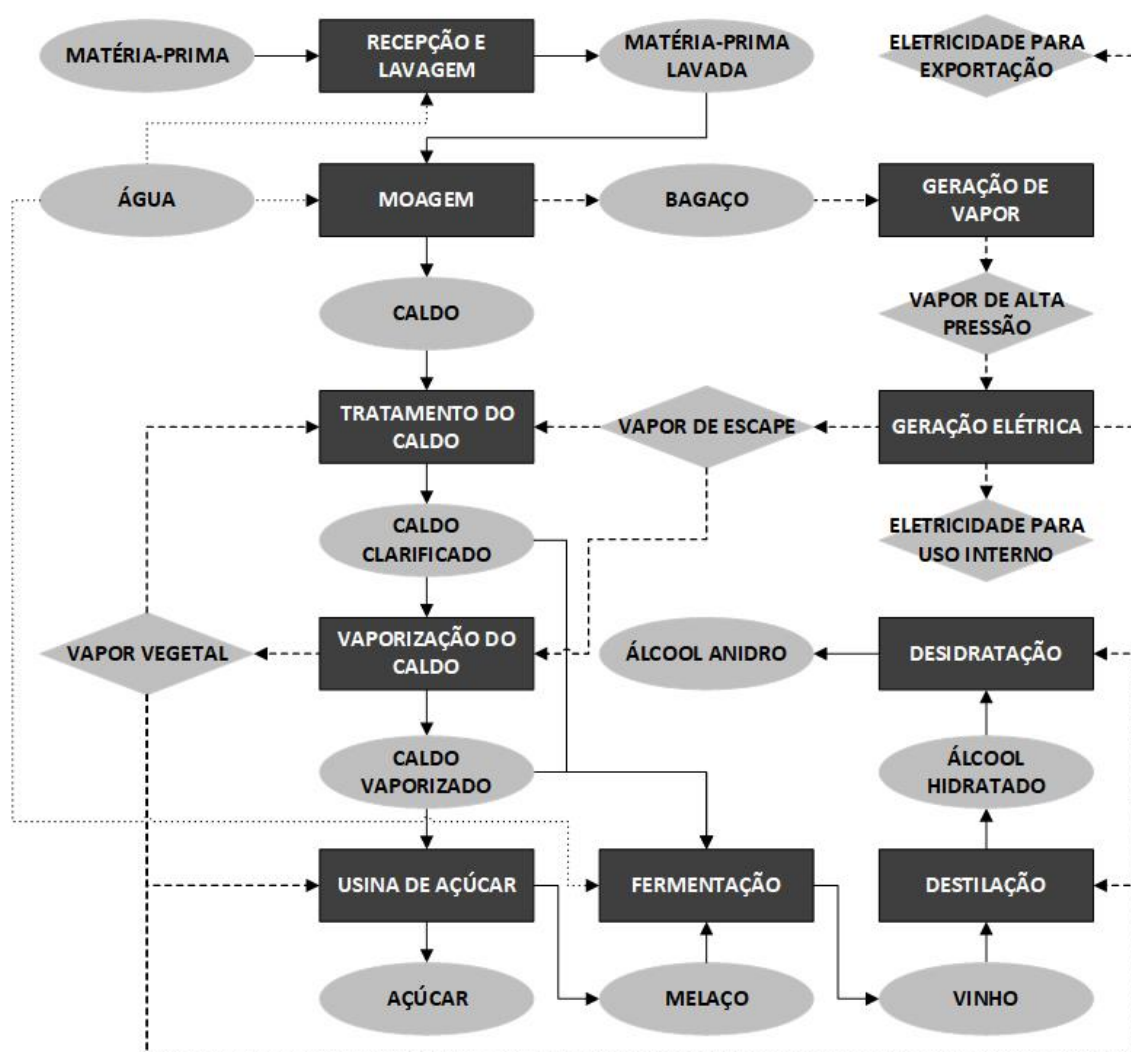
4.3 Modelos das Plantas Sucroalcooleiras do Brasil

O Brasil, desde o período colonial, voltou-se fortemente para a produção de açúcar e hoje, é um dos maiores produtores e o maior exportador da *commodity*. Por esta razão, cerca de 70% das instalações industriais sucroalcooleiras são plantas de produção de açúcar com destilarias anexas. As demais 30% são plantas de destilação autônomas voltadas à produção de etanol e bebidas (HORTA, 2004).

Dessa forma, predomina no Brasil o processo integrado de produção de etanol e açúcar. As plantas de produção também incorporaram a produção de energia elétrica através de termoeletricas a biomassa (HENDLER, 2011).

A Figura 30 ilustra o fluxo dos principais processos de uma planta sucroalcooleira do tipo acima descrita. Boa parte dos processos é a mesma tanto para a produção de açúcar, quanto para a produção de etanol (HENDLER, 2011).

Figura 30 - Principais processos de uma planta sucroalcooleira.



Fonte: O autor, 2023.

4.4 Tipos de Processos Fermentativos

Para se estabelecer um processo fermentativo, seja em pequena ou grande escala, se faz necessário escolher o método que será usado para que a fermentação e produção de etanol aconteça.

Existem três processos fermentativos que são amplamente abordados na literatura e predominantemente aplicados na prática. São eles:

- Fermentação Contínua;
- Fermentação em Batelada;
- Fermentação em Batelada Alimentada.

Todos estes tipos de processo fermentativo podem ou não ter recirculação de fermento em seu ciclo (VILELA, 2015). Nas subseções a seguir, serão apresentados e detalhados os métodos supracitados.

4.4.1 Fermentação em Batelada

De acordo com Porto (2005), o processo de fermentação em batelada, também denominado fermentação descontínua, consiste, inicialmente, em preparar um meio de cultura com nutrientes que permitam o desenvolvimento dos microrganismos. Neste caso, o meio de cultura consiste no substrato contido no mosto de cana-de-açúcar, milho ou de outra matéria prima.

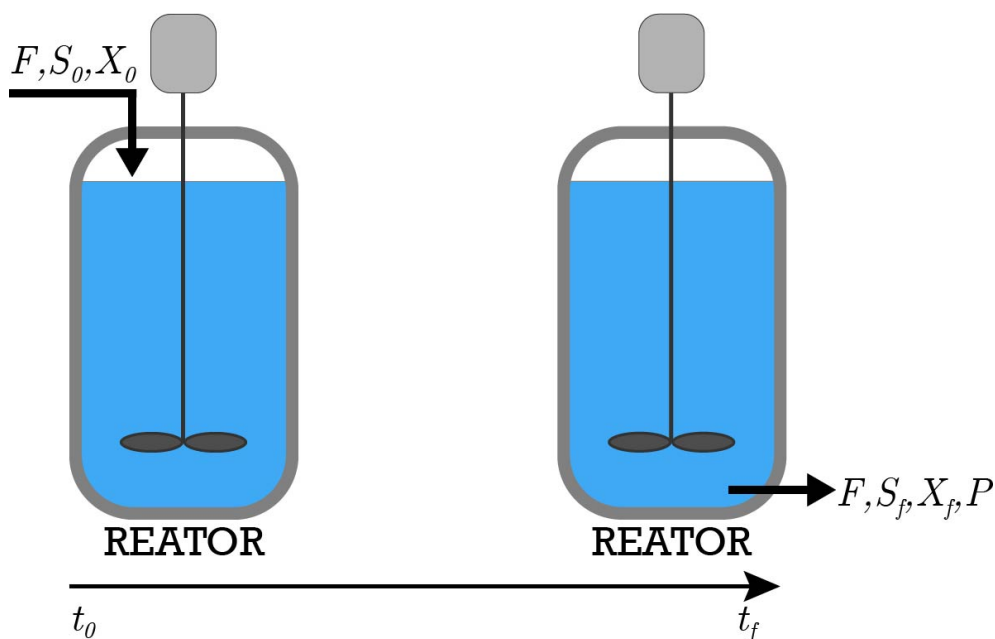
O próximo passo é introduzir o meio de cultura a um reator e adicionar a este, um concentrado de colônias de microrganismos que realizarão o processo biológico de fermentação. Após isto, aguarda-se o tempo necessário para que este processo ocorra.

Finalmente, extrai-se do reator todo o suco fermentado e aplicam-se as etapas que efetuam a separação do produto desejado, que neste caso é etanol. A Figura 31 ilustra esquematicamente o processo de fermentação em batelada, onde F denota a vazão, S_0 a concentração inicial de substrato, X_0 a concentração inicial de microrganismos, t_0 o tempo inicial, S_f a concentração final de substrato, X_f a concentração final de microrganismos, t_f o tempo final e P a concentração de produto.

4.4.1.1 Modelo Prático de Fermentação em Batelada Alimentada

Na prática, o modelo fermentativo comumente adotado por laboratórios e pequenas refinarias, consiste em inserir o mosto e a levedura no reator, aguardando, em seguida,

Figura 31 - Fermentação em batelada.



Fonte: O autor, 2023.

que o processo termine.

A seguir, o suco fermentado segue para um tanque e a partir deste, é enviado para uma centrífuga que separa a levedura do vinho fermentado. A Figura 32 ilustra este processo sem a recirculação de células.

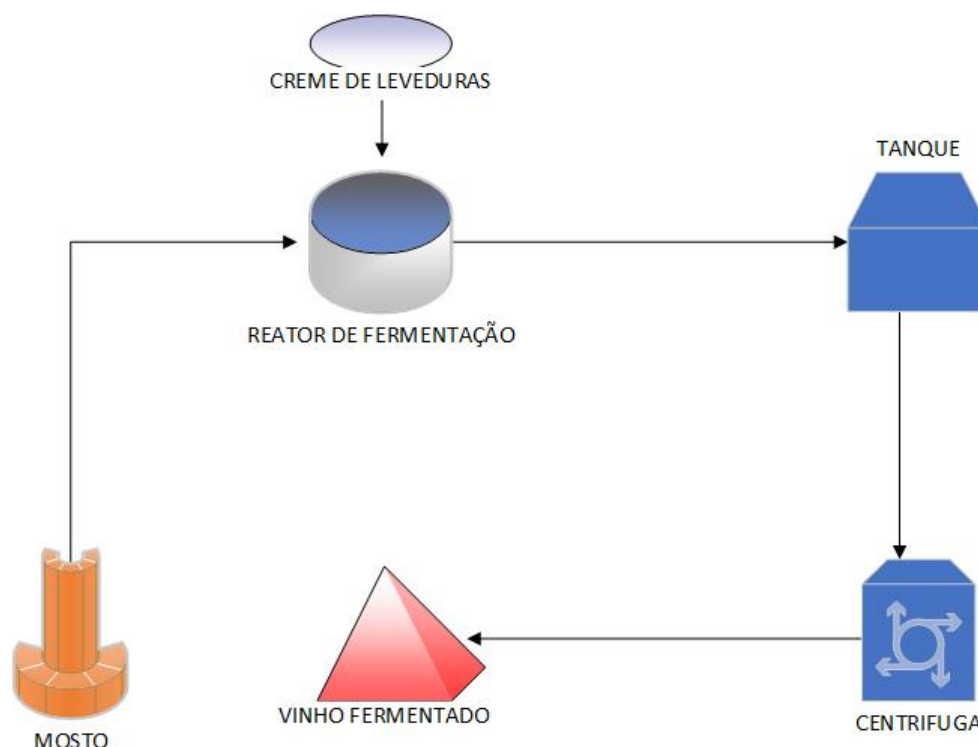
4.4.1.2 Vantagens da Fermentação em Batelada

Uma das problemáticas dos processos fermentativos é a contaminação biológica, que pode acometer o creme de microrganismos, acarretando em consideráveis custos para a minimização desta. Neste contexto, a fermentação em batelada apresenta uma vantagem, que é a assepsia.

Ao final do ciclo de fermentação, conhecido neste tipo de processo como batelada, o reator é esterilizado junto com o meio de cultura, recebendo em seguida um novo fermento devidamente tratado. Assim, é possível controlar e garantir a presença apenas do microrganismo responsável pela fermentação (SCHMIDELL; FACCIOTTI, 2001).

A fermentação em batelada apresenta também maior facilidade de manutenção, por apresentarem reatores independentes e flexibilidade de operação, uma vez que é possível usar os reatores para produzir diferentes produtos (CARVALHO; SATO, 2001).

Figura 32 - Modelo prático da fermentação em batelada.



Fonte: O autor, 2023.

4.4.1.3 Problemáticas da Fermentação em Batelada

Segundo Carvalho e Sato (2001), como o substrato é adicionado por completo no início do processo, conforme ocorre a fermentação, existe uma inibição do metabolismo dos microrganismos ou um desvio deste à geração de outros produtos de fermentação que não são interessantes. Este problema leva o processo a apresentar baixo rendimento e produtividade.

O processo de fermentação em batelada também é mais lento, pois a cada batelada de fermentação, o reator deve ser higienizado e preparado; e o mosto e creme de microrganismos devem ser carregados nele (MAIORELLA; BLANCH; WILKE, 1981).

Em razão destes problemas, o processo de fermentação em batelada é empregado apenas em escala laboratorial ou em refinarias de pequeno porte (PACHECO, 2010).

4.4.2 Fermentação em Batelada Alimentada

O processo de fermentação em batelada alimentada, fermentação descontínua alimentada ou ainda, fermentação Melle-Boinot, é uma evolução da fermentação em batelada, que buscou eliminar o problema de ineficiência presente nesta através da injeção

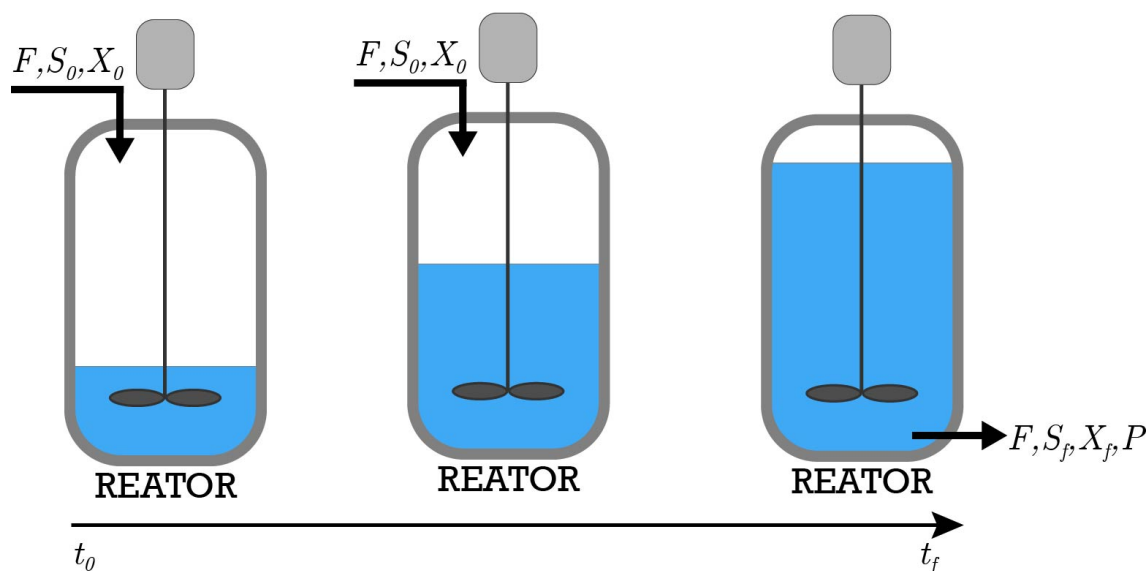
controlada de substrato ao reator até que este atinja seu volume normal de operação.

De acordo com Vilela (2015), a fermentação em batelada alimentada consiste inicialmente na inserção do creme de microrganismos no reator devidamente esterilizado para que não exista contaminantes presentes neste.

Por conseguinte, coloca-se, continuamente ou intermitentemente, os substratos, de forma gradativa no reator. A vazão de entrada de substrato no reator pode ser contínua ou pode variar conforme o instante de tempo. Durante a inserção dos substratos, o processo fermentativo já está sendo realizado pelos microrganismos.

Após o preenchimento do reator, espera-se o tempo restante necessário para que os microrganismos metabolizem todo o substrato, gerando o produto final esperado pelo processo de fermentação, neste caso, o etanol. A Figura 33 ilustra esquematicamente o processo de fermentação em batelada alimentada.

Figura 33 - Fermentação em batelada alimentada.



Fonte: O autor, 2023.

4.4.2.1 Processo Industrial de Fermentação em Batelada Alimentada

O processo geralmente empregado nas usinas que produzem etanol consiste na colocação das leveduras no reator e em seguida, um alimentador injeta de forma contínua ou intermitente o mosto neste reator, onde o processo fermentativo já está ocorrendo.

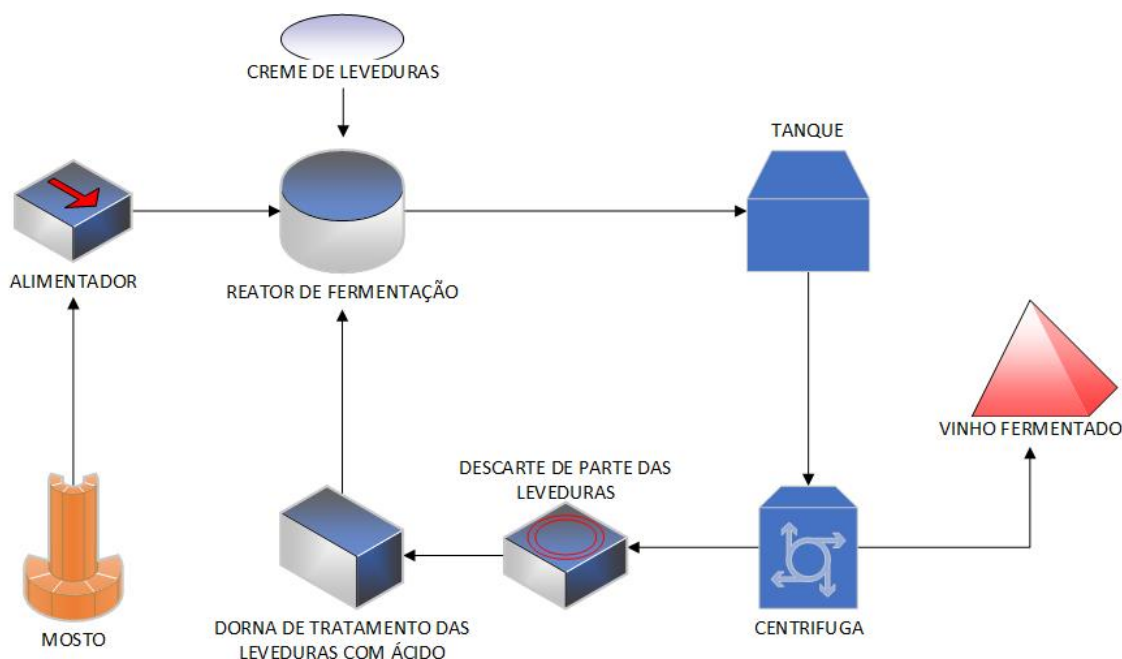
Após o reator estar cheio, espera-se o tempo restante para o término da fermentação e em seguida o suco fermentado segue para um tanque, que a partir deste vai sendo enviado a uma centrífuga que separa a levedura do vinho fermentado.

Em escala industrial, por conta do custo benefício, emprega-se a recirculação de

células, processo em que o creme de levedura separado após a centrifugação tem parte descartada e o restante é submetido a um procedimento com ácido para tratar a contaminação bacteriana, sendo em seguida, colocado novamente no reator.

A Figura 34 ilustra o processo descrito acima.

Figura 34 - Processo industrial de fermentação em batelada alimentada.



Fonte: O autor, 2023.

4.4.2.2 Vantagens e Desvantagens da Fermentação em Batelada Alimentada

A injeção do substrato no reator de maneira intermitente ou contínua reduz consideravelmente a inibição da fermentação do substrato pelos microrganismos, resolvendo o problema de ineficiência que possui o processo em batelada.

Contudo, ao final do processo fermentativo, o efeito de inibição da fermentação provocado pelo próprio etanol produzido não é reduzido ou abrandado neste processo, quando comparado ao processo em batelada (VILELA, 2015).

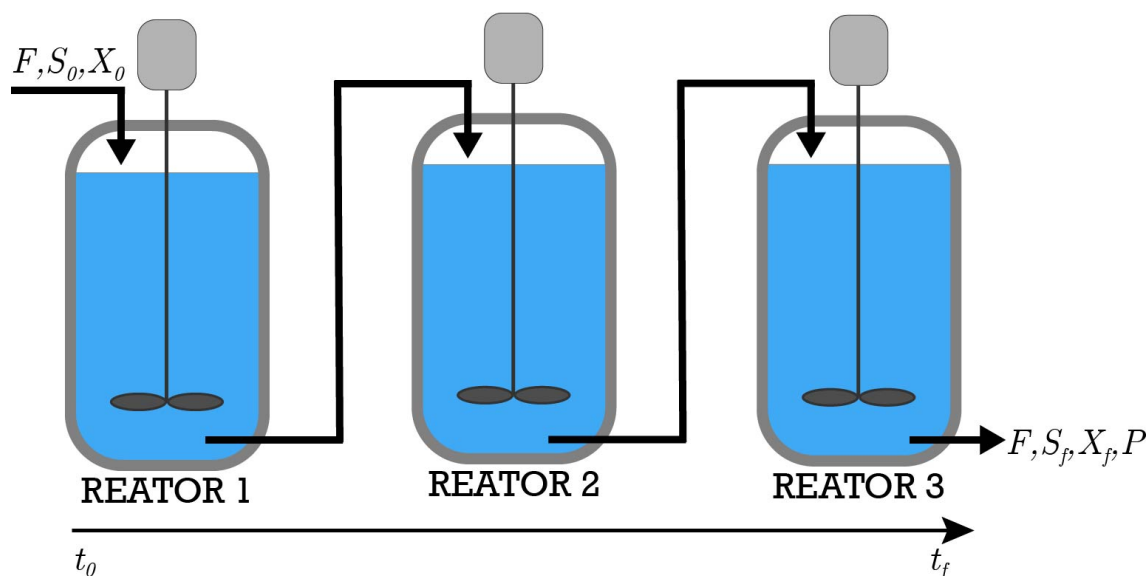
Este processo tem ainda como vantagens, maior facilidade na assepsia do reator e demais equipamentos envolvidos no processo e a possibilidade de operar com alta concentração de substratos, o que confere um aumento na produção de etanol, mesmo trabalhando com volume de reator e vinho reduzidos (VAN *et al.*, 1994).

4.4.3 Fermentação Contínua

O processo de fermentação contínua foi proposto na década de 80 pelo Centro de Tecnologia Canavieira (CTC), no Brasil. A concepção deste modelo de fermentação tem base na ideia de se operar com um ou mais reatores ligados em série, que são alimentados continuamente por substratos, permanecendo sempre cheios durante o processo de fermentação (CNPEM, 2017).

Segundo Porto (2005), o processo de fermentação contínua consiste na injeção contínua, em vazão constante, de substrato no meio de cultura, composto por um ou mais reatores conectados em série, tendo o volume de fermentação constante, mantido por meio da extração contínua do suco fermentado. A Figura 35 ilustra esquematicamente o processo de fermentação contínua num sistema com três reatores ligados em série.

Figura 35 - Fermentação contínua.



Fonte: O autor, 2023.

4.4.3.1 Configurações de Arranjo de Reatores na Fermentação Contínua

Conforme Borzani (1960), existem duas configurações elementares de arranjo de reatores que podem ser adotadas no processo de fermentação contínua. A primeira consiste no emprego de apenas um reator no processo. Essa primeira configuração é adotada quando a velocidade do desenvolvimento dos microrganismos é praticamente invariante diante de diferentes concentrações de substrato, fazendo com que a velocidade de fermentação, quando considerado o uso de mais de um reator em série, seja a mesma em todos os reatores, não havendo vantagem em se trabalhar com mais de um reator.

A segunda configuração consiste no emprego de dois ou mais reatores conectados em série. Essa configuração é adotada quando a velocidade de desenvolvimento dos microrganismos no primeiro reator é maior que segundo, que por sua vez, é maior do que terceiro e assim sucessivamente. Desse modo, os fatores que provocam a queda do desenvolvimento dos microrganismos ficarão isolados nos últimos reatores não incidindo sobre os primeiros, conferindo elevado rendimento e produtividade ao processo.

Quando se considera a fermentação contínua de etanol, a segunda configuração é adotada, uma vez que conforme ocorre o processo fermentativo, a quantidade de etanol presente no suco aumenta e este é um inibidor de microrganismos. Usando a segunda configuração, o teor alcoólico que inibirá a velocidade de reprodução dos microrganismos se encontrará nos últimos reatores, não interferindo nos primeiros e ainda concedendo o tempo necessário para se obter o máximo de produção possível (BORZANI, 1960).

4.4.3.2 Processo Industrial de Fermentação Contínua

O processo comumente adotado nas plantas de produção alcoólicas consiste na inserção contínua de mosto e creme de leveduras em um primeiro reator, onde parte do processo fermentativo ocorre. O suco do primeiro reator vai seguindo constantemente para o segundo reator, onde outra parte da fermentação acontece, e assim sucessivamente, até que o suco chegue no último reator, onde o processo de fermentação terminará.

A seguir, o suco fermentado segue para uma centrífuga que separa a levedura do vinho fermentado. Parte das leveduras oriundas da centrifugação são descartadas e parte são recicladas por um procedimento com ácido para tratar a contaminação biológica, seguindo logo após para o primeiro reator.

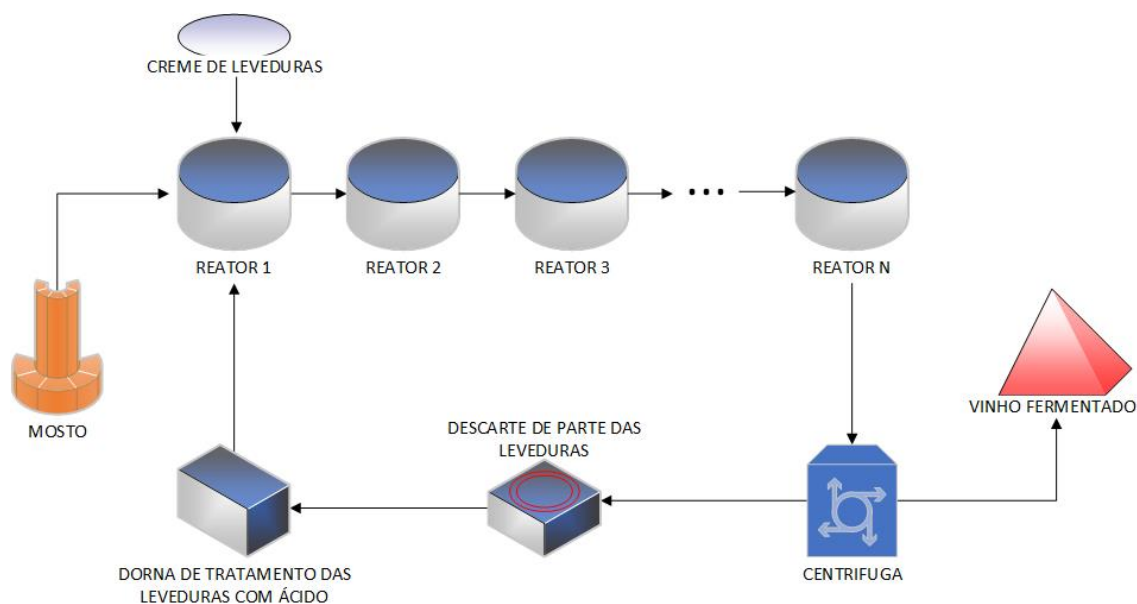
A Figura 34 ilustra o processo descrito com N reatores.

4.4.3.3 Vantagens e Problemáticas da Fermentação Contínua

Por conta do sistema operar em estado estacionário, existe uma maior simplicidade para se empregar sistemas de controle mais precisos, menor necessidade de mão-de-obra para operá-lo e maior uniformidade no suco fermentado extraído, mas em contrapartida, requer um maior investimento inicial, quando comparado ao sistema de fermentação em batelada alimentada (VILELA, 2015).

Um fator decisivo para que o sistema atinja o regime permanente, onde as variáveis de estado, tais como quantidade de substrato, concentração de microrganismos e porção do produto gerado, como etanol, invariarem no tempo, é manter um volume de líquido constante nos reatores (PORTO, 2005).

Figura 36 - Processo industrial de fermentação contínua.



Fonte: O autor, 2023.

Porém, para manter um volume constante de líquido nos reatores é necessário idealmente que as vazões de entrada de insumos e saída de produtos do sistema sejam as mesmas. Manter essas vazões equalizadas é inviável em um sistema real. Apesar desta problemática, o processo de fermentação contínua tem aplicações práticas reais (PORTO, 2005).

O processo de fermentação contínua com múltiplos estágios tem por vantagens, a possibilidade de se operar com altos teores alcoólicos e esgotamento total do mosto. Por desvantagens, o processo tem a complexidade de operar e controlar múltiplos estágios de fermentação (EDUARDO, 1991).

Segundo Zarpelon e Andrietta (1992) modelo de fermentação contínua para etanol apresenta maior estabilidade que o em batelada, uma vez que os efeitos de inibição de fermentação ocorrem apenas nos últimos reatores, conferindo maior produtividade alcoólica.

O processo de fermentação contínua admite a recirculação de microrganismos, que confere alta densidade celular no sistema, podendo contribuir para ampliar a taxa de produção de etanol (FACCIOTTI, 2001).

4.4.4 Contraposição Entre os Tipos de Processos Fermentativos

O sistema de fermentação contínua tem por sua grande vantagem, a redução de custos de implantação, manutenção, automação e operação. Estes custos podem ficar em

torno de 50% a 60% mais baratos que os do sistema em batelada alimentada, uma vez que não se precisa de grandes áreas para a sua instalação. Além disso, o sistema contínuo tem menor custo de automação e é mais barato por ter menor número de reatores (CNPEM, 2017).

A fermentação em batelada alimentada, quando comparada a processo contínuo, possui alto custo de instalação, automação e de manutenção, devido, em parte, ao uso de vários reatores, cada um realizando individualmente o processo, em paralelo.

Entretanto, o sistema contínuo apresenta sérios problemas com relação à infecções biológicas, que ocasionam uma considerável redução no o rendimento da fermentação, o que provoca uma elevação no custo de produção, uma vez que a planta precisa usar muito mais insumos, tais como ácido sulfúrico e antibióticos.

Por outro lado, a fermentação em batelada alimentada possui maior facilidade e viabilidade de higienização de seu sistema, o que confere maior controle microbiológico, acarretando em um rendimento consideravelmente maior e em um investimentos bem menor em insumos para a produção.

Dessa forma, constata-se que, a longo prazo, o processo em batelada apresenta um maior rendimento e menor custo de produção. Com intuito de verificar qual o processo fermentativo possui maior eficiência produtiva, Godoy, Henrique e Lopes (2008) realizaram um estudo que considerou sessenta e duas plantas de fermentação, sendo cinquenta e uma unidades de fermentação em batelada alimentada e onze unidades de fermentação contínua.

No referido estudo, foram comparados diversos parâmetros, tais como quantidade de antibióticos usada para tratar o creme de leveduras, quantidade de bactérias contaminantes que se desenvolveram, quantidade de ácido sulfúrico usada para tratar a levedura e taxa de fermentação.

Este estudo mostrou como resultado, que o sistema em batelada alimentada se mostrou melhor que o contínuo em todos os quesitos avaliados, consumindo menos antibióticos e ácido para tratamento bacteriano, tendo menor concentração de contaminantes biológicos e apresentando maior taxa de fermentação.

Assim, pode-se afirmar que mesmo apresentado menores custos de operação e outras vantagens, o sistema de fermentação contínua, não supera, na maioria dos casos, o rendimento do sistema de fermentação em batelada alimentada (CNPEM, 2017).

Por se restringir a laboratórios e pequenas refinarias, o sistema em batelada sem alimentação não foi incluído neste comparativo.

4.5 Bioquímica do Processo Fermentativo

Como os problemas considerados nesta tese são problemas de fermentação modelados matematicamente, compreender a química da fermentação, uma vez que se trata de um processo bioquímico, e como esse processo é traduzido para um modelo matemático é sumamente importante para entender os modelos referentes aos problemas envolvidos no presente trabalho.

Além disso, para se obter um modelo matemático satisfatório que reflita o processo fermentativo, consideradas certas limitações, não basta conhecer o processo em alto nível de abstração e traçar um modelo para ele. Deve-se conhecer o processo a nível molecular, entendendo como os microrganismos convertem as moléculas de açúcares em moléculas de etanol, e a partir daí, modelar o processo.

A formação do álcool acontece durante o processo de fermentação, que consiste em uma série de reações bioquímicas catalizadas por microrganismos, geralmente fungos da espécie *Saccharomyces cerevisiae*.

4.5.1 Conceitos Fundamentais

O mosto é o composto de substrato e nutrientes que contém açúcares fermentescíveis que serão transformados em álcool através do metabolismo da levedura. A água acrescida ao mosto tem o intuito de diluí-lo, reduzindo a concentração de produtos que inibem a levedura, como o próprio etanol, e acrescentando oxigênio ao meio de cultura. Normalmente, o creme de levedura recebe ácido sulfúrico a fim de que seu PH fique entre 2 e 2,5, com intuito de reduzir a contaminação da levedura por bactérias (PATERNINA, 2011).

Com exceção da água e do mosto, todos os componentes envolvidos no processo fermentativo são suspensões hidroalcoólicas que contêm leveduras. O creme de levedura apresenta concentração máxima de biomassa, enquanto o suco fermentado apresenta uma concentração residual de biomassa de 0,5% de seu volume.

Embora vejamos sempre a fermentação como um processo para gerar um produto para os humanos, a fermentação é, na verdade, o processo pelo qual a levedura obtém energia para viver, metabolizando os açúcares presentes no mosto. O etanol gerado no processo é apenas um subproduto dele. Diante disto, se faz necessário compreender e fornecer as melhores condições possíveis para que a levedura realize o processo de fermentação com a maior eficiência possível na produção alcoólica (VILELA, 2015).

4.5.2 Processo Bioquímico

No decorrer do processo fermentativo, os microrganismos produzem uma enzima denominada invertase, que é a responsável por hidrolisar a sacarose ($C_{12}H_{22}O_{11}$) em duas moléculas menores, denominadas monossacarídeos, cujo balanço é dado pela Equação (62), sendo uma a D-glicose ($C_6H_{12}O_6$) e outra a D-frutose ($C_6H_{12}O_6$), que embora tenham a mesma formulação molecular, possuem diferentes estruturas, conforme denota a Figura 37. Esses dois monossacarídeos são os responsáveis por possibilitar a formação do etanol (SOUZA, 2009).

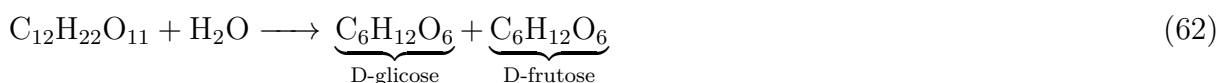
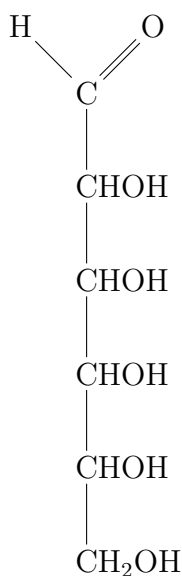
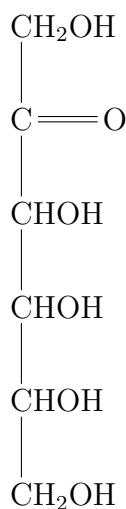


Figura 37 - Fórmulas estruturais da D-glicose e D-frutose.



(a) Estrutura da D-glicose.

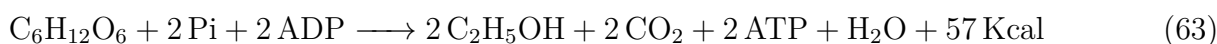


(b) Estrutura da D-frutose.

Fonte: O autor, 2023.

Após a hidrólise da sacarose em D-glicose e D-frutose, estas são convertidas pela zimase em etanol e gás carbônico. A atuação enzimática das leveduras neste processo de conversão difere conforme a presença ou ausência de oxigênio.

Na ausência de oxigênio, as enzimas das leveduras tenderão a gerar álcool (C_2H_5OH) e água (H_2O) a partir do ácido pirúvico (Pi). O referido processo enzimático ocorre no citoplasma celular da levedura, seu balanço é expresso pela equação de Gay-Lussac, dada pela Equação (63).



Por outro lado, na presença de oxigênio, parte do ácido pirúvico (Pi) é desviado para o ciclo de Krebs, onde será oxidado pelas enzimas da levedura, gerando dióxido de carbono (CO₂) e água (H₂O), conforme enuncia a Equação (64), restando uma parcela menor de ácido pirúvico para ser convertida em etanol (C₂H₅OH). Desse modo, quanto menor a concentração de oxigênio, mais favorável à produção de álcool é o meio.



O ciclo de Krebs, ciclo do ácido cítrico ou ciclo do ácido tricarboxílico (TCA), é uma das etapas da respiração celular, sendo um processo aeróbio realizado para obtenção de energia para a célula. Quimicamente, o ciclo de Krebs consiste em uma série de reações químicas que ocorrem na célula, tendo seu metabolismo sido descrito pelo bioquímico alemão Hans Adolf Krebs (NELSON; COX, 2011).

Nota ao Leitor 13 *Neste trabalho, o ciclo de Krebs não será aprofundado, não tendo suas reações descritas e maiores detalhes expostos. Para um maior aprofundamento no assunto, consultar a literatura sugerida ao final do capítulo.*

4.5.3 Fases do Processo Fermentativo

De acordo com Vilela (2015), existem três fases que ocorrem durante o processo de fermentação, que estão descritas abaixo.

Fase Preliminar: Nesta fase, as leveduras estão se adaptando ao meio de cultura, existindo uma produção restrita de dióxido de carbono;

Fase Principal: Nesta fase, a levedura produz, de forma intensa, álcool e gás carbônico;

Fase Complementar: Nesta fase, o substrato presente no meio é exaurido e a produção de álcool decai.

4.5.4 Fases do Crescimento Microbiano no Processo Fermentativo

Da perspectiva do crescimento celular de microrganismos, Doran (1995) elenca quatro fases que ocorrem no processo fermentativo, estando estas descritas abaixo.

Fase Adaptativa: Nesta fase, o metabolismo das leveduras se adapta ao estresse osmótico inicial;

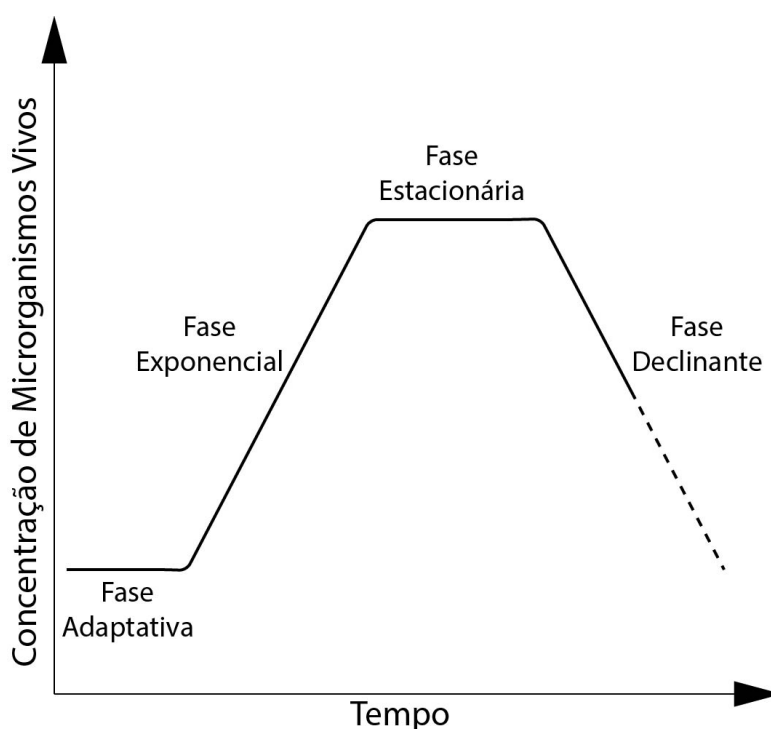
Fase Exponencial: Nesta fase, a levedura possui sua maior taxa de crescimento, elevando a biomassa presente;

Fase Estacionária: Nesta fase, a taxa de crescimento e mortalidade da levedura é praticamente a mesma, fazendo com que o número de microrganismos em desenvolvimento e de microrganismos mortos estejam em equilíbrio;

Fase Declinante: Nesta fase, a taxa de mortalidade da levedura cresce em ritmo mais acelerado que a taxa de crescimento das mesmas, levando à queda no número de microrganismos presentes no meio.

A Figura 38 apresenta um gráfico que esboça a concentração de microrganismos vivos em cada uma das fases de crescimento microbiano no processo fermentativo.

Figura 38 - Concentração de células em cada uma das fases.



Fonte: Adaptado de Doran, 1995.

Nos processos de fermentação em escala industrial, sejam eles contínuos ou em batelada alimentada, deseja-se controlar o processo de modo a ignorar a fase declinante, fazendo com que após a fase estacionária, o suco fermentado siga, de imediato, para a etapa de centrifugação, uma vez que a produtividade na fase declinante não compensa o tempo e recursos empregados. (YABARRENA, 2012).

4.6 Modelagem Matemática do Processo Fermentativo

Otimizar um processo fermentativo através de ajustes e execuções reais deste processo pode ser extremamente demorado ou mesmo inviável no que tange a recursos. Por

esta razão, se faz necessário efetuar simulações computacionais do processo de fermentação, uma vez que estas são mais rápidas e não consomem recursos envolvidos no processo real.

Para que simulações computacionais sejam possíveis, se faz necessário modelar matematicamente o processo fermentativo, a fim de que o modelo obtido seja resolvido e executado diversas vezes.

Desse modo, esta seção apresentará modelos e conceitos inerentes à modelagem matemática do processo fermentativo.

4.6.1 Modelos Cinéticos do Processo Fermentativo

O estudo da cinética de processos microbiológicos tem o intuito de quantificar matematicamente grandezas como, consumo de substrato, crescimento microbiano, formação de produtos, além de outros parâmetros correlatos.

Este estudo também objetiva avaliar a influência de fatores externos, tais como temperatura, PH e inibidores. Quantificar estes parâmetros é vital para projetar e operar uma unidade industrial de produção alcoólica (VIEGAS, 2003).

A cinética do processo de fermentação alcoólica é uma área de considerável interesse de centros de pesquisas e universidades, uma vez que apresenta alto potencial econômico e industrial (LIMA; MARCONDES, 2002).

4.6.1.1 Tipos de Modelos Cinéticos

De acordo com Bailey e Ollis (1986), podemos classificar os modelos cinéticos fermentativos quanto a estrutura e segregação, originando quatro categorias, a saber, modelos não-estruturados e não-segregados, modelos estruturados e não-segregados, modelos não-estruturados e segregados e modelos estruturados e segregados. Cada uma destas categorias está brevemente descrita, abaixo.

Modelos Não-Estruturados e Não-Segregados: Neste tipo de modelo, as células de microrganismos são tratadas como soluto;

Modelos Estruturados e Não-Segregados: Neste tipo de modelo, as células de microrganismos são consideradas seres individuais formados por vários componentes, mas tendo composição semelhante;

Modelos Não-Estruturados e Segregados: Neste tipo de modelo, os microrganismos são tratados como indivíduos distintos formados por um único componente;

Modelos Estruturados e Segregados: Neste tipo de modelo, as células de microrganismos são tratadas como indivíduos formados por distintos componentes com composições distintas.

Os modelos mais usados pela literatura para descrever o comportamento das variáveis envolvidas no processo de fermentação são os Modelos Não-Estruturados e Não-Segregados, devido a sua simplicidade, baixo custo de processamento e capacidade de fornecer boas aproximações do comportamento real (HAN; LEVENSPIEL, 1988).

4.6.1.2 Complexidade e Limitações dos Modelos

A complexidade do modelo cinético depende de diversos fatores, tais como aplicação, precisão desejada e condições físicas. Quanto mais fiel e próximo da realidade, mais complexo o modelo será.

No entanto, não é possível ou viável formular um modelo que contemple todas as características e detalhes que ocorrem na realidade. Dessa forma, um modelo deve ser formulado e entendido como uma aproximação simplificada da realidade, que é satisfatória para aferir os dados que se deseja (STREMEL, 2001).

4.6.1.3 A Equação de Monod

Segundo Han e Levenspiel (1988), a forma mais popular e simples de modelar e descrever o comportamento dos microrganismos no processo fermentativo é através da equação de Monod, que trata o substrato como fator limitante para o desenvolvimento microbiano. A Equação (65) expressa formalmente a Equação de Monod, onde μ é a velocidade da reação, μ_{max} é a velocidade máxima da reação, S é a concentração de substrato final e K_s é a constante de limitação pelo substrato.

$$\mu = \mu_{max} \frac{S}{S + K_s} \quad (65)$$

Cabe ressaltar que a equação de Monod não pode ser aplicada quando houver a presença de produtos metabólicos tóxicos. Uma outra limitação desta é não considerar os efeitos inibidores da própria cinética da fermentação (LUONG, 1985; HAN; LEVENSPIEL, 1988).

4.6.1.4 Modelos Baseados na Cinética de Michaelis-Menten

No processo de fermentação alcoólica, o crescimento de biomassa da levedura reduz conforme a concentração de álcool aumenta no suco, configurando uma relação entre o rendimento da levedura e o efeito inibidor pelo produto. Dessa forma, o efeito inibidor do álcool deve estar presente na descrição matemática do processo (THATIPALAMA; ROHANI; HILL, 1992; AIBA; SHODA; NAGATANI, 1968).

Diversos cientistas analisaram e estudaram a cinética da fermentação de etanol e formularam diversos modelos, considerando a limitação pelo substrato e a possível inibição provocada pelo produto, concentração de microrganismos e pelo próprio substrato (ANDRIETTA, 1994).

Os modelos cinéticos mais populares elaborados foram baseados na chamada cinética de Michaelis-Menten. A seguir, apresentaremos alguns destes modelos não-estruturados e não-segregados propostos para fermentação alcoólica.

Modelo de Ghose e Tyagi: O modelo proposto por Ghose e Tyagi (1979) é um dos mais populares modelos cinéticos para fermentação alcoólica, sendo expresso pela Equação (66), onde μ_i é a velocidade de crescimento microbiano, μ_{max} é a velocidade máxima de crescimento microbiano, S_i é a concentração de substrato, K_i é a constante de inibição pelo substrato, K_s é a constante de limitação pelo substrato, P_i é a concentração inicial de produto e P_{max} é a concentração máxima de produto.

$$\mu_i = \mu_{max} \frac{S_i}{S_i + K_s + \frac{S_i^2}{K_i}} \left(1 - \frac{P_i}{P_{max}}\right) \quad (66)$$

Este modelo tem por características: a presença de substrato limitante, inibição pelo substrato e inibição linear pelo produto.

Modelo de Levenspiel: Levenspiel (1980) propôs o modelo cinético descrito pela Equação (67), onde μ_i é a velocidade de crescimento microbiano, μ_{max} é a velocidade máxima de crescimento microbiano, S_i é a concentração de substrato, K_s é a constante de limitação pelo substrato, P_i é a concentração inicial de produto, P_{max} é a concentração máxima de produto e Y_n é o fator potência de inibição pelo produto.

$$\mu_i = \mu_{max} \left(1 - \frac{P_i}{P_{max}}\right)^{Y_n} \frac{S_i}{K_s + S_i} \quad (67)$$

Este modelo tem por características: a presença de substrato limitante, inibição de potência pelo produto e ausência de inibição pelo substrato.

Modelo de Lee, Pagan e Rogers: O modelo de Lee, Pagan e Rogers (1983) é o dado pela Equação (68), onde μ_i é a velocidade de crescimento microbiano, μ_{max} é a

velocidade máxima de crescimento microbiano, S_i é a concentração de substrato, K_s é a constante de limitação pelo substrato, P_i é a concentração inicial de produto, P_{max} é a concentração máxima de produto, X_i é a concentração inicial de células e X_{max} é a concentração máxima de células, Y_n é o fator potência de inibição pelo produto e Y_m é o fator de inibição por biomassa.

$$\mu_i = \mu_{max} \frac{S_i}{K_s + S_i} \left(1 - \frac{P_i}{P_{max}}\right)^{Y_n} \left(1 - \frac{X_i}{X_{max}}\right)^{Y_m} \quad (68)$$

Este modelo tem por características: a presença de substrato limitante, inibição de potência pelo produto, inibição por altas concentrações de biomassa e ausência de inibição pelo substrato.

Modelo de Sevely: O modelo proposto por Sevely *et al.* (1980) está expresso na Equação (69), onde μ_i é a velocidade de crescimento microbiano, μ_{max} é a velocidade máxima de crescimento microbiano, S_i é a concentração de substrato, K_p é a constante de inibição pelo produto, K_s é a constante de limitação pelo substrato, P_i é a concentração inicial de produto e P_{max} é a concentração máxima de produto.

$$\mu_i = \mu_{max} \frac{S_i}{K_s + S_i} \left(1 - \frac{P_i}{P_{max}}\right) \frac{K_p}{P_i + K_p} \quad (69)$$

Este modelo tem por características: a presença de substrato limitante, inibição parabólica pelo produto e ausência de inibição pelo substrato.

Modelo de Jin: Jin *et al.* (1981) propuseram o modelo descrito pela Equação (70), onde μ_i é a velocidade de crescimento microbiano, μ_{max} é a velocidade máxima de crescimento microbiano, S_i é a concentração de substrato, K_s é a constante de limitação pelo substrato, K_1 é o fator exponencial de inibição pelo produto, K_2 é o fator exponencial de inibição pelo substrato e P_i é a concentração inicial de produto.

$$\mu_i = \mu_{max} \frac{S_i}{K_s + S_i} e^{(-K_1 P_i) - (K_2 S_i)} \quad (70)$$

Este modelo tem por características: a presença de substrato limitante, inibição exponencial pelo substrato e inibição exponencial pelo produto.

Modelo de Tosetto: O modelo proposto por Tosetto (2002) está descrito na Equação (71), onde μ_i é a velocidade de crescimento microbiano, μ_{max} é a velocidade máxima da crescimento microbiano, S_i é a concentração de substrato, K_i é a constante de inibição pelo substrato, K_s é a constante de limitação pelo substrato, P_i é a concentração inicial de produto, P_{max} é a concentração máxima de produto, Y_n é

o fator potência de inibição pelo produto.

$$\mu_i = \mu_{max} \frac{S_i}{S_i + K_s + \frac{S_i^2}{K_i}} \left(1 - \frac{P_i}{P_{max}}\right)^{Y_n} \quad (71)$$

Este modelo tem por características: a presença de substrato limitante, inibição pelo substrato e inibição de potência pelo produto.

4.6.1.5 Importância da Modelagem e Simulação

Todo projeto de planta de fermentação precisa ter uma modelagem detalhada do processo, feita através de modelos cinéticos confiáveis, de modo a estabelecer as melhores e mais eficientes condições de operação.

Manter estas condições requer a adoção de estratégias adequadas de controle e operação. O discernimento sobre quais estratégias escolher se dá por meio de estudo prévio feito sobre a modelagem e simulações computacionais de funcionamento da planta (ANDRIETTA, 1994).

4.6.2 Modelo Elementar do Processo de Fermentação

Aguirre (2004) define a modelagem matemática como o área voltada à avaliação, desenvolvimento e conversão de sistemas do mundo real em modelos matemáticos e vice-versa.

A modelagem de processos fermentativos pode ser estabelecida como a representação matemática dos balanços de massa de cada componente presente no reator, vinculados às reações químicas que acontecem no processo e associados ao tempo que estas reações levam de seu início até a sua conclusão (BONOMI; SCHMIDELL, 2001).

O modelo elementar que reproduz um reator de fermentação é dado pelo sistema de equações ordinárias mostrado na Equação (72), onde t é o tempo, V é o volume ocupado no reator, S é a concentração de substrato, P é a concentração de produto, X é a concentração de células de microrganismos, F é a vazão de saída, F_0 é a vazão de entrada, P_0 é a concentração inicial de produto, S_0 é a concentração inicial de substrato, X_0 é a concentração inicial de células de microrganismos, $Y_{p/s}$ é o coeficiente de rendimento

do produto, $Y_{x/s}$ é o coeficiente de rendimento celular e μ_i é a velocidade da reação.

$$\begin{cases} \frac{dV}{dt} = F_0 - F \\ \frac{dS}{dt} = (S_0 \cdot F_0) - (S \cdot F) + \left(\frac{1}{Y_{x/s}} \cdot X \cdot \mu_i \right) \\ \frac{dP}{dt} = (P_0 \cdot F_0) - (P \cdot F) + \left(\frac{Y_{p/s}}{Y_{x/s}} \cdot X \cdot \mu_i \right) \\ \frac{dX}{dt} = (X_0 \cdot F_0) - (X \cdot F) + (X \cdot \mu_i) \end{cases} \quad (72)$$

Embora este modelo reflita o processo fermentativo dentro de um reator, trata-se de um modelo básico, existindo diversas variantes que refletem especificidades próprias dos problema que estas modelam, uma vez que podem envolver outros fatores ou mesmo tratarem de forma mais específica, os componentes presentes no processo.

Construído o modelo matemático, este deve ser validado através de uma verificação que constata se o comportamento dado pelo modelo equivale ao comportamento do sistema real, apurando também quais são as limitações inerentes ao modelo. A validação requer que o modelo seja posto em simulação, isto é, que o sistema de equações que o compõe seja resolvido, quantas vezes forem necessárias para que se ateste uma fidelidade satisfatória ao sistema real (AGUIRRE, 2004).

4.7 Referências com Informações Complementares

Neste capítulo, foi apresentado o processo de fermentação alcoólica, suas etapas, tipos de processos fermentativos, modelos de plantas, aspectos e transformações químicas da fermentação alcoólica e modelos que reproduzem matematicamente o processo fermentativo, uma vez que estes são conceitos importantes para a compreensão dos problemas considerados nesta tese.

Porém, existem detalhes, especificidades e conceitos acerca da fermentação etanólica, tais como propriedades físicas e químicas do etanol, tratamento de mosto, tratamento da levedura, caracterização da matéria prima, determinação de brix, acidez sulfúrica, sacarose, glicose, frutose, fatores que afetam o processo químico de fermentação, desempenho de diferentes linhagens de leveduras, seleção de cepas, modelos cinéticos de Moser, Hinshelwood, Atala *et al.* e Sansonetti *et al.* não foram percorridos neste capítulo, uma vez que o aprofundamento nestes assuntos não é requisito para o entendimento deste trabalho.

Se existir interesse em obter maior aprofundamento no assunto, conhecendo estes e outros temas acerca da fermentação alcooleira, indicam-se as seguintes obras da literatura:

- Schmidell e Facciotti (2001);

- Aiba, Humphrey e Millis (1973);
- Borzani (1960);
- Eduardo (1991);
- Bailey e Ollis (1986);
- Carvalho e Sato (2001);
- Maiorella, Blanch e Wilke (1981);
- Vilela (2015);
- Bonomi e Schmidell (2001);
- Andrietta (1994);
- Thatipalama, Rohani e Hill (1992);
- Doran (1995);
- Lima, Basso e Amorim (2001);
- Luong (1985);
- Han e Levenspiel (1988);
- Aiba, Shoda e Nagatani (1968);
- Nelson e Cox (2011).

5 FERMENTAÇÃO DE LISINA

Por vezes, sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se lhe faltasse uma gota.

Madre Teresa de Calcuta

Alguns dos processos tratados nesta tese são processos de fermentação de lisina. Assim, para que se possa compreender tais problemas, se faz necessário conhecer o processo de fermentação de lisina, sua importância e características, além das etapas do processo fermentativo, da bioquímica e da modelagem matemática do processo, temas que são abordados neste capítulo.

As proteínas são algumas das macromoléculas biológicas mais abundantes no seres vivos, estando presentes em todas as células destes. Elas apresentam uma alta gama de funções biológicas, sendo os instrumentos moleculares por meio dos quais a informação genética é expressa.

As subunidades básicas das proteínas são os chamados aminoácidos. Existem vinte aminoácidos primários ou comuns, que constituem uma espécie de alfabeto usado por virtualmente todas as proteínas encontradas na natureza.

Os aminoácidos são moléculas estruturais que possuem imprescindível papel fisiológico e metabólico em todos os seres vivos. Entretanto, seres vivos como mamíferos e aves não apresentam a capacidade de sintetizar alguns destes aminoácidos, sendo mais específico, os que compõem o grupo dos aminoácidos essenciais. Dessa forma, os aminoácidos essenciais precisam ser obtidos já formados através da alimentação.

5.1 Grupos R e os L-aminoácidos

Todos os aminoácidos primários são α -aminoácidos porque possuem um grupo carboxila e um grupo amino ligados ao mesmo átomo de carbono (carbono α). Estes aminoácidos diferem entre si em suas cadeias laterais, chamadas de grupos R, que variam em estrutura, tamanho e carga elétrica, fatores que influenciam sua solubilidade em água.

Segundo as propriedades de sua cadeia lateral, os aminoácidos podem ser agrupados em:

- Grupos R carregados positivamente;
- Grupos R carregados negativamente;
- Grupos R alifáticos não polares;

- Grupos R não-carregados polares;
- Grupos R aromáticos;

Os carbonos α de qualquer aminoácido, com exceção da glicina, são compostos quirais, implicando na existência de ao menos dos estereoisômeros em cada aminoácido. Entretanto, quando processados por organismos, passam a apresentar apenas um isômero, uma vez que as enzimas envolvidas na síntese dos aminoácidos são estereoespecíficas, resultando no que chamamos de L-aminoácidos (NELSON; COX, 2011).

Os aminoácidos que compõem as proteínas dos seres vivos, são em sua totalidade L-aminoácidos. Por esta razão, a grande maioria dos aminoácidos fabricados industrialmente são L-aminoácidos.

5.2 A Importância da Indústria de L-lisina

A L-lisina é um aminoácido essencial, que é fundamental não apenas para os humanos, mas também para a pecuária, sendo essencial para animais que apresentam grande importância econômica, como aves, suínos e bovinos.

Para suprir a necessidade dos aminoácidos essenciais, a indústria agropecuária emprega suplementos de aminoácidos em rações para acelerar o crescimento e ganho de peso dos animais, uma vez que a maioria dos grãos como a soja e milho são deficientes em vários aminoácidos.

A adição da L-lisina na ração de animais começou a acontecer na segunda metade do século XX, principalmente por possibilitar a fabricação de rações com componentes mais baratos, mantendo-as com taxas satisfatórias de proteínas, que conferem bons índices de crescimento aos animais consumidores (TORIDE, 2016).

Embora a maioria absoluta de toda a L-lisina produzida seja destinada à suplementação de rações para alimentação animal, uma fatia menor da produção é usada na geração de suplementos alimentares humanos e medicamentos farmacêuticos.

Normalmente, o ser humano obtém a porção necessária de L-lisina por meio da alimentação, uma vez que esta é encontrada em alimentos ricos em proteína, tais como carnes, frutos do mar, soja, leite e seus derivados, sendo também encontrada, em menor concentração, em legumes e verduras (HAND, 2015).

Entretanto, existem situações em que esse aminoácido pode se apresentar deficiente no organismo, ocasionando sintomas como náusea, tontura, fadiga, perda de apetite, problemas reprodutivos e de crescimento. Neste caso, faz-se necessária a suplementação deste aminoácido na alimentação do indivíduo em questão.

Naturalmente, esse tipo de suplementação deve ser indicada e acompanhada por profissionais competentes, uma vez que o excesso de L-lisina no corpo pode provocar

sintomas como diarreia, cólica, elevação do colesterol, danos hepáticos e renais.

Existem também fármacos que são sintetizados a partir de princípios ativos com base na lisina, principalmente o cloridrato de lisina, usado na prevenção da herpes, doença causada pelo vírus *Herpes simplex*, e o clonixinato de lisina, que apresenta efeito analgésico, sendo usado comumente a abrandar sintomas como cólica, cefaleia, dor muscular, articular, pós-traumática e pós-cirúrgica.

Atualmente, os maiores produtores mundiais de L-lisina são a japonesa Ajinomoto, responsável pela produção de cerca de 24% da produção mundial, e a sul coreana CJ CheilJedang, que gera cerca de 23% da produção mundial. Ambas possuem plantas industriais de fermentação de L-lisina em solo brasileiro (LETT, 2014).

Limitando-se ao Brasil, o maior produtor de L-lisina em solo nacional é a alemã Evonik, com uma planta de produção industrial localizada na cidade de Castro, que pertencente ao estado do Paraná (BINDER, 2020).

5.3 Processo Produtivo da L-lisina

Usualmente, a produção de L-lisina se dá através de processos fermentativos, conduzidos por bactérias do gênero *Corynebacterium* ou por algumas cepas específicas de *Escherichia coli*, em que comumente se empregam fermentações líquidas cujo produto final é um sal de lisina (Lisina-HCl) que é quimicamente estável e pode ser cristalizado (JUNIOR; LETTI; SOCCOL, 2016).

Este sal de lisina, denominado Monocloridrato de L-lisina (Lisina-HCl), é a forma predominantemente em que a L-lisina é comercializada, sendo empregada em larga escala como aditivo alimentar na indústria de ração animal, uma vez que a maior parte dos cereais usados na rações são deficientes em L-lisina (ANASTASSIADIS, 2007).

Os principais substratos empregados na indústria de produção de L-lisina são o melão de cana-de-açúcar, melão de beterraba, sacarose e dextrose, que comumente é obtida pela hidrólise do amido.

As principais fontes de carbono empregadas na produção de L-lisina são a frutose, glicose e sacarose. Entretanto o uso da glicose ou da sacarose proporcionam maior geração de L-lisina que o uso da frutose como fonte de carbono. Com relação aos níveis de biomassa, a sacarose e a frutose apresentam níveis similares, enquanto a glicose apresenta níveis superiores (KIEFER; HEINZLE; WITTMANN, 2002).

Existem alguns fatores que encarecem consideravelmente a produção de L-lisina, um deles é a matéria prima para a fermentação, outro são as próprias operações de processamento. Dessa forma, otimizar o processo fermentativo colaboraria com a viabilização econômica da produção de L-lisina, permitindo que novos entrantes no mercado possam competir com as multinacionais que dominam o mercado atualmente.

5.4 Etapas do Processo Fermentativo

O processo de produção de L-lisina pode ser dividido em nove etapas, que abarcam desde o preparo do meio de cultura até a embalagem e armazenamento para posterior distribuição. Cada uma destas etapas está listada abaixo (CHEMICAL ENGINEERING, 2016).

- Preparação;
- Fermentação;
- Ultrafiltração;
- Cromatografia;
- Concentração;
- Acidificação;
- Cristalização;
- Secagem;
- Embalagem e Armazenagem.

Nas subseções a seguir, cada uma das etapas citadas acima será brevemente apresentada e descrita.

5.4.1 Preparação

Nesta primeira etapa, o mosto que será inserido no reator de fermentação em batelada é preparado e formado através da mistura de água, glicose e nutrientes.

5.4.2 Fermentação

Esta é a etapa onde de fato acontece a formação da L-lisina. Por este motivo, dentre todas as etapas listadas, esta é a que possui particular interesse nesta tese, sobre a qual discorreremos amplamente, uma vez que o objetivo é otimizar a produção de L-lisina.

A etapa de fermentação de lisina é realizada no modo descontínuo alimentado e sob condições aeróbicas, se dividindo em duas fases, a saber, a fase de batelada e a fase de batelada alimentada.

5.4.2.1 Fermentação em Batelada

Nesta fase, o reator é previamente enchido parcialmente com mosto, sendo em seguida inoculado com os microrganismos. Após este procedimento, o reator permanece em batelada até o início da fase de alimentação.

5.4.2.2 Fermentação em Batelada Alimentada

Nesta fase, o mosto contendo glicose e nutrientes é fornecido de forma contínua ao reator até que o tempo de alimentação do reator termine. Após a alimentação, o reator entra novamente em batelada até o término do processo de fermentação.

Ao término da etapa de fermentação, o caldo delevedurado segue para um taque a fim de fornecer um fluxo contínuo de suco para as etapas seguintes do processo.

5.4.3 Ultrafiltração

Após a fermentação, o vinho segue para um sistema de ultrafiltração, que objetiva eliminar detritos celulares e outros resíduos sólidos que estejam suspensos no vinho delevedurado a fim de que o suco esteja livre destes nas etapas seguintes.

5.4.4 Cromatografia

Depois de passar pela ultrafiltração, o licor resultante segue para colunas de troca iônica, onde a L-lisina é gradativamente adsorvida a fim de ser purificada pelo processo de cromatografia em coluna.

Conforme a L-lisina adsorvida elui das colunas, esta segue para uma lavagem com amônia para que as resinas de troca iônica sejam removidas.

5.4.5 Concentração

Nesta etapa, a L-lisina advinda da etapa anterior é misturada ao licor mãe reciclado na filtragem que ocorre ao fim do processo, que segue para um vaporizador, onde é concentrada por evaporação.

5.4.6 Acidificação

Na presente etapa, a solução concentrada oriunda da etapa anterior é acidificada com ácido clorídrico, convertendo a L-lisina presente nesta em L-lisina hidrocloreídrica (L-lisina HCl).

5.4.7 Cristalização

Nesta fase, a solução acidificada é enviada para o cristalizador, gerando o sal de lisina, que segue para um filtro, que por sua vez, o separa da solução. O licor residual, chamado de licor mãe, é reciclado, seguindo para a etapa de concentração.

5.4.8 Secagem

Na presente etapa, a massa úmida de sal de lisina segue para o secador, resultando na L-lisina-HCl seca, que apresenta 98,5% de lisina em sua composição.

5.4.9 Embalagem e Armazenagem

Na etapa final, L-lisina-HCl seca segue para ser embalada, ensacada e estocada para posteriormente ser transportada até os consumidores e distribuidores da L-lisina.

5.5 **Processo Fermentativo**

O processo de fermentação empregado para produção de L-lisina é a fermentação em batelada alimentada. Este processo já foi apresentado e explicado na Seção 4.4.2. A única diferença é que aqui ele está voltado para a geração de L-lisina e não de etanol.

5.5.1 Processo Industrial de Produção de Lisina

O processo comumente empregado nas usinas que produzem L-lisina consiste na colocação do mosto devidamente preparado no reator, enchendo-o parcialmente, introduzindo, em seguida, as bactérias. O reator fica neste estado até o início da alimentação.

Depois da batelada inicial, um alimentador injeta de forma contínua o substrato

no reator, onde o processo fermentativo em batelada já está ocorrendo, até que o esteja completamente cheio.

Após o reator estar cheio, espera-se o tempo restante para o término da fermentação e em seguida, o suco fermentado segue para um tanque, que a partir deste segue para uma unidade de ultrafiltração, que elimina resíduos sólidos.

Ao término da ultrafiltração, o licor resultante segue para colunas de troca iônica, onde será purificado de substâncias indesejáveis, passando em seguida por um banho de solução amôniaca para eliminar as resinas da troca iônica.

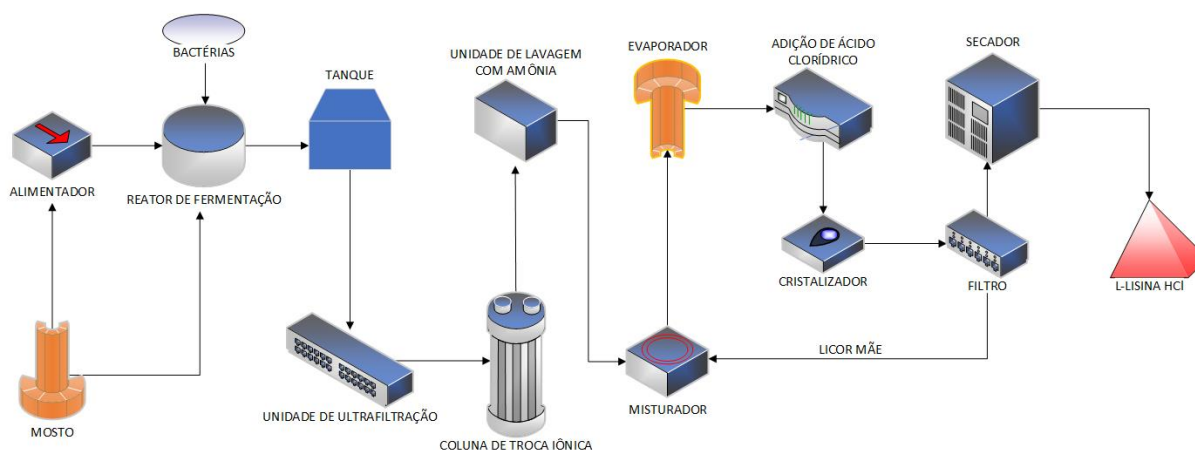
A seguir, a L-lisina é misturada ao licor residual que advêm da filtragem que ocorre ao término do processo, seguindo para uma evaporador que reduz a mistura, deixando-a concentrada, passando, em seguida, por uma adição de ácido clorídrico, resultando na L-lisina HCl.

A solução então segue para um cristalizador que gera o sal de lisina, indo em seguida para um filtro que o separa do licor. O licor segue para ser misturado à lisina que está saindo do banho de amônia.

A massa úmida de sal de lisina, por sua vez, segue para um secador que o coloca em sua forma seca, que é como a lisina é amplamente comercializada.

A Figura 39 ilustra o processo descrito acima.

Figura 39 - Processo industrial de fermentação de L-lisina.



Fonte: O autor, 2023.

5.6 Biossíntese da L-lisina

A maior parte das cepas selvagens de fungos e bactérias não conseguem sintetizar altas quantidades de L-lisina, em virtude de mecanismos específicos de regulação metabólica de sua fisiologia. Porém, a alteração destes mecanismos pode eliminar essa limitação, propiciando produções consideráveis de L-lisina.

Existem duas vias biossintéticas conhecidas para a geração de L-lisina. A primeira é a via do ácido α -aminoadípico, que ocorre em alguns actinomicetos algas, em que o esqueleto de carbono para estruturar a L-lisina advém do acetato de α -cetogluturato.

A segunda via é a do diaminopimelato, que ocorre em certos ficomicetos, plantas superiores, algas azuis e protozoários, onde a L-lisina é sintetizada a partir de piruvato e aspartato (SHAH; HAMEED; KHAN, 2002).

Embora existam diversas variantes e rotas destas vias, nos ateremos aqui à empregada pela *Corynebacterium glutamicum*, que é o microrganismo predominantemente usado na produção de aminoácidos, tal como a lisina.

A *Corynebacterium glutamicum* produz e excreta a lisina a partir fundamentalmente do piruvato e do oxaloacetato, possuindo vias separadas para a síntese de lisina. A presença destas vias alternativas fornecem à bactéria maior tolerância a mudanças de condições do meio (AN; SONG; SINSKEY, 1999).

5.6.1 Processo Bioquímico

No início do processo de síntese da L-lisina, os microrganismos convertem a glicose, na forma ($C_6H_{11}O_9P$), em gliceraldeído ($C_3H_7O_6P$), conforme expressa a Equação (73).



Em seguida, gera-se o fosfoenolpiruvato ($C_3H_5O_6P$) a partir do gliceraldeído ($C_3H_7O_6P$), como denota a Equação (74).



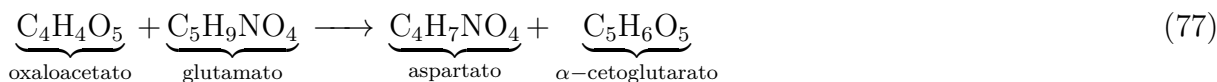
Conforme enunciado na Equação (75), a partir do fosfoenolpiruvato ($C_3H_5O_6P$), gera-se o piruvato ($C_3H_4O_3$), que é um dos elementos fundamentais para a produção da lisina.



A partir do piruvato ($C_3H_4O_3$), segundo a Equação (76), produz-se dióxido de carbono (CO_2) e a acetilcoenzima A ($C_{23}H_{38}N_7O_{17}P_3S$), que segue para o ciclo de Krebs.



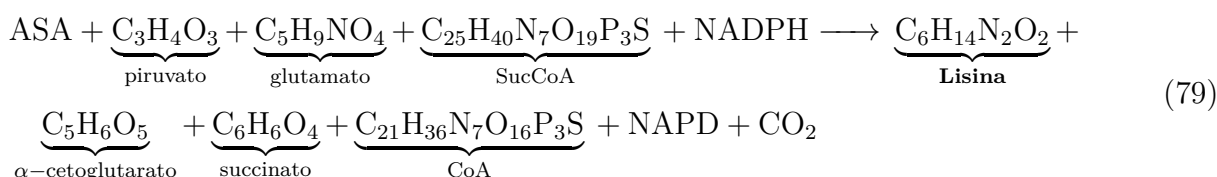
No decorrer do ciclo de Krebs, são gerados o glutamato ($C_5H_9NO_4$) e o oxaloacetato ($C_4H_4O_5$), a partir dos quais, conforme denota a Equação (77), gera-se o aspartato ($C_4H_7NO_4$), que é um dos elementos primordiais para a geração da lisina.



A partir do aspartato ($C_4H_7NO_4$), de acordo com a Equação (78), produz-se o aspartato semialdeído (ASA), que estará diretamente envolvido na geração de lisina.



Obtidos todos os componentes para a geração da lisina, sendo estes o aspartato semialdeído (ASA), o piruvato ($C_3H_4O_3$), o glutamato ($C_5H_9NO_4$) e a Succinilcoenzima A ($C_{25}H_{40}N_7O_{19}P_3S$), que também é gerada durante o ciclo de Krebs, os microrganismos realizam a reação cujo o balanço é dado pela Equação (79), que, a partir dos elementos citados, gera **lisina** ($C_6H_{14}N_2O_2$), α -cetoglutarato ($C_5H_6O_5$), succinato ($C_6H_6O_4$), coenzima A ($C_{21}H_{36}N_7O_{16}P_3S$) e dióxido de carbono (CO_2).



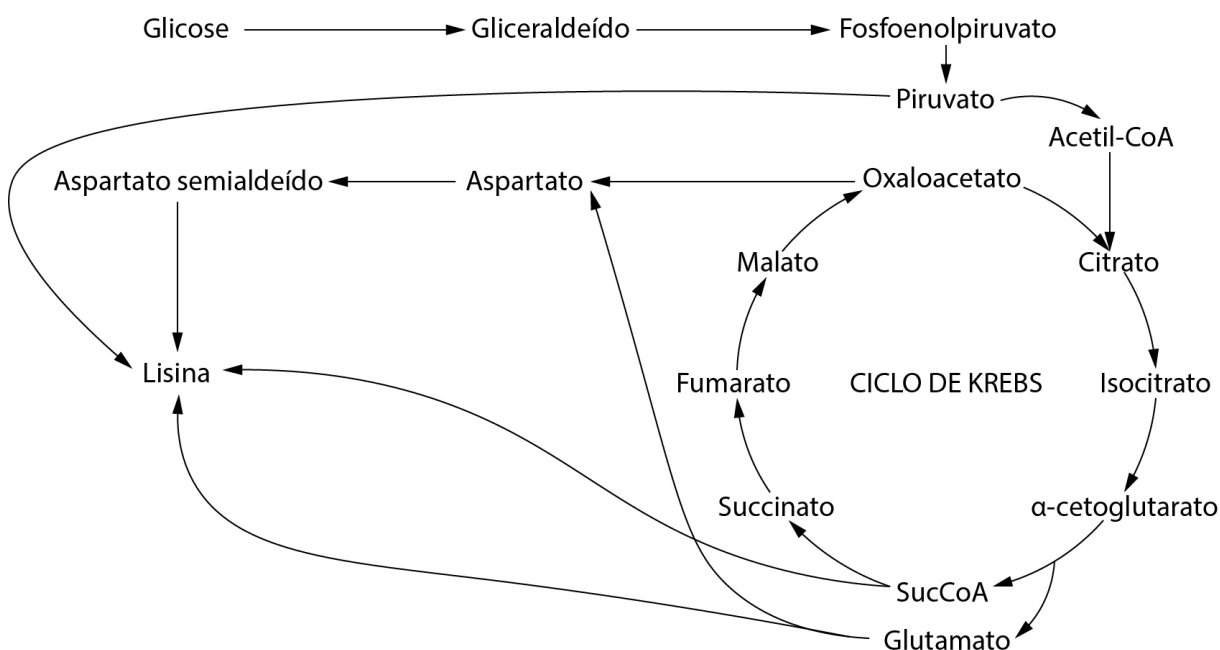
A Figura 40 ilustra uma simplificação do processo metabólico e da biossíntese de lisina referentes à *Corynebacterium glutamicum*.

5.6.2 Rendimento Teórico Máximo

Uma característica importante acerca do processo de produção de L-lisina é o rendimento teórico máximo ou capacidade máxima de síntese de lisina em uma célula de *Corynebacterium glutamicum*. O rendimento teórico máximo permite obter uma estimativa do potencial de otimização remanescente do processo industrial de produção de lisina.

De acordo com Hollander (1994), o rendimento molar máximo na conversão de glicose em lisina é de 75%, podendo chegar em até 82%, dependendo das condições e do fluxo metabólico da *Corynebacterium glutamicum* (SHAH; HAMEED; KHAN, 2002).

Figura 40 - Esquema simplificado do metabolismo central e vias de síntese de lisina na *Corynebacterium glutamicum*.



Fonte: Adaptado de Chen *et al.*, 2014.

5.7 Modelagem Matemática do Processo Fermentativo

Para otimizar um processo fermentativo através de simulações computacionais se faz necessário modelar matematicamente o processo fermentativo a fim de que o modelo obtido possa ser resolvido e executado diversas vezes. Desse modo, esta seção apresentará a modelagem matemática do processo fermentativo de L-lisina.

Diferentemente da fermentação alcoólica, na fermentação de L-lisina, o crescimento da biomassa de bactérias não é reduzido conforme a concentração de lisina aumenta no suco, uma vez que a lisina não é tóxica aos microrganismos. Dessa forma, não existe inibição da reação pelo produto, fator que será naturalmente refletido na descrição matemática do processo.

O modelo cinético mais popular elaborado para descrever a fermentação de L-lisina é o **Modelo de Ohno e Nakanishi**, que foi proposto por Ohno e Nakanishi (1976), sendo amplamente usado na atualidade.

Compõem o modelo em questão, as Equações (80), (81) e (82), onde C_S é a concentração de substrato, V é o volume ocupado no reator, μ é a taxa de crescimento celular, σ é a taxa de consumo de substrato e π é a taxa de formação de produto.

$$\mu = 0,125C_S \quad (80)$$

$$\sigma = \frac{\mu}{0,135} \quad (81)$$

$$\pi = -384\mu^2 + 134\mu \quad (82)$$

A modelagem do processo fermentativo é realizada, neste caso, pela representação matemática dos balanços de massa de cada componente presente no reator de fermentação, vinculados às reações químicas que acontecem no processo e associados ao tempo que estas levam de seu início até o seu término (BONOMI; SCHMIDELL, 2001).

O modelo matemático que reproduz o reator de fermentação de L-lisina, no presente modelo cinético, fornece os balanços de massa dos componentes presentes nele conforme o instante de tempo, sendo expresso pelo sistema de equações diferenciais ordinárias dado na Equação (83), onde X é a quantidade de células de microrganismos, S é a quantidade de substrato, P é a quantidade de produto, V é o volume ocupado no reator, F é a vazão de entrada e C_S é a concentração de substrato na vazão de entrada.

$$\begin{cases} \frac{dX}{dt} = \mu X \\ \frac{dS}{dt} = FC_S - \sigma X \\ \frac{dP}{dt} = \pi X \\ \frac{dV}{dt} = F \end{cases} \quad (83)$$

5.8 Referências com Informações Complementares

No curso deste capítulo, foi apresentado o processo de fermentação de lisina, suas etapas, processo fermentativo, aspectos e transformações químicas da fermentação de lisina e o modelo matemático que reproduzem o processo fermentativo, uma vez que estes são conceitos importantes para a compreensão dos problemas considerados nesta tese.

Entretanto, existem conceitos e temas inerentes à fermentação de lisina, tais como rotas de biossíntese de L-lisina, melhoramento genético, quantificação de componentes e conservação e seleção de cepas produtoras de L-lisina não foram percorridos neste capítulo, uma vez que o aprofundamento nestes assuntos não é requisito para a plena compreensão desta tese.

Se existir interesse em conhecer melhor o assunto, se iterando destes e de outros temas acerca da fermentação de lisina, ficam recomendados os seguintes trabalhos da literatura:

- Liu *et al.* (2022);

- Nelson e Cox (2011);
- Chemical Engineering (2016);
- Shah, Hameed e Khan (2002);
- Lett (2014);
- Bott (2007).

6 ALGORITMO DESENVOLVIDO

Uma mente que se abre a uma nova ideia,
jamais voltará ao seu tamanho original

Albert Einstein

No presente capítulo, o algoritmo intitulado **Computational Ordinary Differential Equations Modeled System Optimization Algorithm (CODEMSOA)**, proposto neste trabalho, será apresentado e especificado, tendo sua lógica descrita e explanada.

Também, neste capítulo, foi enunciado o modelo computacional proposto e empregado na implementação do CODEMSOA, acompanhado das informações e detalhes relacionados a esta implementação.

O intuito deste desenvolvimento foi construir e implementar um algoritmo capaz de otimizar processos modelados por equações diferenciais ordinárias em uma única plataforma de desenvolvimento, sem envolver ferramentas externas e processos de integração entre ferramentas.

6.1 Algoritmo

O algoritmo proposto nesta tese promove a otimização de processos modelados por equações diferenciais ordinárias, permitindo o emprego de métodos numéricos para equações diferenciais ordinárias, métodos de otimização e de interpolação no processo.

O CODEMSOA é composto de três partes elementares, a saber, o solucionador de equações diferenciais ordinárias, o interpolador de soluções e o algoritmo de otimização. Cada uma destas partes será especificada a seguir.

Antes, porém, será introduzido o algoritmo de valoração funcional, proposto por este trabalho, que será empregado inúmeras vezes no CODEMSOA.

6.1.1 Algoritmo de Valoração Funcional

Como nos problemas a serem resolvidos, o problema de valor inicial possui termos que serão variáveis de decisão, este precisa que o valor destes termos seja atualizado a cada iteração do método de otimização, a fim de que, ao ser resolvido, naquela iteração, reflita o comportamento correto para o presente conjunto de parâmetros.

Do mesmo modo, é natural que as funções objetivo do problema de otimização possuam termos que consistem na solução do problema de valor inicial ou em parte dela.

Assim, estes termos também precisam estar atualizados, a fim de que ao serem invocadas pelo método de otimização, as funções objetivo retornem o valor do objetivo condizente com o atual conjunto de parâmetros.

A fim de permitir as atualizações mencionadas, que precisam ocorrer de forma dinâmica durante a execução do algoritmo, propôs-se nesta tese, um algoritmo intitulado Algoritmo de Valoração Funcional.

Este algoritmo permite atribuir valor a uma determinada parte dos argumentos de uma dada função, deixando-a definida apenas em termos da parte remanescente de seus argumentos.

O algoritmo em tela tem por entrada uma função q e um vetor φ , que consiste na parte dos argumentos de q a ser valorada com seus respectivos valores atribuídos; e como saída uma função h , que por sua vez, recebe como argumento o vetor \wp , que consiste na parte dos argumentos de q que não será valorada, e retorna o resultado da chamada $g(\varphi, \wp)$.

O Algoritmo 14 expressa formalmente o algoritmo em questão.

Algoritmo 14: Algoritmo de Valoração Funcional.

Entrada: $q(\cdot), \varphi$
Saída: $h(\cdot)$
1 *VALORAR*($q(\cdot), \varphi$)
2 **início**
3 $h \leftarrow \begin{cases} FUNCAO(\wp) \\ \text{início} \\ \text{retorna } q(\varphi, \wp); \\ \text{fim} \end{cases} ;$
4 **retorna** $h(\cdot)$;
5 **fim**

6.1.2 Solucionador de Equações Diferenciais Ordinárias

Como o próprio nome sugere, o solucionador de equações diferenciais ordinárias é o algoritmo incumbido de resolver o problema de valor inicial oriundo do modelo descrito por equações diferenciais ordinárias.

Este processo consiste inicialmente na composição de um problema de valor inicial ρ envolvendo todas as equações diferenciais ordinárias ε associadas às suas respectivas condições iniciais ς .

Durante a composição do problema de valor inicial, as funções que modelam as equações diferenciais ordinárias, que possuem o parâmetro \bar{x} , que representa o vetor de variáveis do problema de otimização, têm este parâmetro valorado por x , através do

operador de valoração funcional.

Finalmente, emprega-se o método numérico para resolução de equações diferenciais ordinárias σ sobre o problema de valor inicial ρ no intervalo Ξ , que fornece como solução a definição das funções incógnitas μ em cada um dos pontos \varkappa .

O Algoritmo 15 expressa o solucionador de equações diferenciais ordinárias, onde $z \in \mathbb{N}$ é o número de funções incógnitas, $w \in \mathbb{N}$ é o número de pontos nos quais as funções incógnitas são avaliadas, $\varkappa = (\varkappa_1, \varkappa_2, \varkappa_3, \dots, \varkappa_w)$ são os pontos nos quais as funções incógnitas são avaliadas, $\mu = ((\mu_1^1, \mu_1^2, \dots, \mu_1^w), (\mu_2^1, \mu_2^2, \dots, \mu_2^w), \dots, (\mu_z^1, \mu_z^2, \dots, \mu_z^w))$ são as estimativas das funções incógnitas nos pontos \varkappa , $\varepsilon(\varkappa, \mu) = \langle \varepsilon_1(\varkappa, \mu), \varepsilon_2(\varkappa, \mu), \varepsilon_3(\varkappa, \mu), \dots, \varepsilon_z(\varkappa, \mu) \rangle$ são as funções que modelam as equações diferenciais ordinárias, $\varsigma = (\varsigma_1, \varsigma_2, \varsigma_3, \dots, \varsigma_z)$ são as condições iniciais do problema, ρ é o problema de valor inicial a ser resolvido, $\Xi = [\check{\Xi}_0, \check{\Xi}_f]$ é o intervalo de solução do problema e $\sigma(\rho, \Xi)$ é o método numérico para equações diferenciais ordinárias.

Algoritmo 15: Solucionador de Equações Diferenciais Ordinárias.

Entrada: $\sigma(\cdot), \varepsilon(\cdot), \varsigma, \Xi, x$
Saída: (\varkappa, μ)

```

1 EDO_SOLVER( $\sigma(\cdot), \varepsilon(\cdot), \varsigma, \Xi, x$ )
2 início
3    $\rho \leftarrow \emptyset;$ 
4   para  $i = 1$  até  $z$  faça
5      $\varepsilon_i^* \leftarrow VALORAR(\varepsilon_i(\cdot), x);$ 
6      $\rho \leftarrow \rho \cup \langle \varepsilon_i^*(\cdot), \varsigma_i \rangle;$ 
7   fim
8    $(\varkappa, \mu) \leftarrow \sigma(\rho, \Xi);$  %resultado do método numérico para edo
9   retorna  $(\varkappa, \mu);$ 
10 fim
```

6.1.3 Interpolador de Soluções

O interpolador de soluções, conforme sugerido pelo nome, é o algoritmo responsável por interpolar os conjuntos de pontos resultantes do solucionador de equações diferenciais ordinárias.

O presente processo consiste em um *loop*, onde, a cada iteração, é aplicado o método de interpolação γ a (\varkappa, μ_i) , que resulta em uma função contínua ϕ_i que estima o comportamento da função incógnita no intervalo delimitado pelos pontos \varkappa considerados.

O laço de repetição é encerrado quando todas as funções que compõem o problema de equações diferenciais ordinárias estão devidamente interpoladas, provendo um conjunto de funções contínuas ϕ em um determinado intervalo.

O Algoritmo 16 formaliza o interpolador de soluções, onde $z \in \mathbb{N}$ é o número

de funções incógnitas, $w \in \mathbb{N}$ é o número de pontos nos quais as funções incógnitas são avaliadas, $\varkappa = (\varkappa_1, \varkappa_2, \varkappa_3, \dots, \varkappa_w)$ são os pontos nos quais as funções incógnitas são avaliadas, $\mu = ((\mu_1^1, \mu_1^2, \dots, \mu_1^w), (\mu_2^1, \mu_2^2, \dots, \mu_2^w), \dots, (\mu_z^1, \mu_z^2, \dots, \mu_z^w))$ são as estimativas das funções incógnitas nos pontos \varkappa , $\gamma(\varkappa, \mu_i)$ é o método de interpolação, $\phi(\tau) = \langle \phi_1(\tau), \phi_2(\tau), \phi_3(\tau), \dots, \phi_z(\tau) \rangle$ são as funções interpoladas, sendo $\tau \in \Xi$ e $\Xi = [\check{\Xi}_0, \check{\Xi}_f]$ é o intervalo para o qual ϕ está definida.

Algoritmo 16: Interpolador de Soluções.

Entrada: $\gamma(\cdot), \varkappa, \mu$
Saída: $\phi(\cdot)$
1 *SOLUTIONS_INTERPOLATOR*($\gamma(\cdot), \varkappa, \mu$)
2 **início**
3 **para** $i = 1$ **até** z **faça**
4 $\phi_i \leftarrow \gamma(\varkappa, \mu_i)$; *%solução interpolada*
5 **fim**
6 **retorna** $\phi(\cdot)$;
7 **fim**

6.1.4 Algoritmo Principal do CODEMSOA

Enunciados os operadores e conceitos fundamentais, pode-se então enunciar o algoritmo principal do CODEMSOA, que consiste primeiramente na inicialização do método de otimização escolhido, que gerará os valores iniciais das variáveis de decisão, que também fazem parte do problema de valor inicial a ser resolvido.

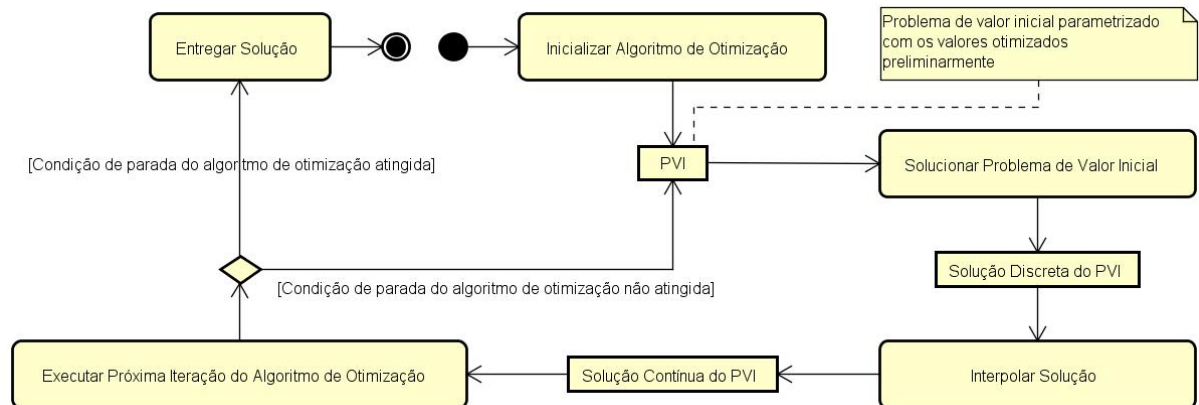
Em seguida, o problema de valor inicial é solucionado por um método numérico para equações diferenciais ordinárias anteposto, que gera uma solução discreta. Esta solução é interpolada através de um método de interpolação escolhido, originando uma solução contínua estimada para o problema de valor inicial.

Finalmente, executa-se a próxima iteração do algoritmo de otimização, que gera novos valores para as variáveis de decisão. Caso o critério de parada do método de otimização seja atendido, a solução do problema de otimização é entregue e o algoritmo é encerrado. Caso contrário, o problema de valor inicial é atualizado com os novos valores das variáveis de decisão, seguindo para ser novamente solucionado e a partir desta etapa, o algoritmo se repete.

A Figura 41 apresenta um diagrama que ilustra o fluxo de atividades do algoritmo CODEMSOA.

Descrevendo de forma pormenorizada, o CODEMSOA, inicialmente, secciona o método de otimização α em duas partes, sua parte inicial α_{\downarrow} , em que consiste a inicialização do método, e em sua parte recorrente α_{\cup} , que compõe o *loop* principal do método.

Figura 41 - Etapas do CODEMSOA.



Fonte: O autor, 2023.

A seguir, a parte inicial α_1 do algoritmo de otimização é executada, gerando a população inicial de soluções \mathcal{P}_0 . A seguir, segue-se para o *loop* principal do problema.

No início do *loop*, para cada solução a ser avaliada, aplica-se o solucionador de equações diferenciais ordinárias sobre o problema de valor inicial estabelecido, considerando os valores das variáveis de decisão da solução atual x , que retorna os pontos nos quais as funções incógnitas \varkappa foram avaliadas e a estimativa destas funções nestes pontos μ .

Subsequentemente, a solução discreta oriunda do solucionador de equações diferenciais ordinárias é submetida ao interpolador de soluções, que retorna o conjunto de funções contínuas ϕ que solucionam o problema de valor inicial em um dado intervalo.

A seguir, define-se o par ordenado ψ , composto pelos pontos \varkappa e pelas funções interpoladas ϕ , e cada uma das componentes das funções f^x , g^x e h^x através da aplicação do operador de valoração funcional sobre as respectivas componentes de f , g e h , onde o termo $\bar{\psi}$ representa a solução do problema de valor inicial interpolada, devidamente valorada pelo par ψ .

Após a realização das etapas acima descritas para cada solução presente na população, a parte recorrente do algoritmo de otimização é executada, realizando assim uma nova iteração deste, que resulta em uma nova população de soluções \mathcal{P}_{i+1} .

Neste momento, o critério de parada do método de otimização α é avaliado. Caso este tenha sido atendido, o conjunto atual de soluções \mathcal{P}_i é retornado. Caso contrário, o *loop* do algoritmo é novamente executado.

O Algoritmo 17 sintetiza integralmente o algoritmo CODEMSOA, que foi especificado acima, onde $m \in \mathbb{N}$ é o número de objetivos, $k \in \mathbb{N}$ é o número de restrições de igualdade, $l \in \mathbb{N}$ é o número de restrições de desigualdade, $x = (x_1, x_2, x_3, \dots, x_n)$ é um vetor de variáveis de decisão, $\epsilon = (\epsilon_1, \epsilon_2, \epsilon_3, \dots, \epsilon_n)$ são os limites inferiores das variáveis de decisão, $\omega = (\omega_1, \omega_2, \omega_3, \dots, \omega_n)$ são os limites superiores das variáveis de decisão, $f(x) =$

$\langle f_1(x), f_2(x), f_3(x), \dots, f_m(x) \rangle$ é a função de objetivos, $h(x) = \langle h_1(x), h_2(x), h_3(x), \dots, h_k(x) \rangle$ é a função de restrições de igualdade, $g(x) = \langle g_1(x), g_2(x), g_3(x), \dots, g_l(x) \rangle$ é a função de restrições de desigualdade, f^x , g^x e h^x são respectivamente as funções de objetivos, de restrições de igualdade e de restrições de desigualdade com o valor de $\bar{\psi}$ valorado conforme os resultados do PVI para a solução x em questão, $f^{\mathcal{P}_i}(x) = \langle f^x(x) \rangle \forall x \in \mathcal{P}_i$, $g^{\mathcal{P}_i}(x) = \langle g^x(x) \rangle \forall x \in \mathcal{P}_i$ e $h^{\mathcal{P}_i}(x) = \langle h^x(x) \rangle \forall x \in \mathcal{P}_i$

Algoritmo 17: CODEMSOA.

Entrada: $\alpha(\cdot), f(\cdot), g(\cdot), h(\cdot), \epsilon, \omega, \sigma(\cdot), \varepsilon(\cdot), \varsigma, \Xi$
Saída: \mathcal{P}_i

- 1 $CODEMSOA(\alpha(\cdot), f(\cdot), g(\cdot), h(\cdot), \epsilon, \omega, \sigma(\cdot), \varepsilon(\cdot), \varsigma, \Xi, \gamma(\cdot))$
- 2 **início**
- 3 $(\alpha_{\downarrow}, \alpha_{\circ}) \leftarrow \alpha;$
- 4 $\mathcal{P}_0 \leftarrow \alpha_{\downarrow}(\epsilon, \omega);$ %soluções geradas na inicialização de α
- 5 $i \leftarrow 0;$
- 6 **enquanto** (a condição de parada de α não for atingida) **faça**
- 7 **para cada** $x \in \mathcal{P}_i$ **faça**
- 8 $(\varkappa, \mu) \leftarrow EDO_SOLVER(\sigma(\cdot), \varepsilon(\cdot), \varsigma, \Xi, x);$
- 9 $\phi \leftarrow SOLUTIONS_INTERPOLATOR(\gamma(\cdot), \varkappa, \mu);$
- 10 $\psi \leftarrow \langle \varkappa, \phi(\cdot) \rangle;$
- 11 **para** $j = 1$ **até** m **faça**
- 12 $f_j^x \leftarrow VALORAR(f_j(\cdot), \psi);$
- 13 **fim**
- 14 **para** $j = 1$ **até** l **faça**
- 15 $g_j^x \leftarrow VALORAR(g_j(\cdot), \psi);$
- 16 **fim**
- 17 **para** $j = 1$ **até** k **faça**
- 18 $h_j^x \leftarrow VALORAR(h_j(\cdot), \psi);$
- 19 **fim**
- 20 **fim**
- 21 $\mathcal{P}_{i+1} \leftarrow \alpha_{\circ}(\mathcal{P}_i, f^{\mathcal{P}_i}(\cdot), g^{\mathcal{P}_i}(\cdot), h^{\mathcal{P}_i}(\cdot), \epsilon, \omega);$ %próxima iteração de α
- 22 $i \leftarrow i + 1;$
- 23 **fim**
- 24 **retorna** $\mathcal{P}_i;$
- 25 **fim**

6.2 Modelo Computacional

O modelo computacional desenvolvido para a implementação do CODEMSOA empregou uma abordagem multiparadigma, envolvendo predominantemente a orientação a objetos e também a orientação funcional.

O modelo em questão pode se dividir em três modelos menores. Esses modelos são:

o modelo do solucionador de equações diferenciais ordinárias, modelo do interpolador e modelo do otimizador. Cada um destes será apresentado e descrito abaixo.

No entanto, antes de apresentar tais modelos, serão mostradas algumas estruturas utilitárias e genéricas que estarão presentes neles.

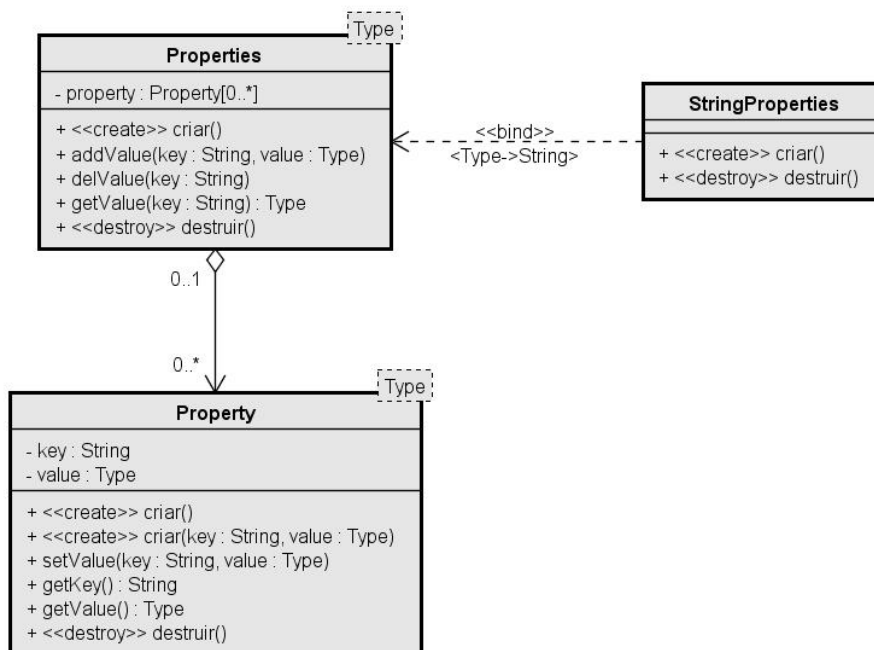
6.2.1 Estruturas Utilitárias

Algumas estruturas foram construídas para representar alguns tipos de dados que serão empregados pelos demais modelos componentes da ferramenta computacional modelada, consistindo assim em classes auxiliares do sistema. Neste modelo foram criadas três classes elementares, a saber, Property, Properties e StringProperties.

A classe Property define uma propriedade que possui um valor de qualquer tipo associado a uma chave do tipo String. A classe Properties por sua vez, gerencia um conjunto de propriedades, permitindo acessar valores por suas chaves. A classe StringProperties é uma especificação da classe Properties que admite apenas valores do tipo String em suas propriedades.

A Figura 42 ilustra o diagrama de classes das estruturas utilitárias com suas respectivas relações.

Figura 42 - Diagrama de classes das estruturas utilitárias.



Fonte: O autor, 2023.

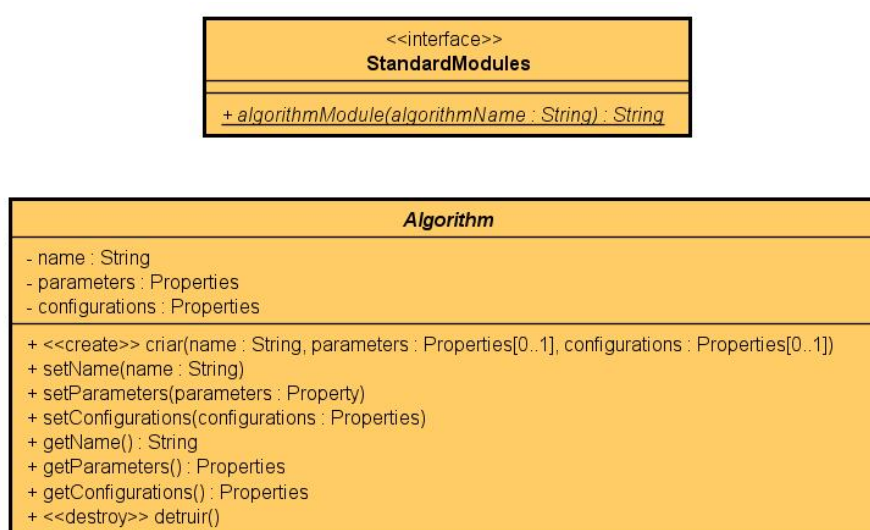
6.2.2 Estruturas Genéricas

Existem alguns recursos propostos que são compartilhados por todos os submodelos enumerados anteriormente. São eles: a interface `StandardModules` e a classe abstrata `Algorithm`.

A interface `StandardModules` estabelece quais as operações um computador dinâmico de módulos deve suportar e a classe `Algorithm` define o conjunto de atributos e operações que todo algoritmo deve possuir.

A Figura 43 ilustra as estruturas enunciadas.

Figura 43 - Diagrama de classes das estruturas genéricas.



Fonte: O autor, 2023.

6.2.3 Modelo do Solucionador de Equações Diferenciais Ordinárias

O presente modelo contém as estruturas propostas para a implementação do solucionador de equações diferenciais ordinárias do CODEMSOA. Para este modelo, foram propostas as seguintes classes, interfaces e recursos: `ODE`, `ODESolution`, `ODESolverAlgorithm`, `ODENumericalSolver`, `FunctionalODENumericalSolver`, `StandardODENumericalSolverModules`, `StandardODENumericalSolver`, `StandardFunctionalODENumericalSolver` e `FunctionalODENumericalSolver`.

A classe `ODE` modela uma equação diferencial ordinária associada a sua respectiva condição inicial e a classe `ODESolverAlgorithm` define, de forma específica, um método para equações diferenciais ordinárias com seus parâmetros e configurações.

A interface `ODENumericalSolver` define as operações que um solucionador de EDOs

deve suportar e a classe `StandardODENumericalSolver` fornece uma implementação para esta interface, sendo responsável por instanciar recursos, incluindo o algoritmo de resolução de equações diferenciais ordinárias, cujo o modulo é computado dinamicamente pela classe `StandardODENumericalSolverModules`, e solucionar o problema de valor inicial em questão.

Também foi oferecida uma interface funcional, caso não se deseje trabalhar diretamente com objetos. Esta é composta pela interface `FunctionalODENumericalSolver`, que define a operação de resolução parametrizada apenas em função de dados, não requerendo manipulação de objetos próprios do modelo.

A classe `StandardFunctionalODENumericalSolver` dispõe uma implementação para `FunctionalODENumericalSolver`, que consiste em formatar e manipular as entradas de dados nos objetos pertinentes, e usar `StandardODENumericalSolver` para resolver o problema de valor inicial.

A classe `ODESolution` modela a solução do problema de valor inicial, composta pelos pontos nos quais as funções incógnitas são avaliadas e pelas estimativas destas funções nestes pontos, além de outros dados, que podem estar presentes.

A Figura 44 ilustra o diagrama das classes que compõem o solucionador de equações diferenciais ordinárias com suas respectivas relações.

6.2.4 Modelo do Interpolador de Soluções

Este modelo é composto pelas estruturas aventadas para implementar o interpolador de soluções do CODEMSOA. As classes, interfaces e recursos propostos para este modelo são: `Interpolator`, `FunctionalInterpolator`, `InterpolationAlgorithm`, `StandardInterpolator`, `StandardFunctionalInterpolator` e `StandardInterpolationModules`.

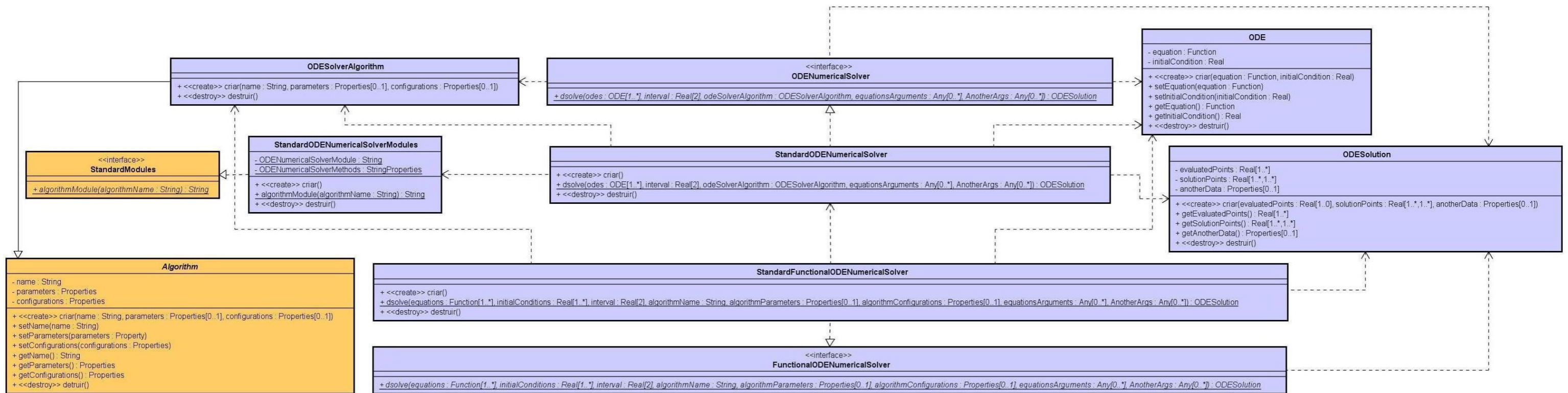
A classe `InterpolationAlgorithm` define de forma específica um método de interpolação com seus parâmetros e configurações.

A interface `Interpolator` define as operações que um interpolador deve oferecer e a classe `StandardInterpolator` oferece uma implementação para esta interface, sendo responsável por instanciar recursos, incluindo o algoritmo de interpolação, cujo o modulo é computado dinamicamente pela classe `StandardInterpolationModules`, e interpolar a solução numérica considerada.

Também neste modelo, foi disponibilizada uma interface funcional, que dispensa a manipulação direta de objetos. Esta consiste na interface `FunctionalInterpolator`, que define a operação de resolução parametrizada apenas em função de dados, não requerendo manipulação de objetos próprios do modelo.

A classe `StandardFunctionalInterpolator` fornece uma implementação para `FunctionalODENumericalSolver`, que consiste em formatar e manipular as entradas de dados

Figura 44 - Diagrama de classes do solucionador de equações diferenciais ordinárias.



Fonte: O autor, 2023.

nos objetos pertinentes, e usar `StandardInterpolator` para promover a interpolação.

A Figura 45 ilustra o diagrama das classes que compõem o interpolador de soluções com suas respectivas relações.

6.2.5 Modelo do Otimizador

O presente modelo é composto pelas estruturas propostas para a implementação do otimizador em que consiste a parte principal do CODEMSOA. Para o presente modelo, foram propostas as seguintes classes, interfaces e recursos: `OptimizationProblem`, `OptimizationSolution`, `Optimizer`, `FunctionalOptimizer`, `OptimizationAlgorithm`, `StandardOptimizer`, `StandardProblem`, `StandardFunctionalOptimizer` e `StandardOptimizationModules`.

A classe `OptimizationProblem` modela um problema de otimização com seus objetivos, variáveis, restrições e operações intermediárias; e a classe `OptimizationAlgorithm` define de forma específica um método de otimização com seus parâmetros e configurações.

A interface `Optimizer` define as operações que devem ser disponibilizadas por um otimizador e a classe `StandardOptimizer` oferece uma implementação para esta interface, sendo incumbida de alocar recursos, incluindo o algoritmo de otimização, cujo o módulo é computado dinamicamente pela classe `StandardOptimizationModules`, e solucionar o problema de otimização.

Para realizar as operações enunciadas acima, a classe `StandardOptimizer` faz uso de uma classe aninhada denominada `StandardProblem` para representar dinamicamente o problema, a cada iteração do método.

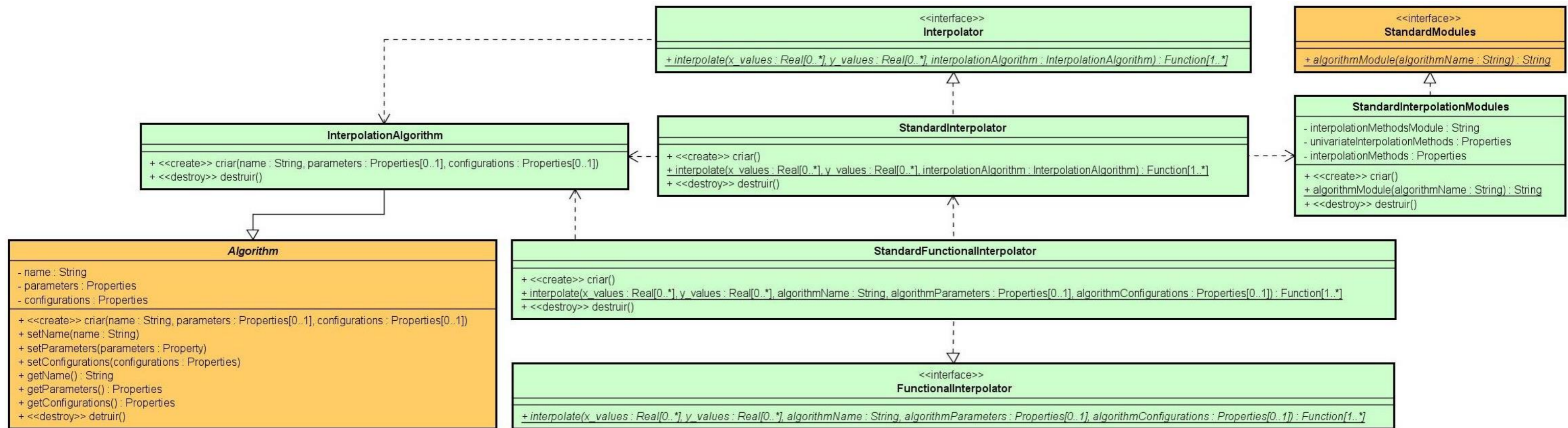
Assim como nos modelos apresentados anteriormente, neste modelo, está disponível uma interface funcional, que desobriga o uso direto de objetos do modelo. Esta é composta pela interface `FunctionalOptimizer`, que define a operação de resolução parametrizada apenas em função de dados, não requerendo manipulação de objetos próprios do modelo.

A classe `StandardFunctionalOptimizer` implementa `FunctionalOptimizer`, formatando e manipulando as entradas de dados nos objetos pertinentes, e usando `StandardOptimizer` para resolver o problema de otimização.

A classe `OptimizationSolution` modela a solução do problema de otimização, composta pelo conjunto ótimo de Pareto e a fronteira de Pareto, além de outros dados, que podem estar presentes.

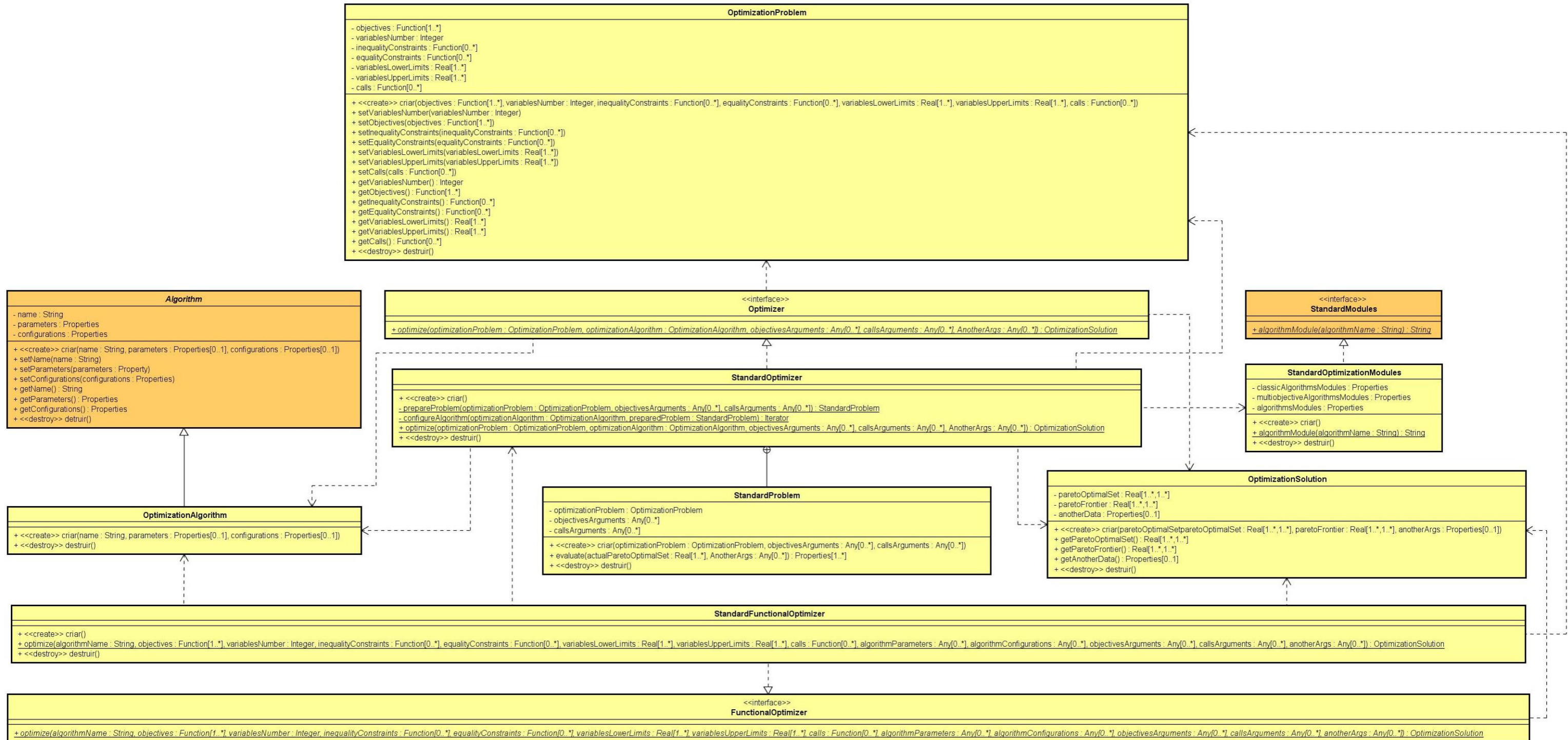
A Figura 46 ilustra o diagrama das classes que compõem o otimizador com suas respectivas relações.

Figura 45 - Diagrama de classes do interpolador de soluções.



Fonte: O autor, 2023.

Figura 46 - Diagrama de classes do otimizador.



Fonte: O autor, 2023.

6.2.6 Modelo Principal

Apresentados e descritos todos os componentes do presente modelo principal para implementação do CODEMSOA, Pode-se sintetizar este modelo, cujas estruturas estão ilustradas na Figura 47.

Nota ao Leitor 14 *Para viabilizar a apresentação de todas as classes e interfaces do modelo, foram mantidos apenas os principais atributos e operações de cada uma, sendo os argumentos e demais características omitidos.*

6.3 Detalhes de Implementação

O modelo apresentado foi implementado na forma de uma biblioteca de *software*, capaz de ser incorporada e utilizada em variados projetos de desenvolvimento de *software*, que necessitem de seus recursos.

O sistema que implementa o modelo enunciado, construído no decorrer da presente tese, foi integralmente desenvolvido na linguagem Python, afim de oferecer todo o arcabouço necessário em apenas uma plataforma de desenvolvimento.

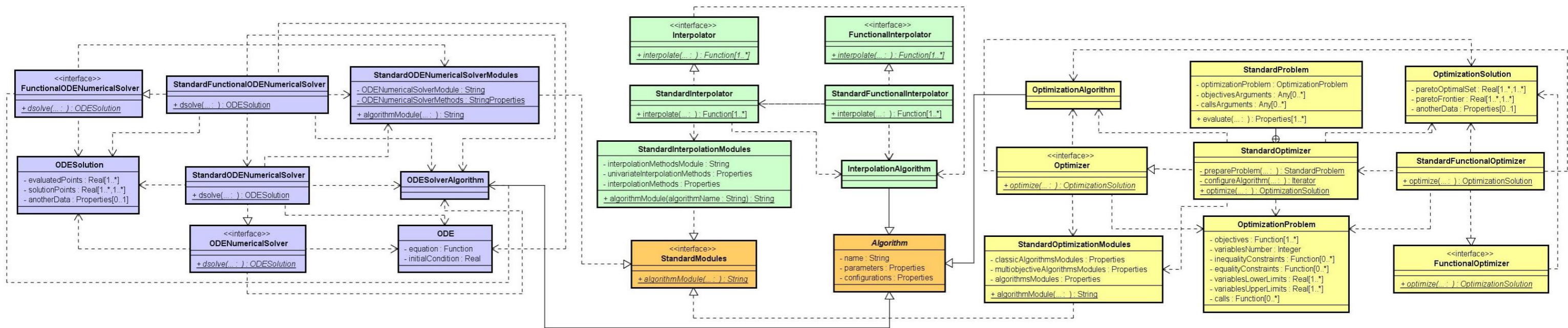
Os métodos numéricos para solução de equações diferenciais ordinárias, métodos de interpolação e métodos de otimização disponíveis na presente implementação do modelo de *software* apresentado estão enumerados no Apêndice A.

As informações inerentes à versão do Python e das bibliotecas de código que foram utilizadas e são conseqüentemente requeridas para a correta execução da biblioteca desenvolvida constam no Apêndice B.

As demais considerações acerca da implementação, tais como escolha de linguagem, paradigma, métodos e outros assuntos correlatos já foram discutidas na Metodologia da presente tese.

O Apêndice C apresenta um exemplo de uso da implementação do CODEMSOA dada pela biblioteca de código desenvolvida nesta tese.

Figura 47 - Diagrama de classes total do modelo.



Fonte: O autor, 2023.

7 FORMULAÇÃO DOS PROCESSOS E PROBLEMAS

Cada sonho que você deixa para trás é um pedaço do seu futuro que deixa de existir.

Steve Jobs

Neste capítulo, serão definidos e formulados os processos fermentativos considerados na presente tese e os problemas de otimização propostos e trabalhados nesta produção acadêmica.

Desse modo, serão apresentados aqui, as estruturas, modelos, parâmetros, valores e outros dados que estão relacionados aos processos fermentativos e aos problemas de otimização considerados.

7.1 Estudo de Caso 1: Fermentação de Lisina

Nesta seção, está formulado o processo fermentativo e os problemas de otimização que serão objetos da inquirição realizada neste estudo de caso.

7.2 Processo de Fermentação

O processo fermentativo considerado neste estudo de caso foi o mesmo tratado por Ani, Gujarathi e Vakili-Nezhaad (2021) cujo modelo consiste em um sistema composto por um reator de fermentação em batelada alimentada.

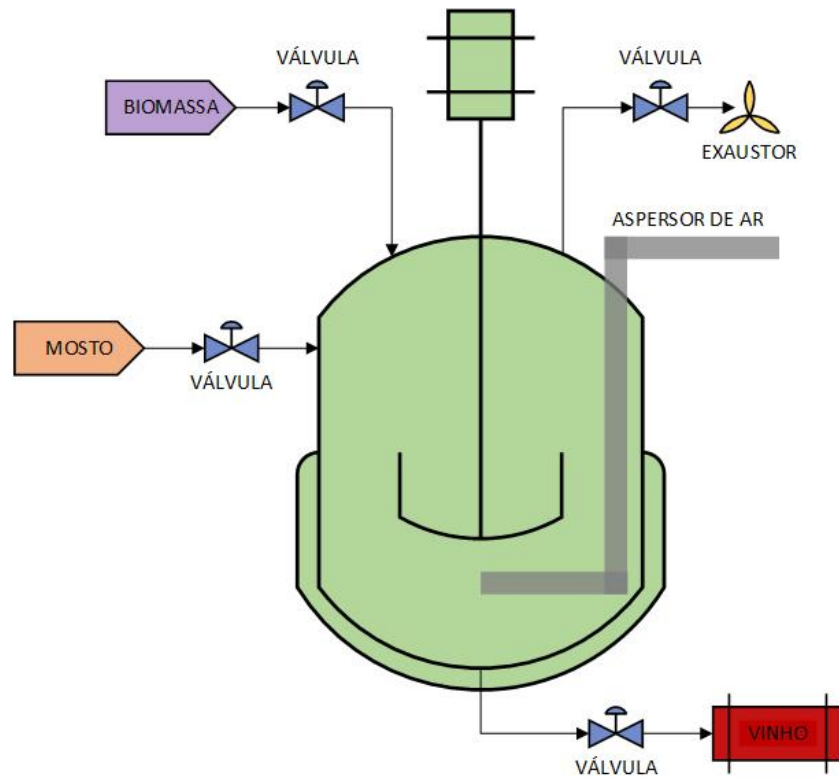
O modelo estrutural do reator de fermentação em batelada alimentada de lisina está ilustrado na Figura 48.

7.2.1 Modelo Cinético do Reator

A modelagem do processo de fermentação considerada neste estudo de caso empregou o modelo cinético apresentado por Ohno e Nakanishi (1976), composto pelas Equações (84), (85) e (86), em que S é a quantidade de substrato, V é o volume ocupado no reator, μ é a taxa de crescimento de células de biomassa, σ é a taxa de consumo de substrato e π é a taxa de formação de produto.

$$\mu = \mu_P \frac{S}{V} \quad (84)$$

Figura 48 - Diagrama de engenharia de um reator de lisina.



Fonte: Adaptado de Ani, Gujarathi e Vakili-Nezhaad, 2021.

$$\sigma = \frac{\mu}{\sigma_P} \quad (85)$$

$$\pi = -\pi_{P_1}\mu^2 + \pi_{P_2}\mu \quad (86)$$

Os balanços de massa dos componentes presentes no reator de fermentação conforme o instante de tempo, no modelo considerado, são dados pelo problema de valor inicial expresso na Equação (87), onde X é a quantidade de células de microrganismos, S é a quantidade de substrato, P é a quantidade de produto, V é o volume ocupado no reator, u é a vazão de entrada, C_{SF} é a concentração de substrato na vazão de entrada, X_0 é a quantidade inicial de microrganismos no reator, S_0 é a quantidade inicial de substrato no reator, P_0 é a quantidade inicial de produto no reator e V_0 é o volume inicial ocupado

no reator.

$$\left\{ \begin{array}{l} \frac{dX}{dt} = \mu X \\ \frac{dS}{dt} = uC_{SF} - \sigma X \\ \frac{dP}{dt} = \pi X \\ \frac{dV}{dt} = u \\ X|_{t_0} = X_0 \\ S|_{t_0} = S_0 \\ P|_{t_0} = P_0 \\ V|_{t_0} = V_0 \end{array} \right. \quad (87)$$

Os valores dos parâmetros do modelo cinético adotados neste estudo de caso são os mesmos que foram definidos por Ohno e Nakanishi (1976), estando estes dispostos na Tabela 1.

Tabela 1 - Valores dos parâmetros usados no modelo cinético de Ohno e Nakanishi (1976).

Parâmetro	Unidade	Valor
μ_P	g/L	0,125
σ_P	$g_{substrato}/g_{celulas}h$	0,135
π_{P_1}	$g_{produto}/g_{celulas}h$	384,0
π_{P_2}	$g_{produto}/g_{celulas}$	134,0

Fonte: O autor, 2023.

7.2.2 Especificações e Parâmetros de Simulação

As simulações de reator realizadas neste estudo de caso levaram em conta os parâmetros empregados por Ani, Gujarathi e Vakili-Nezhaad (2021). No caso em questão, infunde-se 0,1 g de bactérias e 14 g de açúcares, ocupando um volume de 5 L no reator.

Levando em conta as informações enunciadas, podem-se especificar os demais parâmetros envolvidos no processo em questão, com exceção dos que não foram propositalmente definidos por Ani, Gujarathi e Vakili-Nezhaad (2021), por estarem no conjunto de variáveis de decisão dos estudos de otimização tratados.

A princípio, serão especificados os instantes de tempo envolvidos no processo fer-

mentativo do reator, em horas, estando apresentados na Tabela 2, onde t_0 é o instante de início da operação, t_A é o instante de início da alimentação do reator, quando o alimentador é ligado, t_B é o instante de início da batelada, quando o alimentador é desligado, e t_f é o instante de término da operação.

Tabela 2 - Instantes de início das etapas do reator no Estudo de Caso 1.

Instante	Valor (h)
t_0	0,0
t_A	<i>Indefinido</i>
t_B	<i>Indefinido</i>
t_f	40,0

Legenda: No presente estudo de caso, t_A e t_B não estão definidos a priori.

Fonte: O autor, 2023.

Estabelecidos os instantes de tempo, define-se na sequência, a função u que modela a vazão de entrada de suco no reator de fermentação em batelada alimentada dada pela Equação (88), onde ν_u é a quantidade de suco entrante no reator, caso este esteja na etapa de alimentação.

$$u(t) = \begin{cases} \nu_u, & t_A \leq t < t_B \\ 0, & \text{caso contrário} \end{cases} \quad (88)$$

Os valores dos parâmetros referentes a vazão de entrada no reator estão estabelecidos na Tabela 3, estando em conformidade com as especificações dadas por Ani, Gujarathi e Vakili-Nezhaad (2021).

Finalmente, são definidas as condições iniciais do processo, que consistem nas quantidades iniciais de cada um dos componentes no reator, conforme mostrado na Tabela 4, onde X_0 é a quantidade inicial de microrganismos no reator, S_0 é a quantidade inicial de substrato no reator, P_0 é a quantidade inicial de produto no reator e V_0 é o volume inicial ocupado no reator.

Nota ao Leitor 15 Uma importante consideração é a de que, nos problemas de otimi-

Tabela 3 - Valores dos parâmetros de vazão de entrada do reator no Estudo de Caso 1.

Parâmetro	Unidade	Valor
ν_u	L/h	2
C_{SF}	g/L	2.8

Fonte: O autor, 2023.

Tabela 4 - Concentrações iniciais do reator no Estudo de Caso 1.

Parâmetro	Unidade	Valor
X_0	g	0,1
S_0	g	14,0
P_0	g	0,0
V_0	L	5,0

Fonte: O autor, 2023.

zação que serão definidos adiante, quando qualquer um dos parâmetros, definidos aqui, for uma variável de decisão, terá seu valor dado pelo processo de otimização, sendo desconsiderado seu valor prévio definido nesta seção.

7.3 Processo de Otimização

Este trabalho objetivou, neste estudo de caso, apresentar a flexibilidade do algoritmo CODEMSOA e de sua implementação proposta por esta tese, através da otimização de parâmetros inerentes ao processo fermentativo apresentado na seção anterior.

Para tal, foram considerados os problemas de otimização do segundo estudo de caso apresentado por Ani, Gujarathi e Vakili-Nezhaad (2021), que consistem em uma série de dois problemas de otimização, que consideram diferentes conjuntos de variáveis de decisão e objetivos.

Tais problemas já foram resolvidos por Ani, Gujarathi e Vakili-Nezhaad (2021). Dessa forma, o intuito de abordá-los aqui é contrastar os resultados obtidos nesta tese com os obtidos em Ani, Gujarathi e Vakili-Nezhaad (2021), além de apresentar análises e discussões acerca destes.

Os problemas de otimização considerados neste estudo de caso são um tipo de-

nominado Problema de Controle Ótimo (PCO) Multiobjetivo, que consiste em obter os perfis de variáveis de controle que otimizem determinados índices de desempenho, estando sujeitos a restrições (PFEIFER; LOBATO, 2013).

As subseções a seguir especificam e descrevem cada um dos problemas mencionados.

7.3.1 Produtividade e Rendimento

Antes de apresentar os problemas de otimização, serão enunciados os conceitos de produtividade e rendimento, expressos na forma em que estes foram empregados nos problemas de otimização considerados.

7.3.1.1 Produtividade

O conceito de produtividade empregado neste estudo consiste na razão entre a quantidade de produto gerado e o tempo gasto no processo, sendo modelado pela Equação (89), onde P é a quantidade de produto e t_f é o instante de término do processo.

$$Prod(t_f) = \frac{P(t_f)}{t_f} \quad (89)$$

7.3.1.2 Rendimento

O conceito de rendimento adotado neste caso consiste na razão entre a quantidade de produto formado e o quantidade de substrato consumida no processo, sendo modelado pela Equação (90), onde P é a quantidade de produto, S é a quantidade de substrato, V é o volume ocupado no reator, C_{SF} é a concentração de substrato no suco, t_0 é o instante de início do processo, t_f é o instante de término do processo e V_0^* é o valor nominal constante de 5 L, que indica o volume inicial ocupado no reator na configuração original do processo.

$$Rend(t_f) = \frac{P(t_f)}{S(t_0) + C_{SF}[V(t_f) - V_0^*] - S(t_f)} \quad (90)$$

7.3.2 Problema de Otimização A

O primeiro problema de otimização teve o intuito de aferir para quais tempos de alimentação, de operação e volumes iniciais ocupados no reator obtém-se o maior rendimento e produtividade de lisina.

Para isto, foram definidos como objetivos do problema de otimização em questão, a maximização da produtividade, uma vez que quanto mais rápida a geração de produto, menor o tempo gasto no processo, e a maximização do rendimento, pois quanto menor a quantidade de substrato gasto na formação de produto, maior a economia do insumo.

Como espera-se saber para quais tempos de alimentação, de operação e volumes iniciais ocupados no reator, os objetivos são melhor alcançados, foram definidas como variáveis de decisão deste problema de otimização, o instante de tempo t_A , que define quando a alimentação do reator começa, o instante t_B , que denota o término da alimentação do reator e o início da batelada, o instante t_f , que expressa o fim do processo fermentativo, e o volume V_0 que indica a ocupação inicial do reator.

Durante o processo de otimização, tanto t_A quanto t_B devem ter seus valores ajustados dentro do intervalo $[0,40]$, o instante t_f , por sua vez, deve estar contido em $[20,40]$, enquanto o volume V_0 deve ter seus valores ajustados sem sair do intervalo $[5,20]$.

Para que o processo de otimização respeite as limitações definidas, foram impostas restrições ao problema. A primeira determina que o instante t_A seja menor ou igual a t_B , uma vez que não faz sentido encerrar a alimentação do reator antes de iniciá-la. A segunda restrição impõe que o instante t_B não seja maior que t_f , pois não é consistente encerrar o processo antes de terminar a alimentação do reator.

Definidos os objetivos, variáveis e restrições, estabelece-se o problema de otimização considerado através da Equação (91), que o denota formalmente, onde $Prod$ é a produtividade e $Rend$ é o rendimento.

$$\begin{array}{ll}
 \text{Maximizar} & Prod(t_f) \\
 & Rend(t_f) \\
 \text{Sujeito a} & t_A \leq t_B \\
 & t_B \leq t_f \\
 & 0 \leq t_A \leq 40 \\
 & 0 \leq t_B \leq 40 \\
 & 20 \leq t_f \leq 40 \\
 & 5 \leq V_0 \leq 20
 \end{array} \tag{91}$$

Da perspectiva de um problema de controle ótimo, este problema objetiva aferir os perfis das variáveis de controle t_A , t_B , t_f e V_0 que otimizem os índices de desempenho

$Prod(t_f)$ e $Rend(t_f)$, sujeitos as restrições $0 \leq t_A \leq t_B$, $t_A \leq t_B \leq t_f$, $20 \leq t_f \leq 40$ e $5 \leq V_0 \leq 20$.

7.3.3 Problema de Otimização B

O segundo problema de otimização objetiva elencar para quais tempos de alimentação, de operação e volumes iniciais ocupados no reator obtém-se a maior produtividade de lisina, levando o menor tempo de operação.

Por se tratar de uma variante do problema anterior, este problema apresenta as mesmas variáveis, restrições e objetivos deste, com exceção do objetivo de maximização do rendimento, que deu lugar a um novo objetivo, que consiste na minimização do tempo de operação.

Desse modo, o presente problema de otimização é formalizado através da Equação (92).

$$\begin{aligned}
 & \underset{t_A, t_B, t_f, V_0}{\text{Maximizar}} && Prod(t_f) \\
 & \underset{t_A, t_B, t_f, V_0}{\text{Minimizar}} && t_f \\
 & \text{Sujeito a} && t_A \leq t_B \\
 & && t_B \leq t_f \\
 & && 0 \leq t_A \leq 40 \\
 & && 0 \leq t_B \leq 40 \\
 & && 20 \leq t_f \leq 40 \\
 & && 5 \leq V_0 \leq 20
 \end{aligned} \tag{92}$$

Do ponto de vista de um problema de controle ótimo, este problema objetiva aferir os perfis das variáveis de controle t_A , t_B , t_f e V_0 que otimizem os índices de desempenho $Prod(t_f)$ e t_f , sujeitos as restrições $0 \leq t_A \leq t_B$, $t_A \leq t_B \leq t_f$, $20 \leq t_f \leq 40$ e $5 \leq V_0 \leq 20$.

7.3.4 Inconsistências Identificadas nas Formulações Originais

Nas formulações consideradas neste estudo de caso, que foram originalmente dadas por Ani, Gujarathi e Vakili-Nezhaad (2021), identificou-se uma série de inconsistências, algumas presentes nas formulações, mas não na implementação e outras que acabaram por serem implementadas e impactarem os resultados obtidos.

A primeira inconsistência detectada foi a apresentação incorreta do modelo cinético proposto por Ohno e Nakanishi (1976), contendo erros em suas equações. Portanto, o

modelo expresso na Equação (87) desta tese difere do exposto em Ani, Gujarathi e Vakili-Nezhaad (2021), por conter a formulação correta do referido modelo cinético.

A segunda inconsistência identificada foi a fórmula do rendimento descrita por Ani, Gujarathi e Vakili-Nezhaad (2021), que a definiu como $\frac{P(t_f)}{C_{SF}(V(t_f)-V(0))}$. Entretanto essa fórmula leva os algoritmos de otimização a escolherem soluções que não apresentam quase nenhuma alimentação do reator, pois quanto menor a diferença entre a quantidade de substrato no início e no fim do processo, menor o denominador e conseqüentemente, maior o rendimento. Logo se essa diferença tender a zero, o rendimento tenderá ao infinito.

O problema da fórmula exposta é não levar em conta a quantidade de substrato no início e no final do processo, tal como considera a expressa na Equação (90) desta tese. Como o comportamento descrito não foi visto nos resultados expressos por Ani, Gujarathi e Vakili-Nezhaad (2021), concluiu-se que na implementação feita neste, foi empregada a fórmula dada na Equação (90).

A terceira inconsistência encontrada foi a unidade da vazão de entrada, que foi especificada por Ani, Gujarathi e Vakili-Nezhaad (2021) em g/h , enquanto a usada realmente em sua implementação foi em L/h . Esse erro pode induzir o leitor a acreditar que a vazão adotada é tal que forneça ao reator, uma determina quantidade de substrato por hora.

A quarta inconsistência detectada está relacionada à elicitação do volume inicial do reator como variável de decisão dos problemas de otimização formulados. A princípio, o valor do volume inicial mencionado em Ani, Gujarathi e Vakili-Nezhaad (2021) é de 5 L e a quantidade inicial de substrato é de 14 g. Neste sentido, a concentração inicial de substrato no reator é de 2,8 g/L, que é justamente a concentração presente na vazão de entrada. Dessa forma, conclui-se que 2,8 g/L é a concentração de substrato no suco, de modo geral.

Entretanto, embora exista uma relação direta entre a quantidade inicial de substrato e volume inicial de suco no reator, essa quantidade inicial de substrato não está expressa em função do volume inicial. Desse modo, quando o algoritmo de otimização avaliar soluções com maiores volumes iniciais, a quantidade inicial de substrato de 14 g permanecerá a mesma.

Assim, ao se iniciar o reator com qualquer volume acima de 5 L, todo volume que exceder 5 L não será referente a suco, mas simplesmente à adição de água.

A quinta inconsistência identificada está associada ao volume inicial presente a fórmula do rendimento adotada na implementação do problema em Ani, Gujarathi e Vakili-Nezhaad (2021). Como o volume inicial é uma variável de decisão, é natural que este mude a cada solução testada.

No entanto, na implementação adotada por Ani, Gujarathi e Vakili-Nezhaad (2021), o valor inicial na fórmula do rendimento é sempre 5, independente do valor inerente às soluções avaliadas pelo algoritmo de otimização.

Essa conclusão foi obtida após algumas simulações preliminares realizadas neste trabalho, que permitiram constatar que os valores dos resultados obtidos por Ani, Gujarathi e Vakili-Nezhaad (2021) só são atingidos na presença das configurações aferidas.

Como o objetivo deste estudo de caso foi empregar o CODEMSOA para resolver os problemas de otimização e contrastar seus resultados com os obtidos por Ani, Gujarathi e Vakili-Nezhaad (2021), a fim de atestar a flexibilidade do algoritmo e de sua implementação proposta por esta tese, a implementação dos problemas realizada neste trabalho considerou as mesmas configurações usadas na implementação feita por Ani, Gujarathi e Vakili-Nezhaad (2021), inclusive a quarta e quinta inconsistências.

7.4 Estudo de Caso 2: Fermentação Alcoólica

Nesta seção, estão formulados o processo fermentativo considerado e os problemas de otimização propostos, que serão objetos da inquirição promovida neste estudo de caso.

7.5 Processo de Fermentação

O processo fermentativo considerado neste estudo de caso foi relatado por Fonseca, Costa e Cruz (2017). Seu modelo consiste em um sistema envolvendo reatores de fermentação descontínua alimentada.

O sistema de fermentação em batelada alimentada simulado neste estudo de caso foi modelado de acordo com as especificações de uma planta de fermentação industrial brasileira instalada no estado de São Paulo, descrita por Fonseca, Costa e Cruz (2017).

Cada reator da referida planta realiza um ciclo de fermentação de 12h, que consiste em 30min de inoculação de leveduras, 5h de injeção de mosto, 3h de batelada de fermentação, 1h30min de drenagem e 2h de higienização e inatividade. Após esta sequência de processos, um novo ciclo de fermentação se inicia.

A reciclagem das células do creme de leveduras é realizada através de um centrífuga que separa a vazão de vinho fermentado em uma corrente de vinho centrifugado sem fermento e outra de creme de microrganismos com 180 kg/m^3 de biomassa celular de leveduras.

Após o processo de centrifugação, cerca de 7,5% do creme de levedura separado é descartado e o restante segue para o tratamento com ácido para descontaminação, a fim de retornar aos reatores após este processo.

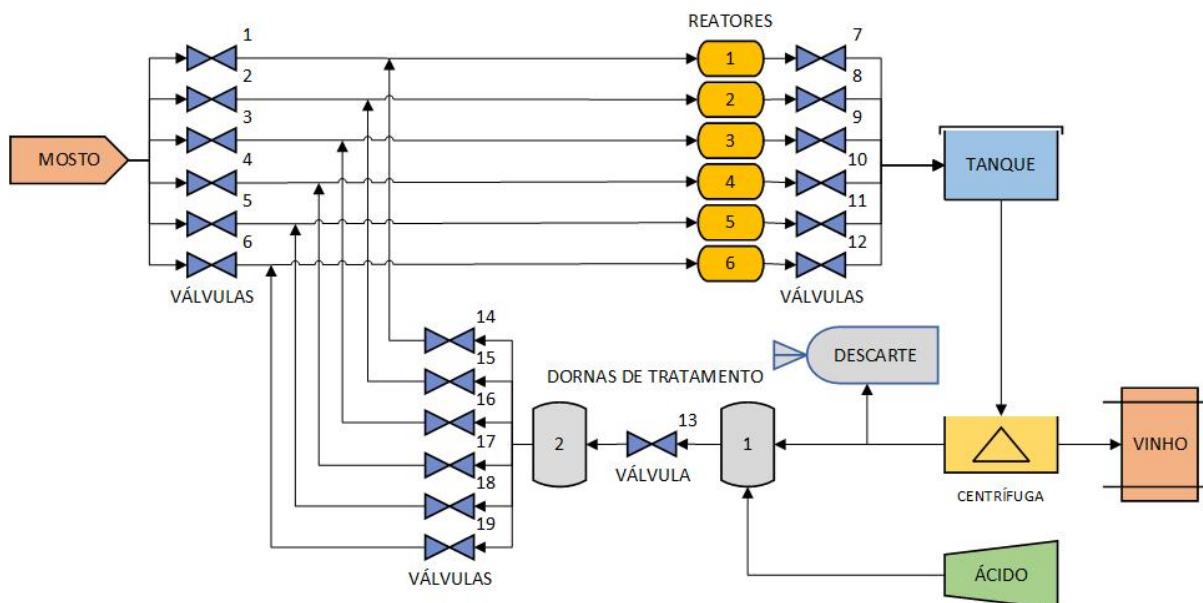
O modelo estrutural da planta de fermentação em batelada alimentada em questão apresenta reatores descontínuos operando em paralelo, um tanque de armazenamento de suco fermentado, uma centrífuga de separação, dornas de tratamento e válvulas de

controle.

Nota ao Leitor 16 Como no presente trabalho, o interesse está voltado para os reatores de fermentação, os demais elementos não serão especificados e detalhados aqui.

O modelo de unidade de fermentação ilustrado na Figura 49 apresenta uma simplificação da planta de fermentação industrial em questão.

Figura 49 - Diagrama de engenharia da planta de fermentação.



Fonte: Adaptado de Fonseca, Costa e Cruz, 2017.

7.5.1 Modelo Cinético do Reator

A modelagem do processo fermentativo considerada neste estudo de caso empregou o modelo cinético de fermentação etanólica proposto por Ghose e Tyagi (1979), expresso na forma dada pela Equação (93), onde μ é a taxa de crescimento celular, μ_{max} é a taxa máxima de crescimento celular, C_S é a concentração de substrato, K_S é a constante de saturação ou limitação pelo substrato, K_I é a constante de inibição pelo substrato, C_P é a concentração do produto e C_P^* é a concentração máxima de produto ou constante de inibição pelo produto.

$$\mu = \mu_{max} \frac{C_S}{K_S + C_S + \frac{C_S^2}{K_I}} \left(1 - \frac{C_P}{C_P^*} \right) \quad (93)$$

O modelo do processo fermentativo que estabelece os balanços de massa dos componentes presentes em um reator, conforme o instante de tempo, que foi considerado,

tem sua forma dada pelo problema de valor inicial expresso na Equação (94), onde t é o tempo, V é o volume ocupado pelo suco no reator, C_c é a concentração de um dado componente no reator, F^\uparrow é a vazão de saída de suco do reator, F^\downarrow é a vazão de entrada de suco no reator, t_0 é o instante de tempo inicial, V_0 é o volume inicial ocupado no reator, C_{c0} é a concentração inicial de um dado componente no reator, C_c^\downarrow é a concentração de um dado componente na vazão de entrada no reator e r_c é a taxa de reação de um dado componente.

$$\left\{ \begin{array}{l} \frac{dV}{dt} = F^\downarrow - F^\uparrow \\ \frac{dC_c}{dt} = \frac{F^\downarrow}{V}(C_c^\downarrow - C_c) + r_c \\ V|_{t_0} = V_0 \\ C_c|_{t_0} = C_{c0} \end{array} \right. \quad (94)$$

No presente estudo de caso, foram considerados os três principais componentes envolvidos em um processo de fermentação, a saber, o substrato, as células de biomassa dos microrganismos e o produto.

Desta forma, o problema de valor inicial que modela o reator pode ser apresentado pela Equação (95), onde C_S é a concentração de substrato, C_X é a concentração de células de microrganismos, C_P é a concentração de produto, C_{S0} é a concentração inicial de substrato, C_{X0} é a concentração inicial de células de microrganismos, C_{P0} é a concentração inicial de produto, r_S é a taxa de reação do substrato, r_X é a taxa de reação dos microrganismos, r_P é a taxa de reação do produto.

$$\left\{ \begin{array}{l} \frac{dV}{dt} = F^\downarrow - F^\uparrow \\ \frac{dC_S}{dt} = \frac{F^\downarrow}{V}(C_S^\downarrow - C_S) + r_S \\ \frac{dC_X}{dt} = \frac{F^\downarrow}{V}(C_X^\downarrow - C_X) + r_X \\ \frac{dC_P}{dt} = \frac{F^\downarrow}{V}(C_P^\downarrow - C_P) + r_P \\ V|_{t_0} = V_0 \\ C_S|_{t_0} = C_{S0} \\ C_X|_{t_0} = C_{X0} \\ C_P|_{t_0} = C_{P0} \end{array} \right. \quad (95)$$

As taxas de reação dos componentes são expressões que denotam a velocidade do aumento ou redução de cada um dos componentes envolvidos. No caso em questão, as taxas de reação dos componentes considerados estão expressas na Equação (96), sendo a taxa de reação do substrato mostrada na Equação (96a), a taxa de reação dos mi-

crorganismos determinada pela Equação (96b) e a taxa de reação do produto dada pela Equação (96c), onde C_X é a concentração celular, $Y_{P/S}$ é o coeficiente de rendimento do produto, $Y_{X/S}$ é o coeficiente de rendimento celular, μ é a taxa de crescimento celular e K_D é a taxa de morte celular.

$$r_S = -\mu C_X \frac{1}{Y_{X/S}} \quad (96a)$$

$$r_X = C_X(\mu - K_D) \quad (96b)$$

$$r_P = \mu C_X \frac{Y_{P/S}}{Y_{X/S}} \quad (96c)$$

Os valores dos parâmetros do modelo cinético adotados nesta tese consistiram nos mesmos que foram estabelecidos no trabalho original realizado por Ghose e Tyagi (1979), que estão mostrados na Tabela 5.

Tabela 5 - Valores dos parâmetros usados no modelo cinético de Ghose e Tyagi (1979).

Parâmetro	Unidade	Valor
μ_{max}	h^{-1}	0,36
K_D	h^{-1}	0,0083
K_S	kg/m^3	0,48
K_I	kg/m^3	203,5
C_P^*	kg/m^3	90,0
$Y_{X/S}$	$kg_{biomassa}/kg_{substrato}$	0,035
$Y_{P/S}$	$kg_{produto}/kg_{substrato}$	0,48

Fonte: Adaptado de Fonseca, Costa e Cruz, 2017.

7.5.2 Especificações e Parâmetros de Simulação

As simulações de reator referentes a este estudo de caso, que foram promovidas nesta tese, consideraram os parâmetros empregados nos reatores da planta de fermentação industrial mencionada. Cada reator em questão possui a capacidade máxima de 1.699 m^3 . Nestes, na fase de inoculação, infunde-se 548 m^3 de creme de levedura, contendo 90 kg/m^3 de biomassa celular de microrganismos. Na fase de alimentação, injeta-se 230

m^3/h de mosto, contendo 20% de açúcares totais. Na fase de batelada, nada é inserido no reator. Na fase de drenagem, descarrega-se $1.132 \text{ m}^3/\text{h}$ de suco delevedurado. Na fase de inatividade, o reator é higienizado para o início do próximo ciclo de fermentação.

Diante das informações apresentadas, podemos especificar os demais parâmetros envolvidos no modelo considerado. Antes, porém, cabe uma consideração acerca das unidades métricas de concentração de substrato e de produto.

No jargão industrial, comumente as concentrações de substrato são expressas em %massa/massa, enquanto as concentrações de produto são expressas em %vol/vol. Entretanto, no modelo de fermentação considerado, estes parâmetros devem estar expressos em kg/m^3 a fim de estarem consonantes com as unidades em que estão os parâmetros do modelo cinético.

Neste sentido, a fim de dimensionar corretamente os valores no modelo, deve-se converter tanto a concentração de produto, dada em %vol/vol, quanto a concentração de substrato, dada em %massa/massa, em concentrações dadas em kg/m^3 .

Para promover a conversão de unidades descrita acima, deve-se empregar os fatores de conversão constantes na Equação (97), onde a Equação (97a) apresenta o fator de conversão de %vol/vol para kg/m^3 e a Equação (97b) mostra o fator de conversão de %massa/massa para kg/m^3 (FONSECA; COSTA; CRUZ, 2017).

$$1 \text{ \%vol/vol} = 7,893 \text{ kg/m}^3 \quad (97a)$$

$$1 \text{ \%massa/massa} = 10 \text{ kg/m}^3 \quad (97b)$$

Nota ao Leitor 17 *Nas quantificações de componentes que se sucederão, o produto e o substrato serão denotados em kg/m^3 , tendo as devidas conversões já sido realizadas.*

Inicialmente, definiremos os instantes de tempo de início de cada uma das etapas de um reator de fermentação em batelada alimentada, em horas, que estão apresentados na Tabela 6, onde t_I é o instante de início da fase de inoculação, t_A é o instante de início da fase de alimentação, t_B é o instante de início da fase de batelada, t_D é o instante de início da fase de drenagem e t_H é o instante de início da fase de higienização.

Nota ao Leitor 18 *No presente caso, o instante de tempo inicial é o mesmo instante em que se inicia a fase de inoculação no reator, isto é, $t_0 = t_I$.*

Estabelecidos os instantes de tempo das etapas do reator, é possível definir as funções que modelam as vazões de entrada e saída de suco e componentes no reator de fermentação. A função F^\downarrow , que modela a vazão de entrada de suco no reator, é dada pela Equação (98), onde ν_{F^\downarrow} é a quantidade de suco entrante no reator, caso este esteja na fase

Tabela 6 - Instantes de início das etapas do reator no Estudo de Caso 2.

Instante	Valor (h)
t_I	0,0
t_A	0,5
t_B	5,5
t_D	8,5
t_H	10,0

Fonte: O autor, 2023.

de alimentação.

$$F^\downarrow(t) = \begin{cases} \nu_{F^\downarrow}, & t_A \leq t < t_B \\ 0, & \text{caso contrário} \end{cases} \quad (98)$$

A função C_c^\downarrow , modela a quantidade de um dado componente na vazão de entrada de suco no reator, sendo dada pela Equação (99), onde $\nu_{C_c^\downarrow}$ é a quantidade deste componente que está entrando no reator através da vazão de entrada, caso este esteja na fase de alimentação. No presente caso, a função C_c^\downarrow é definida para $c = S, X, P$, modelando as quantidades entrantes de substratos, células de biomassa e produto, respectivamente.

$$C_c^\downarrow(t) = \begin{cases} \nu_{C_c^\downarrow}, & t_A \leq t < t_B \\ 0, & \text{caso contrário} \end{cases} \quad (99)$$

A função F^\uparrow , que modela a vazão de saída de suco no reator é dada pela Equação (100), onde ν_{F^\uparrow} é a quantidade de suco saindo do reator, caso este esteja na fase de drenagem.

$$F^\uparrow(t) = \begin{cases} \nu_{F^\uparrow}, & t_D < t < t_H \\ 0, & \text{caso contrário} \end{cases} \quad (100)$$

Os valores dos parâmetros inerentes à vazão de entrada e saída de suco e componentes no reator estão determinados na Tabela 7, estando consonantes com as especificações da planta de fermentação considerada.

Tabela 7 - Valores dos parâmetros de vazão de entrada e saída do reator no Estudo de Caso 2.

Componente	Unidade	Quantidade
$\nu_{F\downarrow}$	m^3/h	230,0
$\nu_{F\uparrow}$	m^3/h	1132,0
$\nu_{C_S\downarrow}$	kg/m^3	200,0
$\nu_{C_X\downarrow}$	kg/m^3	0,0
$\nu_{C_P\downarrow}$	kg/m^3	0,0

Fonte: O autor, 2023.

Também cabe definir, a partir das especificações dadas, o volume máximo comportado pelo reator de batelada alimentada. Este está definido na Tabela 8, onde V_{max} representa o parâmetro em questão.

Tabela 8 - Capacidade máxima do reator no Estudo de Caso 2.

Parâmetro	Unidade	Valor
V_{max}	m^3	1699,0

Fonte: O autor, 2023.

Finalmente, definimos as condições iniciais do processo, que consistem nas concentrações iniciais dos componentes no reator, conforme apresenta a Tabela 9, onde V_0 é o volume inicial ocupado no reator, C_{S_0} é a concentração inicial de substrato, C_{X_0} é a concentração inicial de células de microrganismos e C_{P_0} é a concentração inicial de produto.

Tabela 9 - Concentrações iniciais do reator no Estudo de Caso 2.

Componente	Unidade	Quantidade
V_0	m^3	548,0
C_{S_0}	kg/m^3	0,0
C_{X_0}	kg/m^3	90,0
C_{P_0}	kg/m^3	0,0

Fonte: O autor, 2023.

Nota ao Leitor 19 *Assim como no estudo de caso anterior, nos problemas de otimização que serão definidos adiante, quando qualquer um dos parâmetros, definidos aqui, for uma variável de decisão, terá seu valor dado pelo processo de otimização, sendo desconsiderado seu valor prévio definido nesta seção.*

7.6 Processo de Otimização

Esta tese visou aprimorar o processo fermentativo apresentado na seção anterior por meio da otimização dos parâmetros concernentes ao modelo de reator de fermentação em batelada alimentada inerente a este.

Tal escolha se deu em virtude do maior potencial de otimização se encontrar no reator, uma vez que é nele que o processo de geração etanólica acontece e é onde se envolve a maior parte dos parâmetros do modelo.

Neste sentido, foi proposta uma série de dois problemas de otimização considerando diferentes conjuntos de variáveis de decisão e objetivos, cujos resultados trouxessem contribuições relevantes para a indústria alcooleira e para a comunidade acadêmica.

Diferentemente do estudo de caso anterior, os problemas aqui propostos, bem como suas resoluções têm caráter inédito, não tendo sido propostos ou resolvidos por nenhum outro trabalho, até onde se sabe.

As próximas subseções apresentam e descrevem cada um dos problemas propostos por este trabalho.

7.6.1 Problema de Otimização A

A proposição do primeiro problema de otimização tem a intenção de verificar quais tempos de duração das fases de alimentação e de batelada oferecem a melhor produção alcoólica, conservação de microrganismos e conversão totais de açúcares.

Para tal, foram definidos como objetivos do problema de otimização proposto, três metas, sendo a primeira, a minimização da concentração de açúcares no vinho delevedurado, no instante em que termina a drenagem do reator $C_S(t_H)$, pois quanto menor a quantidade destes ao final do processo, maior a quantidade produto gerado em virtude da conversão destes em etanol pela levedura.

A segunda meta do problema de otimização consiste na maximização da concentração de células de biomassa de levedura no vinho delevedurado presente no reator, no instante em que termina a drenagem do reator $C_X(t_H)$, uma vez que quanto maior a quantidade de células vivas ao final do processo, maior o reaproveitamento destas para o próximo ciclo de fermentação.

A terceira e última meta do problema em questão é a maximização da concentração de produto, que neste caso é o etanol, no vinho delevedurado constante no reator, no instante em que termina a drenagem do reator $C_P(t_H)$, porque o objetivo do processo é gerar álcool. Assim, quanto maior a quantidade gerada, mais produtivo é o processo de fermentação.

Como espera-se conhecer para quais tempos de duração das fases de alimentação e de batelada os objetivos descritos são melhor atingidos, foram definidas como variáveis deste problema de otimização, o instante de tempo t_B , que define quando a alimentação do reator termina e começa a batelada, e o instante de tempo t_D , que expressa quando termina a fase de batelada e começa a drenagem do vinho.

Durante a otimização, tanto t_B , quanto t_D devem ter seus valores ajustados sem sair do intervalo $[t_A, t_D^{max}]$, onde $t_D^{max} = 20$, permitindo que o início da drenagem do reator ocorra em até 20h de operação, mas não mais que isso.

Nota ao Leitor 20 *Como no processo de otimização, t_D assumirá diferentes valores em cada uma das soluções, consideraremos que t_H se dará sempre 1h30min depois do início da drenagem do reator, isto é, $t_H = t_D + 1,5$, uma vez que este é o tempo de drenagem usado no problema original, e também que $\nu_{F\uparrow}$ está definido de modo a esvaziar completamente o reator dentro do intervalo de 1h30min.*

Para que o processo de otimização considere as limitações inerente ao processo fermentativo real, foram impostas uma série de restrições. A primeira impõe que o instante de tempo t_B seja menor ou igual ao instante t_D , pois não faz sentido o reator começar a ser drenado sem nem mesmo começar a batelada de fermentação, que neste caso, nem ocorreria.

A segunda restrição determina que o volume ocupado no reator de fermentação não exceda a capacidade máxima deste em nenhum momento durante a batelada de fermentação alcoólica.

Estando estabelecidos, os objetivos, variáveis e restrições, define-se o problema de otimização em questão através da Equação (101), que o expressa matematicamente, onde C_S , C_X e C_P são as soluções do problema de valor inicial dado pela Equação (95).

$$\begin{aligned}
 & \underset{t_B, t_D}{\text{Minimizar}} && C_S(t_H) \\
 & \underset{t_B, t_D}{\text{Maximizar}} && C_X(t_H) \\
 & && C_P(t_H) \\
 \text{Sujeito a} &&& t_B \leq t_D \\
 &&& V(t) \leq V_{max}, \quad t_A \leq t \leq t_H \\
 &&& t_A \leq t_B \leq t_D^{max} \\
 &&& t_A \leq t_D \leq t_D^{max}
 \end{aligned} \tag{101}$$

Da ponto de vista de um problema de controle ótimo, este problema objetiva aferir os perfis das variáveis de controle t_B e t_D que otimizem os índices de desempenho $C_S(t_H)$, $C_X(t_H)$ e $C_P(t_H)$, sujeitos as restrições $t_I \leq t_B \leq t_D$, $t_B \leq t_D \leq t_D^{max}$ e $V(t) \leq V_{max}$.

7.6.2 Problema de Otimização B

A proposta do segundo problema de otimização teve o intuito de elencar, além dos tempos de duração das fases de alimentação e de batelada do reator, as vazões de entrada de suco no reator para as quais têm-se a melhor produtividade de etanol, aproveitamento de microrganismos e conversão de açúcares.

O objetivo deste problema é permitir a análise da influência da vazão de entrada na produtividade do reator relacionado a diferentes tempos de alimentação e de batelada.

Por se tratar de uma expansão do problema anterior, o problema em questão apresenta os mesmos objetivos, variáveis e restrições que estão presentes nele, possuindo adicionalmente a vazão de entrada de mosto no reator $\nu_{F\downarrow}$ como variável de decisão.

Durante o ajuste de $\nu_{F\downarrow}$, este não deverá assumir valores que estejam fora do intervalo $[0, \nu_{F\downarrow}^{max}]$, onde $\nu_{F\downarrow}^{max} = 10000$, uma vez que não faz sentido existir uma quantidade negativa de vazão entrada, além desta não poder exceder a vazão máxima de entrada $\nu_{F\downarrow}^{max}$.

Nota ao Leitor 21 *Embora a vazão de entrada esteja elencada como variável de decisão neste problema, esta refere-se exclusivamente à quantidade de suco entrante no reator. O comportamento da função que modela a vazão de entrada manteve-se o mesmo empregado no problema original, conforme denota a Equação (98), sendo variável de decisão apenas o termo $\nu_{F\downarrow}$. Desse modo, não foi realizado por este trabalho qualquer estudo que visasse determinar a taxa ótima de alimentação de substrato no reator, como o realizado por Lobato (2008).*

Determinados os objetivos, variáveis e restrições, estabelece-se o presente problema de otimização por meio da Equação (102), que o representa matematicamente, em que C_S , C_X e C_P são as soluções do problema de valor inicial dado pela Equação (95).

$$\begin{aligned}
& \underset{\nu_{F\downarrow}, t_B, t_D}{\text{Minimizar}} && C_S(t_H) \\
& \underset{\nu_{F\downarrow}, t_B, t_D}{\text{Maximizar}} && C_X(t_H) \\
& && C_P(t_H) \\
\text{Sujeito a} &&& t_B \leq t_D \\
& && V(t) \leq V_{max}, \quad t_A \leq t \leq t_H \\
& && t_I \leq t_B \leq t_D^{max} \\
& && t_I \leq t_D \leq t_D^{max} \\
& && 0 \leq \nu_{F\downarrow} \leq \nu_{F\downarrow}^{max}
\end{aligned} \tag{102}$$

Da perspectiva de um problema de controle ótimo, este problema objetiva aferir os perfis das variáveis de controle $\nu_{F\downarrow}$, t_B e t_D que otimizem os índices de desempenho $C_S(t_H)$, $C_X(t_H)$ e $C_P(t_H)$, sujeitos as restrições $t_I \leq t_B \leq t_D$, $t_B \leq t_D \leq t_D^{max}$, $V(t) \leq V_{max}$ e $0 \leq \nu_{F\downarrow} \leq \nu_{F\downarrow}^{max}$.

Nota ao Leitor 22 *Os objetivos dos problemas de otimização propostos nesta seção consideram individualmente a concentração de cada um dos componentes no reator de fermentação. A opção de adotar diretamente as concentrações nos objetivos foi feita levando em conta o fato das especificações de componentes, tais como centrífuga e destilador, que sucedem o reator no processo fermentativo, serem definidas em concentrações, uma vez que estes equipamentos são projetados para concentrações específicas de células de biomassa e etanol. Dessa forma, o tomador de decisão poderá adotar soluções de acordo com a infraestrutura a sua disposição.*

8 SIMULAÇÕES, RESULTADOS, ANÁLISE E DISCUSSÃO

Eu prefiro ter perguntas que não podem ser respondidas a ter respostas que não podem ser questionadas.

Richard Feynman

Neste capítulo, estão especificadas e descritas as simulações promovidas por esta tese, detalhando os parâmetros, algoritmos e ambiente que estas utilizaram em suas respectivas execuções.

Também serão apresentados aqui, os resultados obtidos a partir das simulações mencionadas, além das análises e discussões que foram realizadas sobre estes, considerando seus respectivos contextos.

8.1 Ambiente

As simulações computacionais realizadas nesta tese foram executadas no Cluster do Laboratório de Computação de Alto Desempenho (LCAD) do Instituto Politécnico (IPRJ) da Universidade do Estado do Rio de Janeiro (UERJ).

Trata-se de um Cluster Linux, que possui oito nós de processamento, onde cada nó possui duas pastilhas de silício, cada uma com seis núcleos Hyper-threading (HT), totalizando 192 núcleos de processamento em todo cluster.

As simulações propostas neste trabalho foram realizadas em paralelo, umas com as outras, cada uma em um diferente nó do Cluster. Os detalhes de hardware de cada um dos nós do cluster estão mostrados e relacionados no Apêndice D.

8.2 Simulações e Resultados do Estudo de Caso 1

No primeiro estudo de caso, foram realizadas seis simulações computacionais que objetivaram resolver os problemas de otimização multiobjetivo considerados, através do *Computational Ordinary Differential Equations Modeled System Optimization Algorithm* (CODEMSOA), usando diferentes métodos de otimização.

Em todas estas simulações, foi adotado o Método de Runge-Kutta de 2ª Ordem (RK-2) para solução das diversas configurações do problema de valor inicial dado pela Equação (95) e *Splines* Cúbicas para a interpolação de suas soluções, que foram geradas no processo de otimização do CODEMSOA.

Tanto para o Problema A quanto para o Problema B, foram realizadas três simula-

ções de otimização com o CODEMSOA, a primeira empregando o método de otimização *Non-dominated Sorting Genetic Algorithm II* (NSGA-II), a segunda o método *Adaptive Geometry Estimation Based Multiobjective Evolutionary Algorithm* (AGE-MOEA) e a terceira o *Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization* (C-TAEA).

Nota ao Leitor 23 *O uso de diferentes algoritmos de otimização realizado neste trabalho não teve qualquer pretensão de realizar análises e comparações entre estes algoritmos, nem mesmo relacionadas à específica aplicação deles aos problemas considerados. Seu emprego foi feito apenas para mostrar a flexibilidade do algoritmo proposto em trabalhar com variados métodos de otimização, bem como o suporte da biblioteca de software desenvolvida a diferentes algoritmos, além da apresentação de diferentes soluções para os problemas propostos, esperando que estas corroborassem umas com as outras.*

Em todas as simulações de otimização, para todos os métodos de otimização empregados, trabalhou-se com uma população de 600 indivíduos e 50 gerações evolutivas, sendo esta última, o critério de parada adotado para os algoritmos de otimização. Em cada geração, foram concebidos 600 descendentes, totalizando 1200 indivíduos, dos quais 600 são escolhidos para a geração seguinte, sem apresentar indivíduos repetidos.

Para o método numérico de equações diferenciais ordinárias, foram especificados 1200 instantes de tempo igualmente espaçados dentro do intervalo $[0, 40]$ nos quais as funções incógnitas do problema de valor inicial dado pela Equação (87) foram estimadas.

Considerando os parâmetros descritos acima, verifica-se que em cada um dos problemas de otimização considerados, cada algoritmo resolveu o problema de valor inicial 600 vezes a cada iteração, resultando em 30.000 resoluções, o que totaliza 120.000 equações diferenciais resolvidas por simulação. Como foram realizadas três simulações para cada problema de otimização, têm-se um total de 720.000 equações diferenciais resolvidas ao todo.

As próximas subseções apresentarão os resultados e informações referentes à execução das simulações de otimização sobre os problemas A e B considerados no presente estudo de caso.

8.2.1 Problema de Otimização A

Nesta subseção, serão apresentadas as informações específicas e os resultados obtidos a partir de cada uma das simulações de otimização feitas sobre o Problema de Otimização A do Estudo de Caso 1.

8.2.1.1 Simulação com o NSGA-II

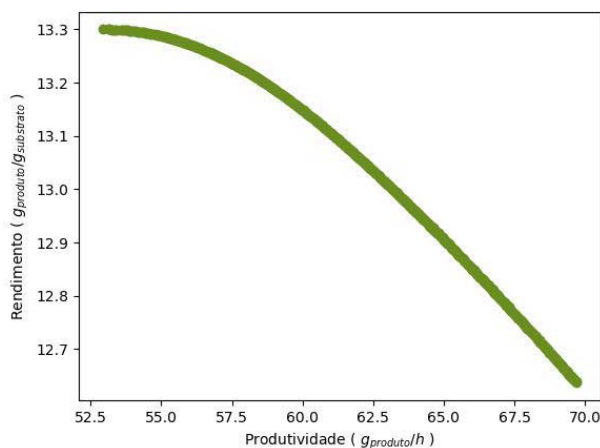
A simulação que buscou solucionar o Problema de Otimização A através do método NSGA-II levou 9 minutos e 57 segundos para atingir as 50 gerações. O método de otimização convergiu na 23^a geração, encontrando 600 soluções não dominadas.

As soluções obtidas nesta simulação estão descritas, detalhadas e ilustradas nas subseções abaixo.

8.2.1.1.1 Fronteira de Pareto

O Gráfico 3 apresenta a frente de Pareto do problema de otimização em questão, que foi estimada nesta simulação.

Gráfico 3 - Fronteira de Pareto do Problema A do Estudo de Caso 1 obtida pelo NSGA-II.

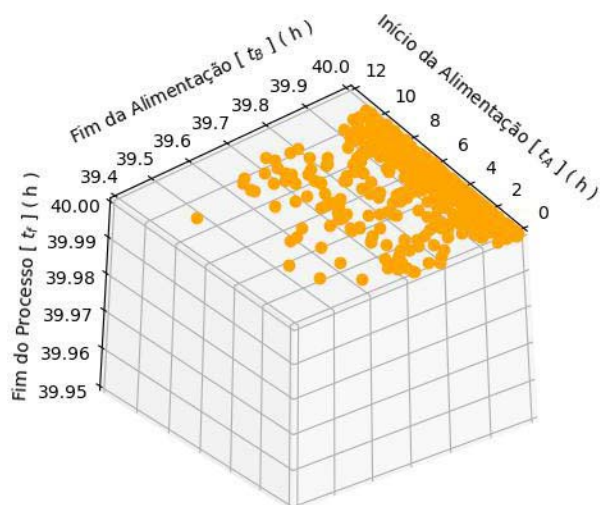


Fonte: O autor, 2023.

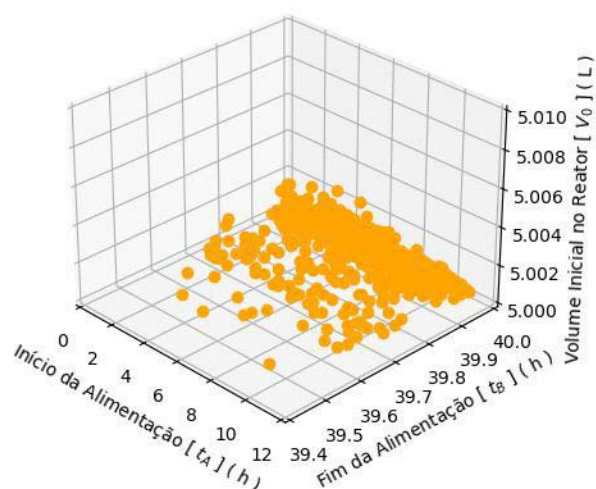
8.2.1.1.2 Conjunto Ótimo de Pareto

O Gráfico 4 mostra os otimizadores de Pareto encontrados nesta simulação de otimização, nos quais consiste o conjunto ótimo de Pareto estimado.

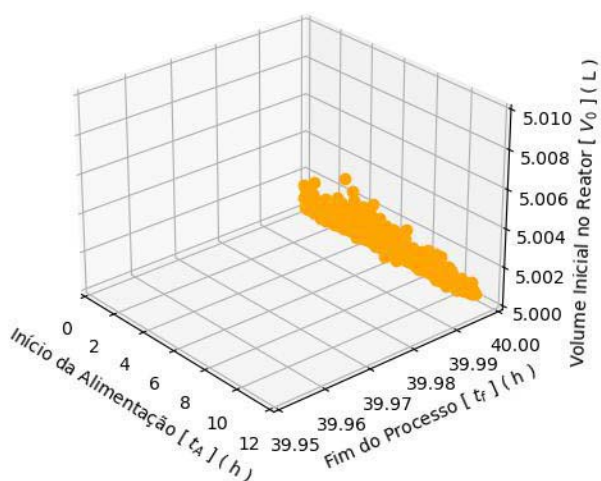
Gráfico 4 - Conjunto ótimo de Pareto do Problema A do Estudo de Caso 1 obtido pelo NSGA-II.



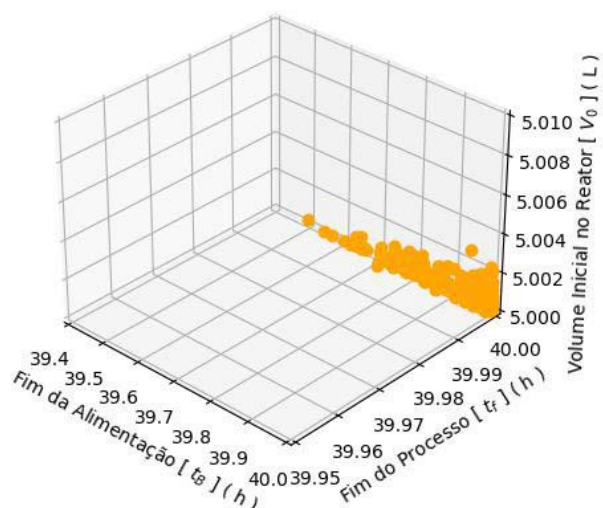
(a) I. Alim. (t_A) \times F. Alim. (t_B) \times F. Proc. (t_f).



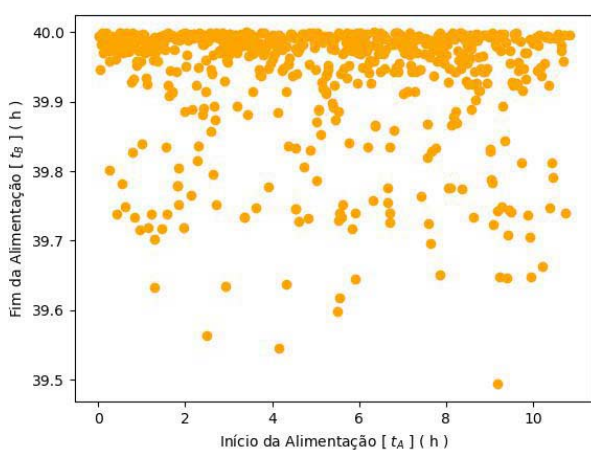
(b) I. Alim. (t_A) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).



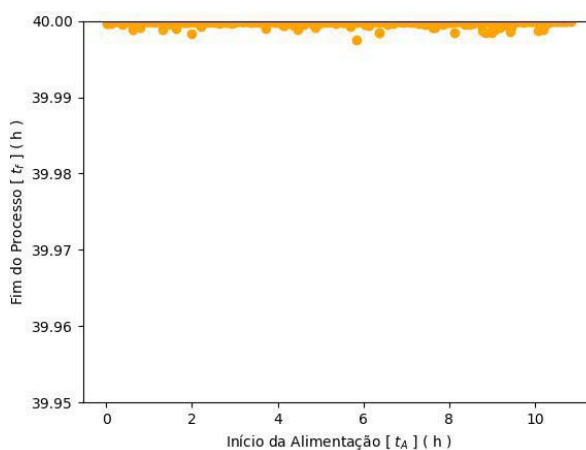
(c) I. Alim. (t_A) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).



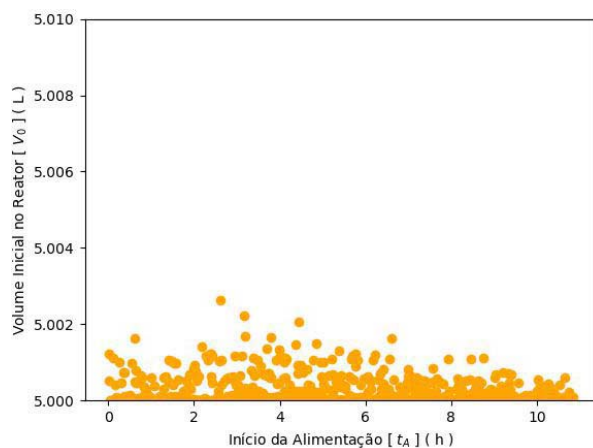
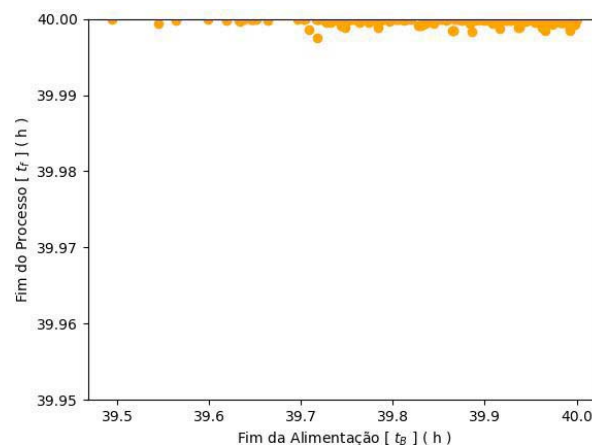
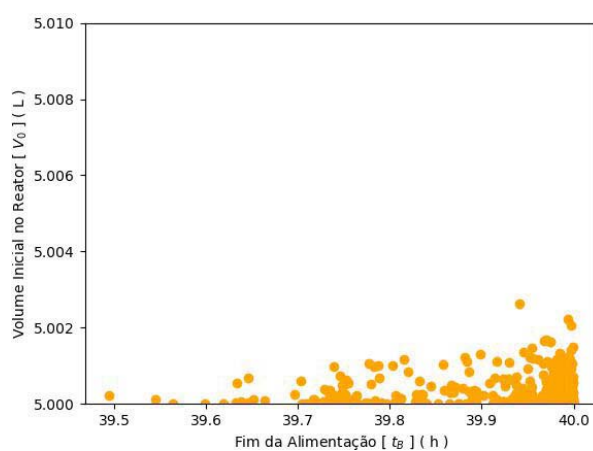
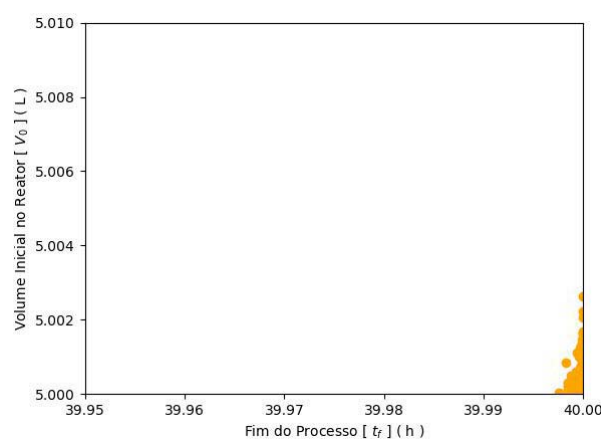
(d) F. Alim. (t_B) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).



(e) I. Alim. (t_A) \times F. Alim. (t_B).



(f) I. Alim. (t_A) \times F. Proc. (t_f).

(g) I. Alim. (t_A) \times V. Ini. Reat. (V_0).(h) F. Alim. (t_B) \times F. Proc. (t_f).(i) F. Alim. (t_B) \times V. Ini. Reat. (V_0).(j) F. Proc. (t_f) \times V. Ini. Reat. (V_0).

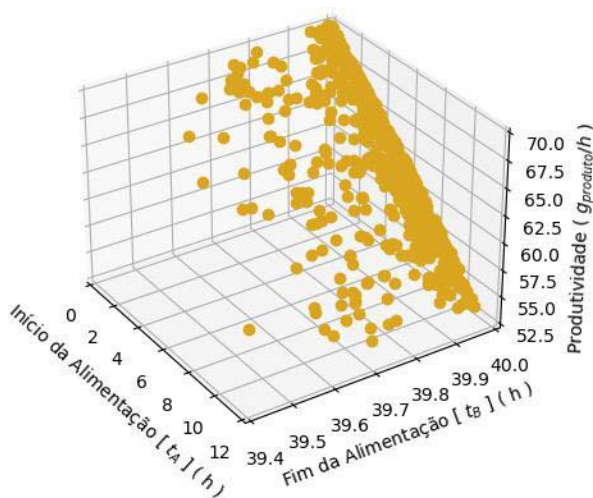
Fonte: O autor, 2023.

8.2.1.1.3 Relação entre os Otimizadores de Pareto e os Objetivos

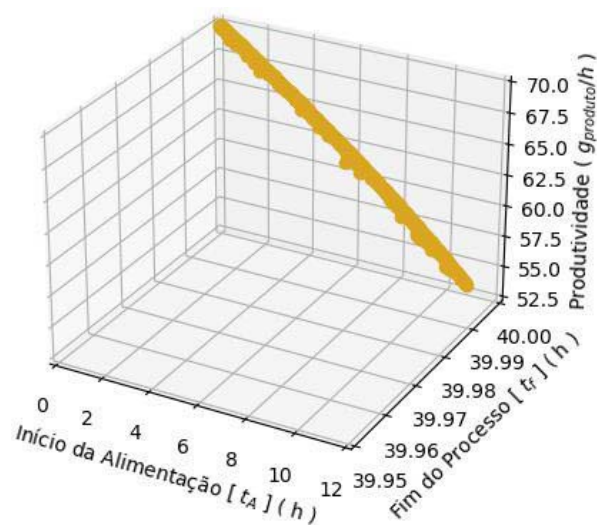
A seguir, serão apresentadas as correspondências dos pontos do conjunto ótimo de Pareto, mostrado no Gráfico 4, com cada um dos objetivos do problema de otimização em questão.

O Gráfico 5 mostra a produtividade relacionada a cada um dos otimizadores do conjunto ótimo de Pareto e o Gráfico 6 enuncia o rendimento relacionado a cada um dos otimizadores de Pareto encontrados.

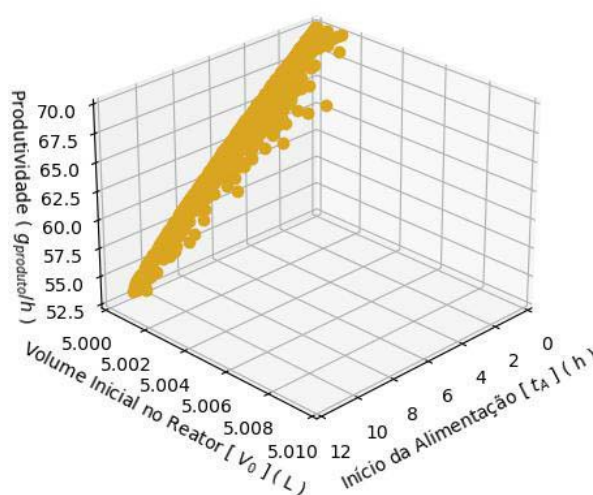
Gráfico 5 - Produtividade correspondente aos otimizadores de Pareto do Problema A do Estudo de Caso 1 obtidos pelo NSGA-II.



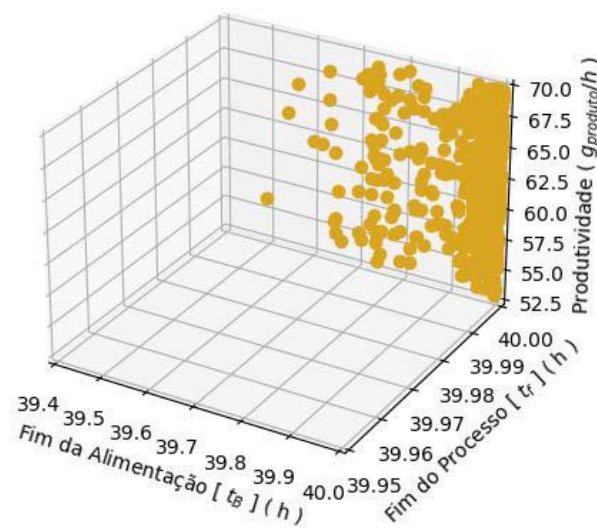
(a) Produtividade ($Prod$) \times I. Alim. (t_A) \times F. Alim. (t_B).



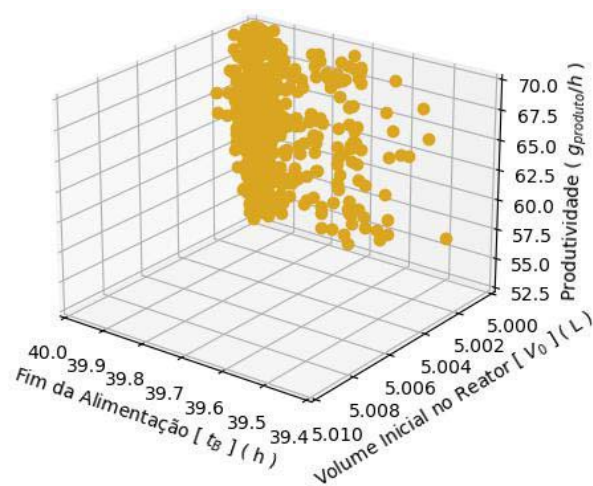
(b) Produtividade ($Prod$) \times I. Alim. (t_A) \times F. Proc. (t_f).



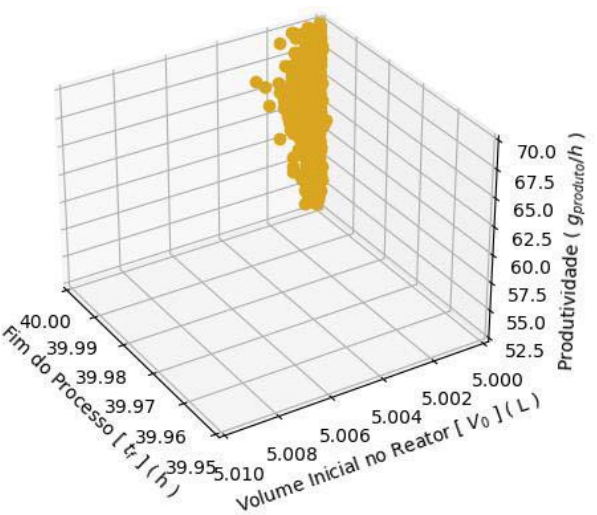
(c) Produtividade ($Prod$) \times I. Alim. (t_A) \times V. Ini. Reat. (V_0).



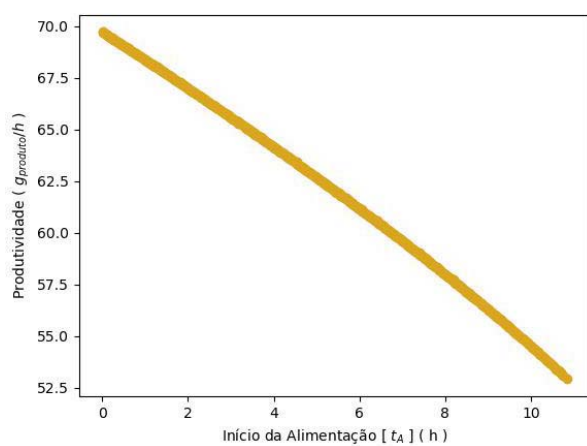
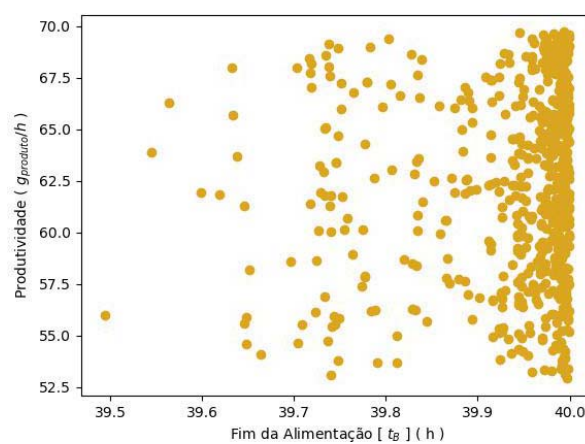
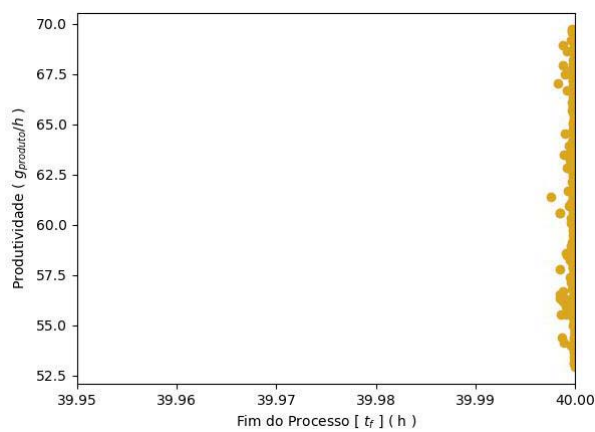
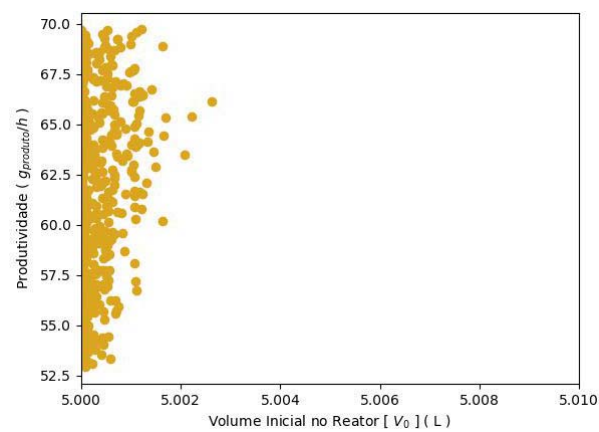
(d) Produtividade ($Prod$) \times F. Alim. (t_B) \times F. Proc. (t_f).



(e) Produtividade ($Prod$) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).

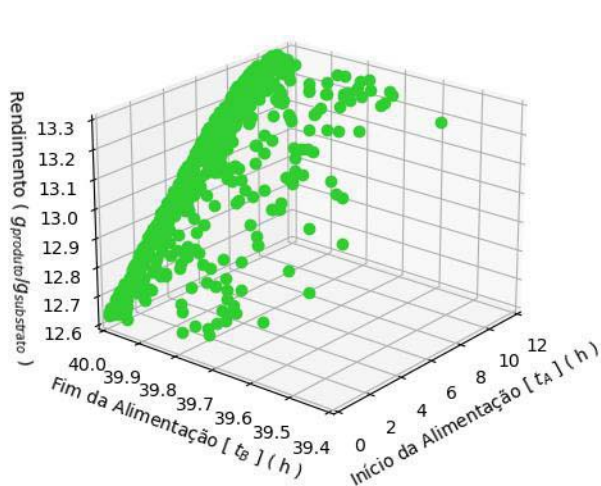


(f) Produtividade ($Prod$) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).

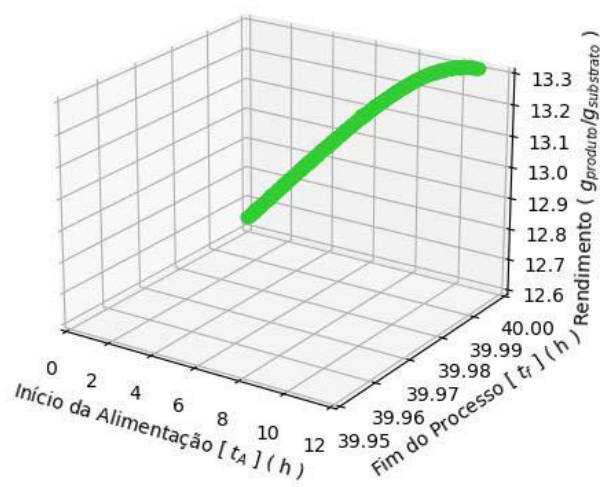
(g) Produtividade ($Prod$) \times I. Alim. (t_A).(h) Produtividade ($Prod$) \times F. Alim. (t_B).(i) Produtividade ($Prod$) \times F. Proc. (t_f).(j) Produtividade ($Prod$) \times V. Ini. Reat. (V_0).

Fonte: O autor, 2023.

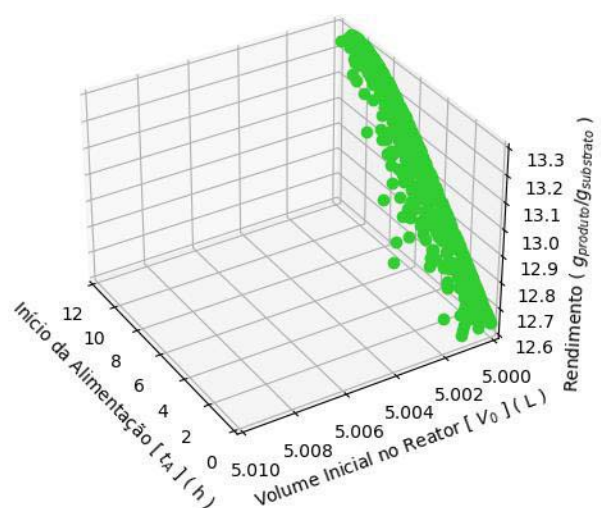
Gráfico 6 - Rendimento correspondente aos otimizadores de Pareto do Problema A do Estudo de Caso 1 obtidos pelo NSGA-II.



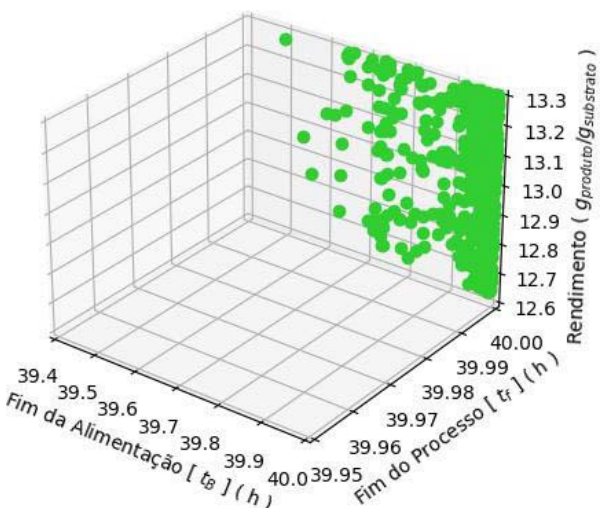
(a) Rendimento ($Rend$) \times I. Alim. (t_A) \times F. Alim. (t_B).



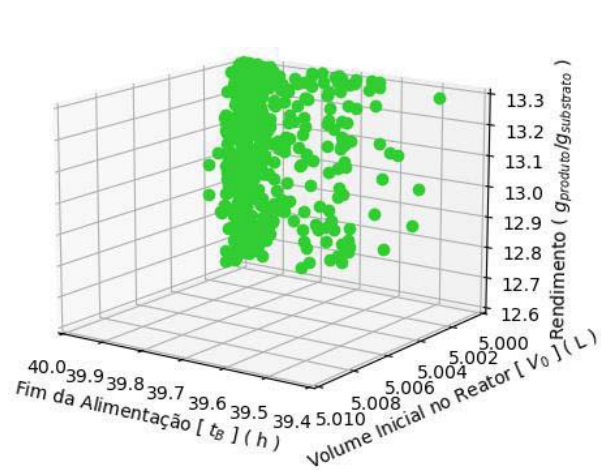
(b) Rendimento ($Rend$) \times I. Alim. (t_A) \times F. Proc. (t_f).



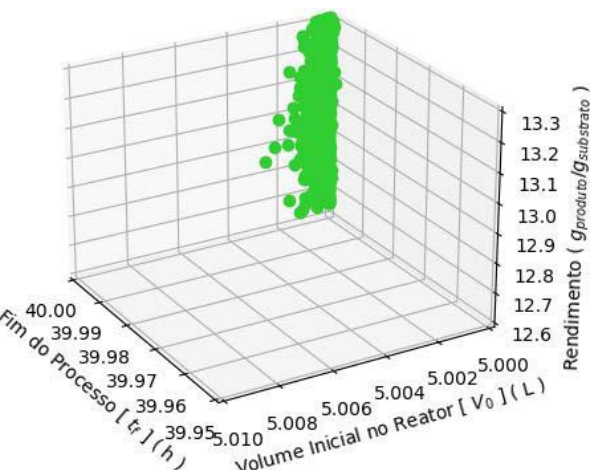
(c) Rendimento ($Rend$) \times I. Alim. (t_A) \times V. Ini. Reat. (V_0).



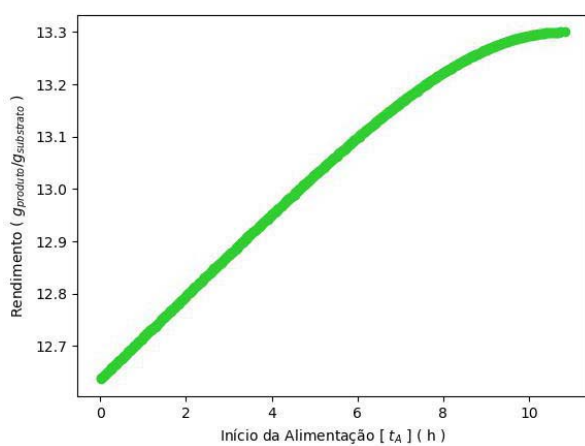
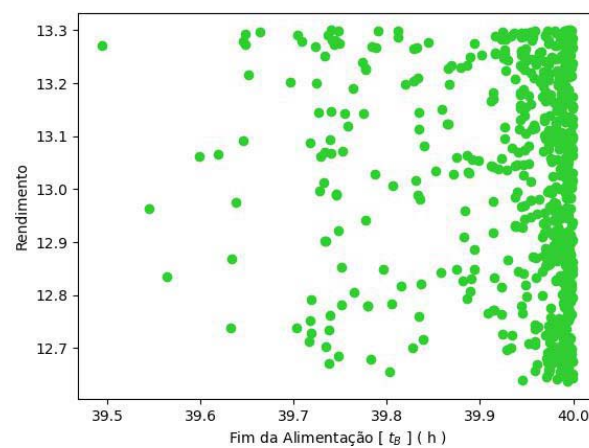
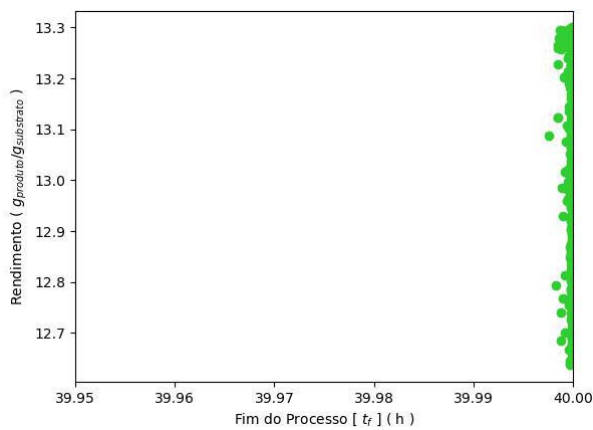
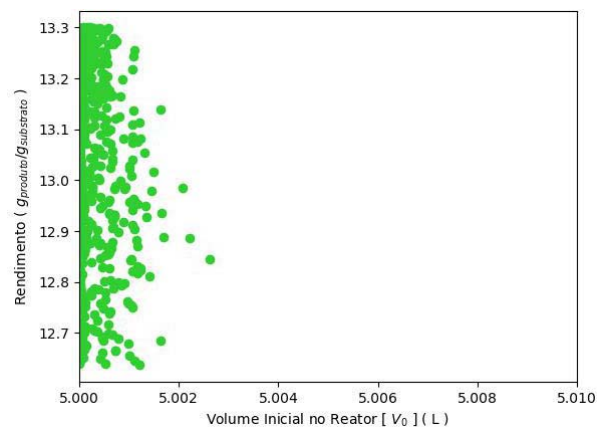
(d) Rendimento ($Rend$) \times F. Alim. (t_B) \times F. Proc. (t_f).



(e) Rendimento ($Rend$) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).



(f) Rendimento ($Rend$) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).

(g) Rendimento ($Rend$) \times I. Alim. (t_A).(h) Rendimento ($Rend$) \times F. Alim. (t_B).(i) Rendimento ($Rend$) \times F. Proc. (t_f).(j) Rendimento ($Rend$) \times V. Ini. Reat. (V_0).

Fonte: O autor, 2023.

8.2.1.2 Simulação com o AGE-MOEA

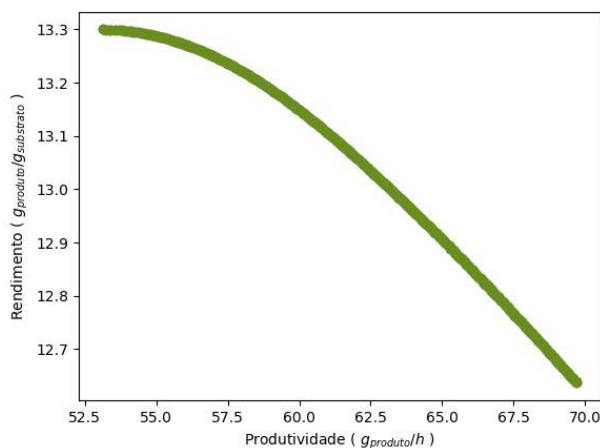
A simulação que buscou solucionar o Problema de Otimização A através do método AGE-MOEA 5 minutos e 33 segundos para atingir as 50 gerações. O método de otimização convergiu na 22^a geração, encontrando 600 soluções não dominadas.

As soluções obtidas nesta simulação estão descritas, detalhadas e ilustradas nas subseções abaixo.

8.2.1.2.1 Fronteira de Pareto

O Gráfico 7 apresenta a frente de Pareto do problema de otimização em questão, que foi estimada nesta simulação.

Gráfico 7 - Fronteira de Pareto do Problema A do Estudo de Caso 1 obtida pelo AGE-MOEA.

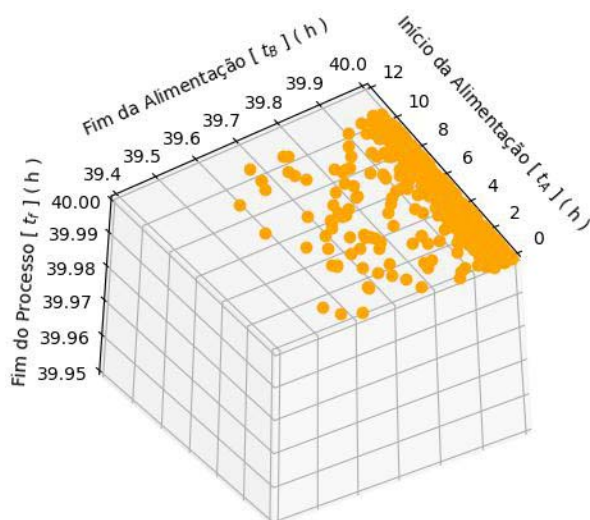


Fonte: O autor, 2023.

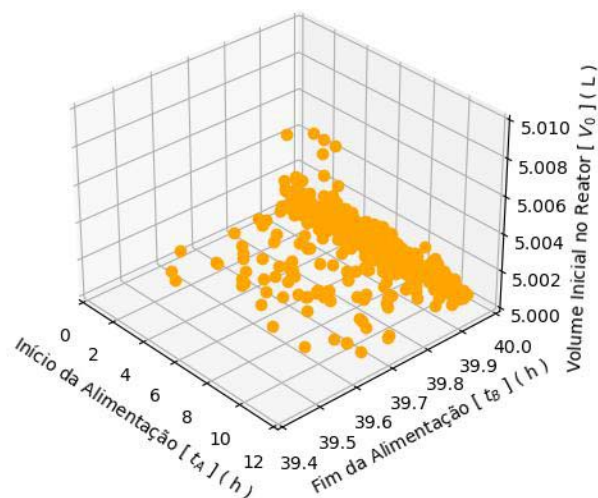
8.2.1.2.2 Conjunto Ótimo de Pareto

O Gráfico 8 mostra os otimizadores de Pareto encontrados nesta simulação de otimização, nos quais consiste o conjunto ótimo de Pareto estimado.

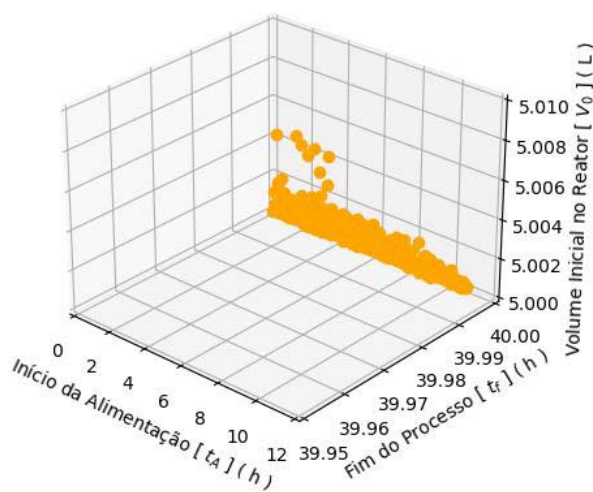
Gráfico 8 - Conjunto ótimo de Pareto do Problema A do Estudo de Caso 1 obtido pelo AGE-MOEA.



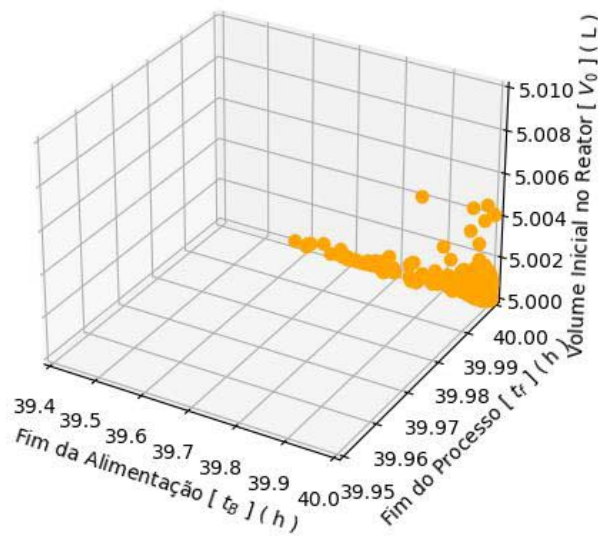
(a) I. Alim. (t_A) \times F. Alim. (t_B) \times F. Proc. (t_f).



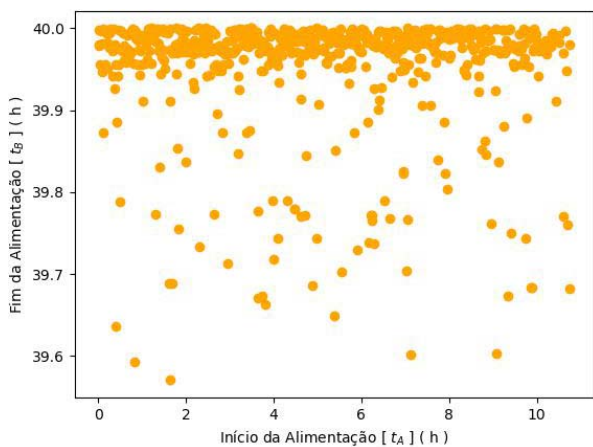
(b) I. Alim. (t_A) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).



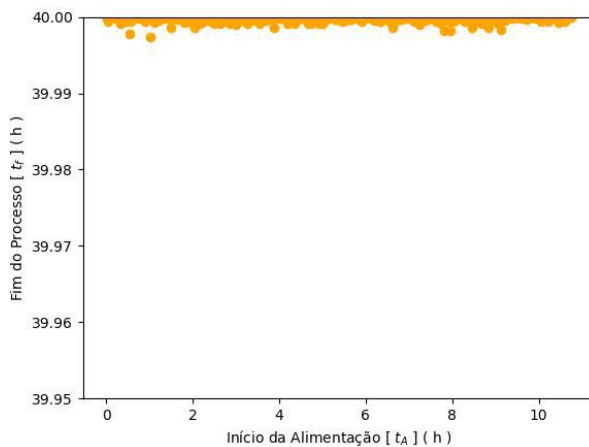
(c) I. Alim. (t_A) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).



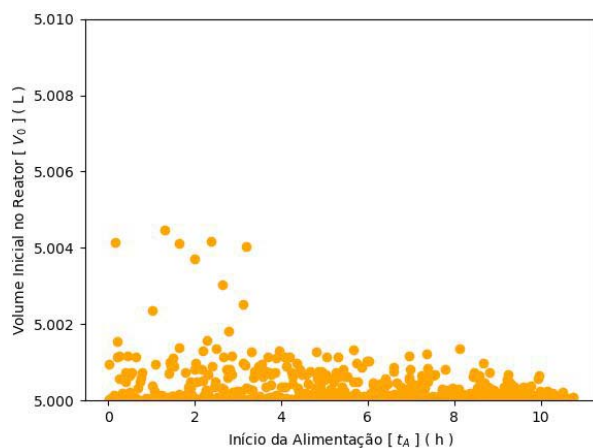
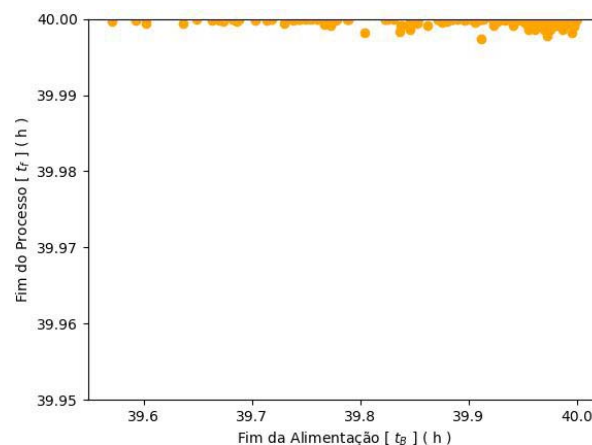
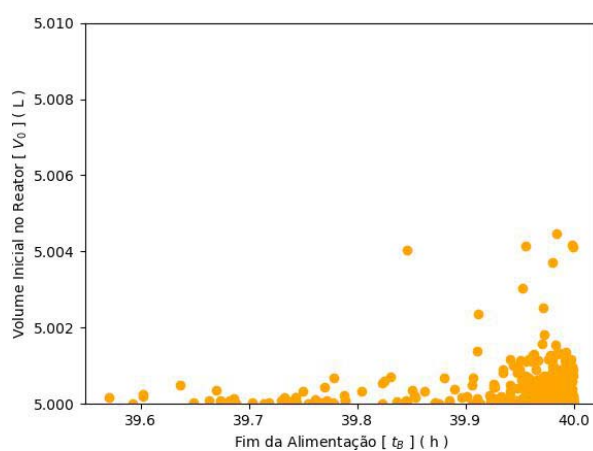
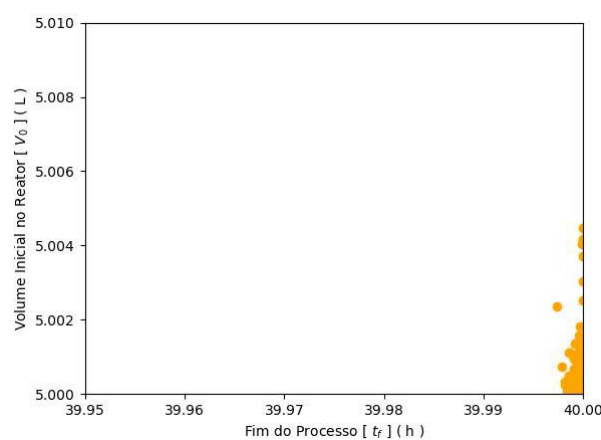
(d) F. Alim. (t_B) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).



(e) I. Alim. (t_A) \times F. Alim. (t_B).



(f) I. Alim. (t_A) \times F. Proc. (t_f).

(g) I. Alim. (t_A) \times V. Ini. Reat. (V_0).(h) F. Alim. (t_B) \times F. Proc. (t_f).(i) F. Alim. (t_B) \times V. Ini. Reat. (V_0).(j) F. Proc. (t_f) \times V. Ini. Reat. (V_0).

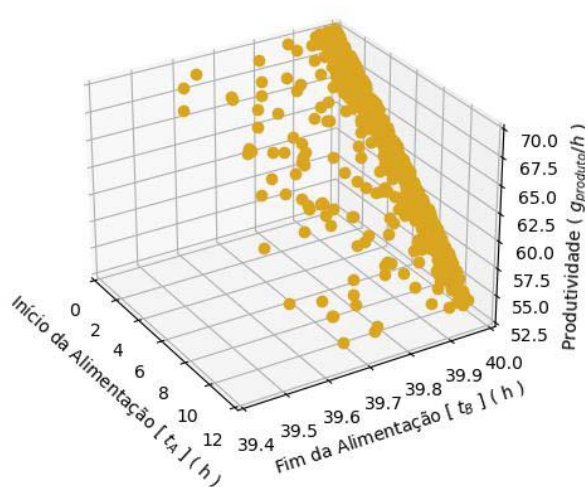
Fonte: O autor, 2023.

8.2.1.2.3 Relação entre os Otimizadores de Pareto e os Objetivos

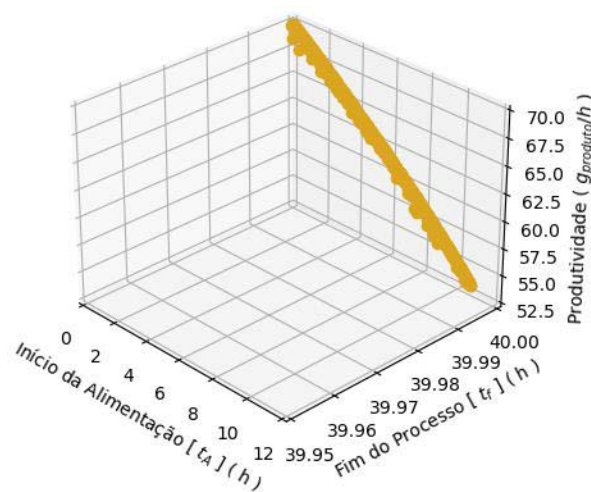
A seguir, serão apresentadas as correspondências dos pontos do conjunto ótimo de Pareto, mostrado no Gráfico 8, com cada um dos objetivos do problema de otimização em questão.

O Gráfico 9 mostra a produtividade relacionada a cada um dos otimizadores do conjunto ótimo de Pareto e o Gráfico 10 enuncia o rendimento relacionado a cada um dos otimizadores de Pareto encontrados.

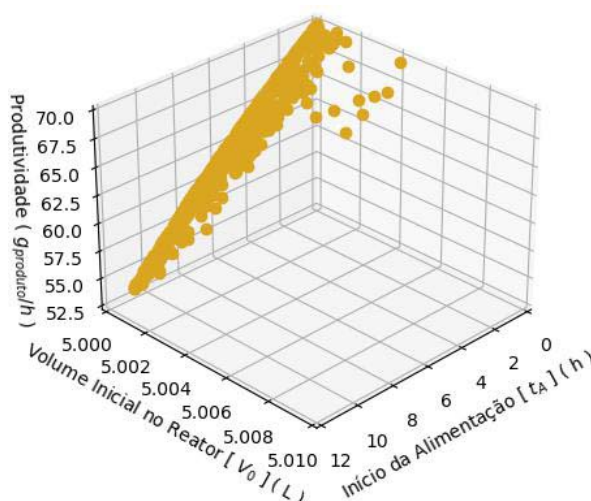
Gráfico 9 - Produtividade correspondente aos otimizadores de Pareto do Problema A do Estudo de Caso 1 obtidos pelo AGE-MOEA.



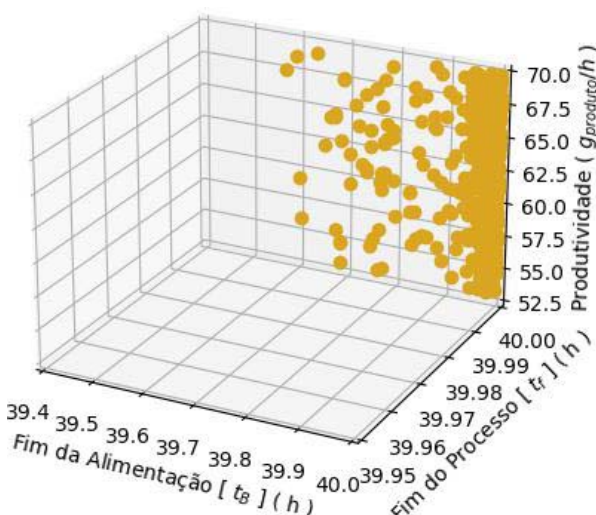
(a) Produtividade ($Prod$) \times I. Alim. (t_A) \times F. Alim. (t_B).



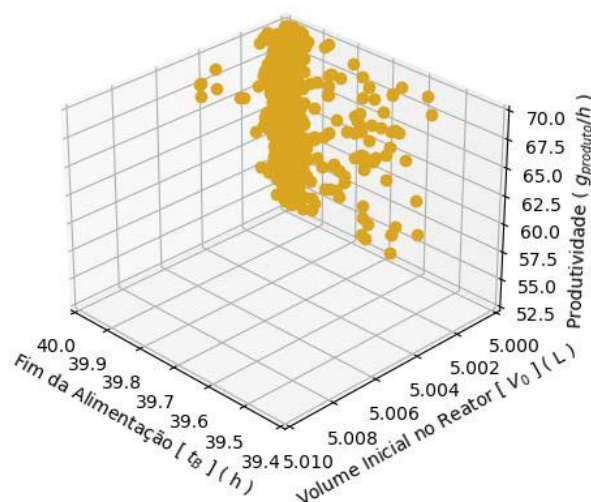
(b) Produtividade ($Prod$) \times I. Alim. (t_A) \times F. Proc. (t_f).



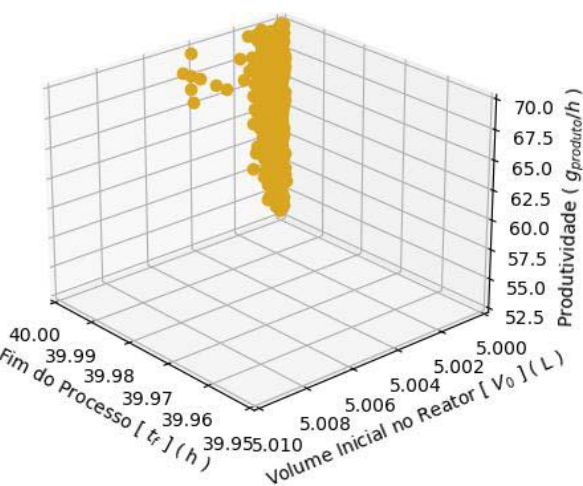
(c) Produtividade ($Prod$) \times I. Alim. (t_A) \times V. Ini. Reat. (V_0).



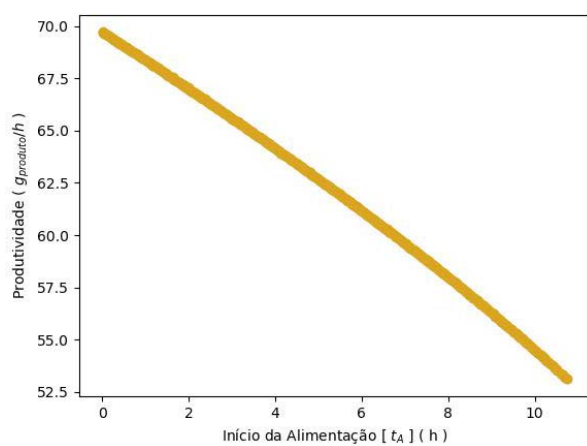
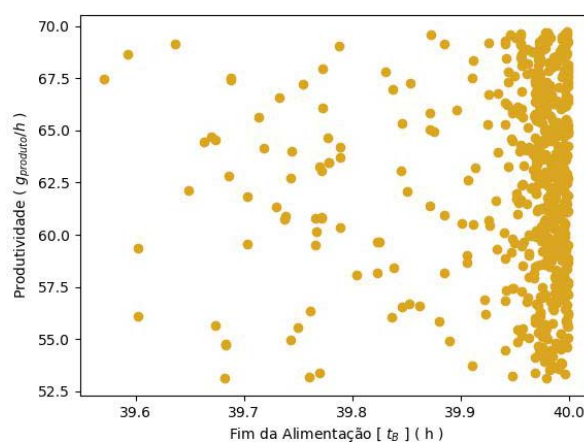
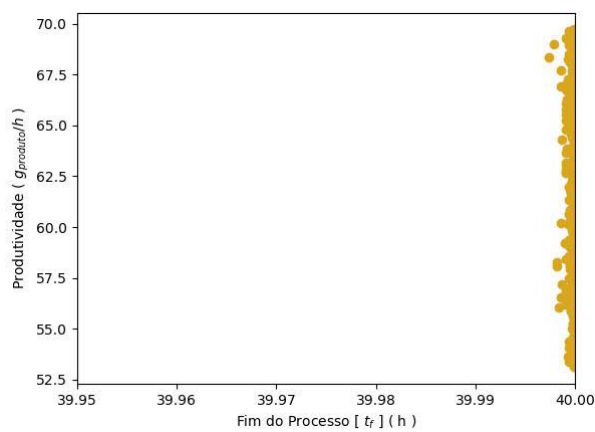
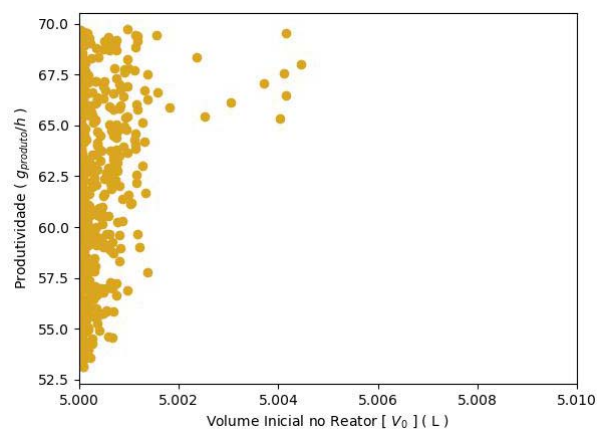
(d) Produtividade ($Prod$) \times F. Alim. (t_B) \times F. Proc. (t_f).



(e) Produtividade ($Prod$) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).

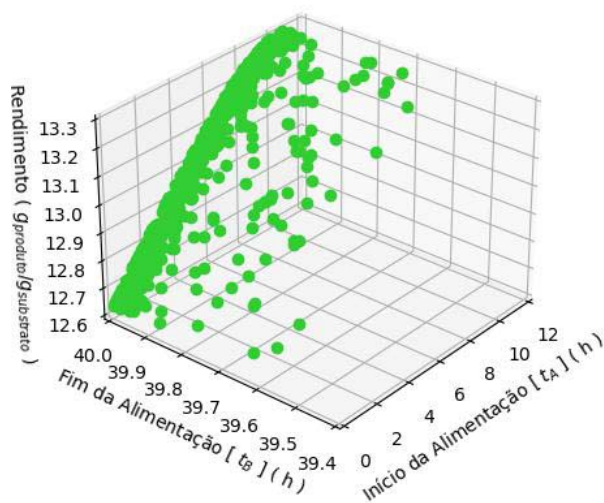


(f) Produtividade ($Prod$) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).

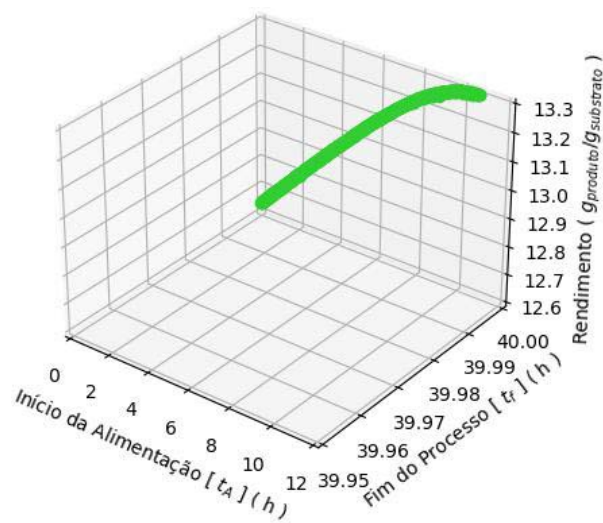
(g) Produtividade ($Prod$) \times I. Alim. (t_A).(h) Produtividade ($Prod$) \times F. Alim. (t_B).(i) Produtividade ($Prod$) \times F. Proc. (t_f).(j) Produtividade ($Prod$) \times V. Ini. Reat. (V_0).

Fonte: O autor, 2023.

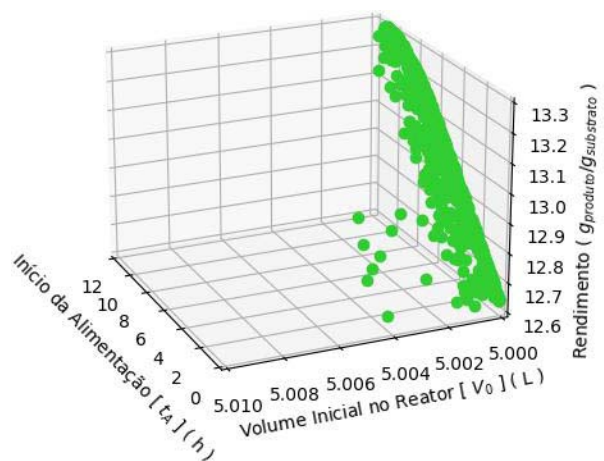
Gráfico 10 - Rendimento correspondente aos otimizadores de Pareto do Problema A do Estudo de Caso 1 obtidos pelo AGE-MOEA.



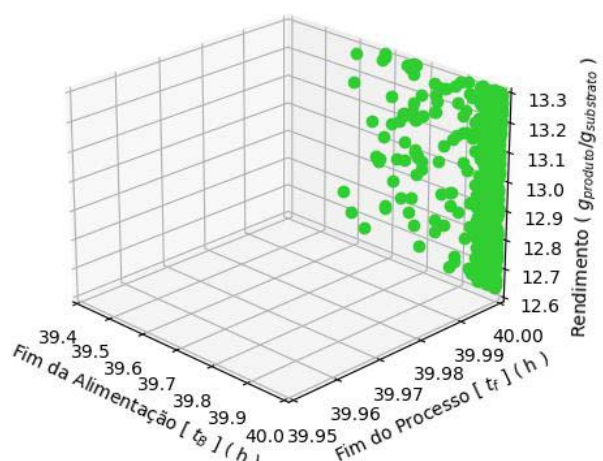
(a) Rendimento ($Rend$) \times I. Alim. (t_A) \times F. Alim. (t_B).



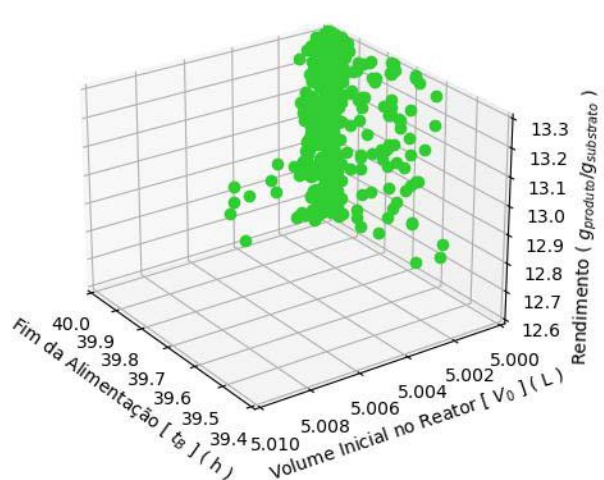
(b) Rendimento ($Rend$) \times I. Alim. (t_A) \times F. Proc. (t_f).



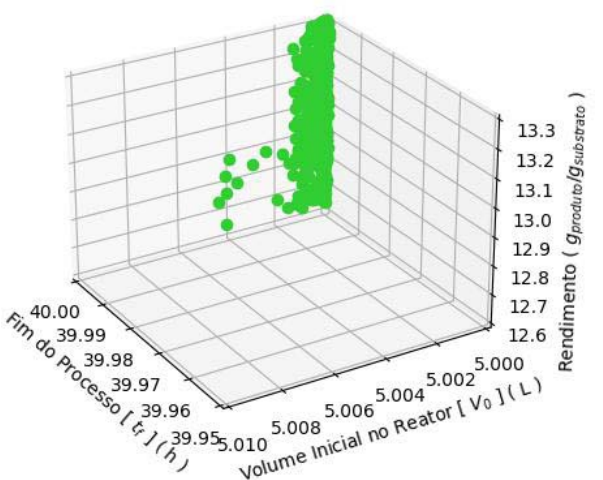
(c) Rendimento ($Rend$) \times I. Alim. (t_A) \times V. Ini. Reat. (V_0).



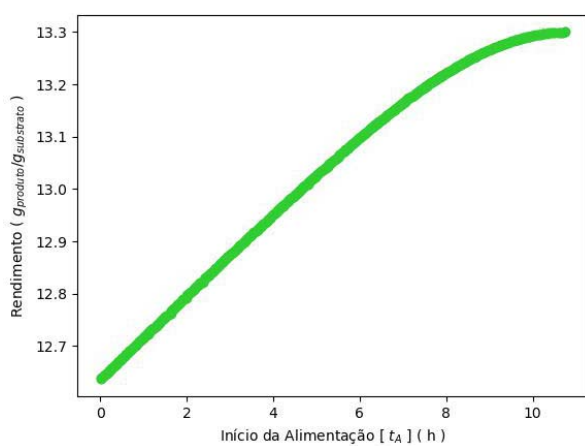
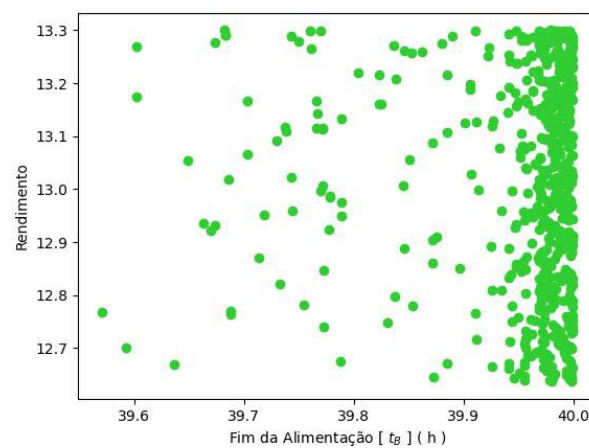
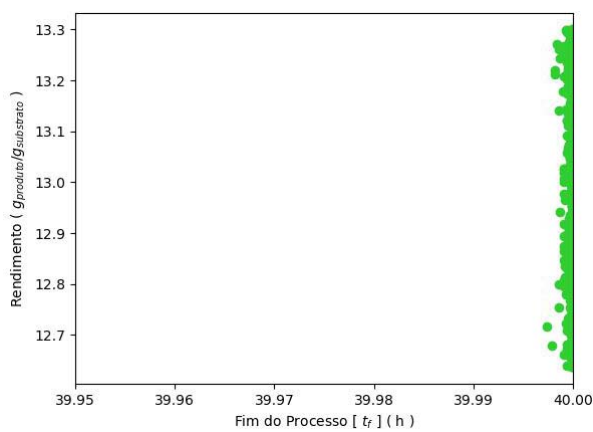
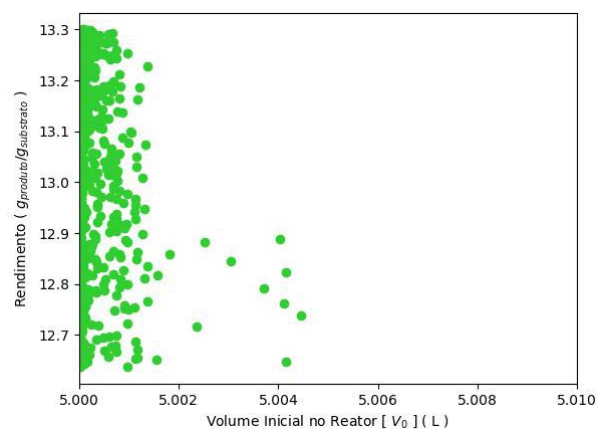
(d) Rendimento ($Rend$) \times F. Alim. (t_B) \times F. Proc. (t_f).



(e) Rendimento ($Rend$) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).



(f) Rendimento ($Rend$) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).

(g) Rendimento ($Rend$) \times I. Alim. (t_A).(h) Rendimento ($Rend$) \times F. Alim. (t_B).(i) Rendimento ($Rend$) \times F. Proc. (t_f).(j) Rendimento ($Rend$) \times V. Ini. Reat. (V_0).

Fonte: O autor, 2023.

8.2.1.3 Simulação com o C-TAEA

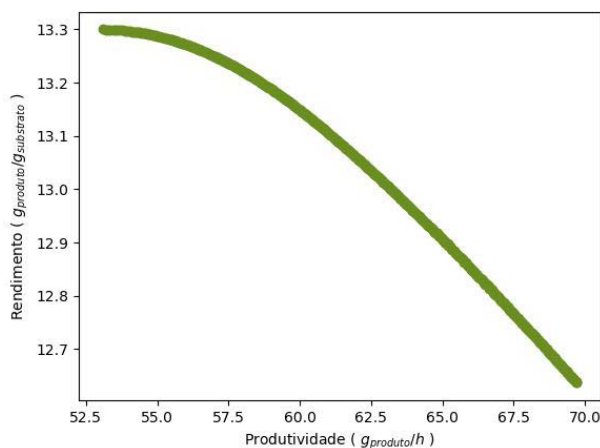
A simulação que buscou solucionar o Problema de Otimização A através do método C-TAEA 9 minutos e 34 segundos para atingir as 50 gerações. O método de otimização convergiu na 23^a geração, encontrando 600 soluções não dominadas.

As soluções obtidas nesta simulação estão descritas, detalhadas e ilustradas nas subseções abaixo.

8.2.1.3.1 Fronteira de Pareto

O Gráfico 11 apresenta a frente de Pareto do problema de otimização em questão, que foi estimada nesta simulação.

Gráfico 11 - Fronteira de Pareto do Problema A do Estudo de Caso 1 obtida pelo C-TAEA.

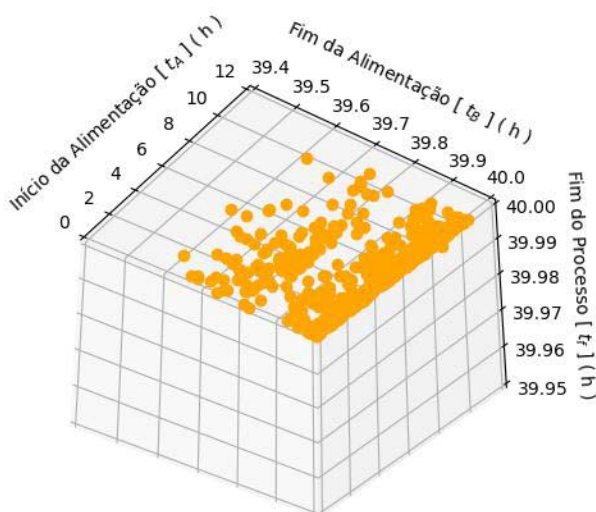


Fonte: O autor, 2023.

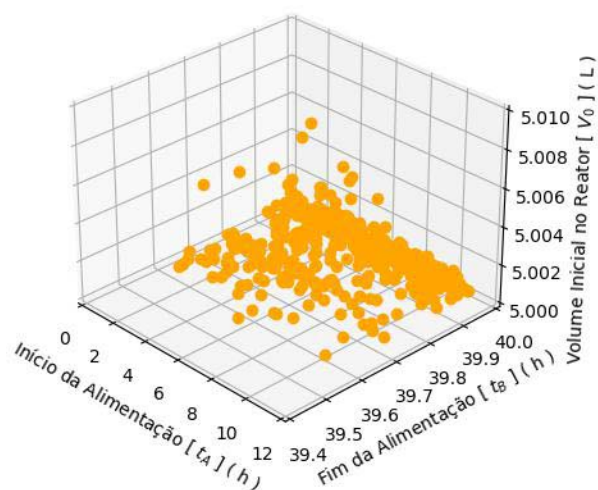
8.2.1.3.2 Conjunto Ótimo de Pareto

O Gráfico 12 mostra os otimizadores de Pareto encontrados nesta simulação de otimização, nos quais consiste o conjunto ótimo de Pareto estimado.

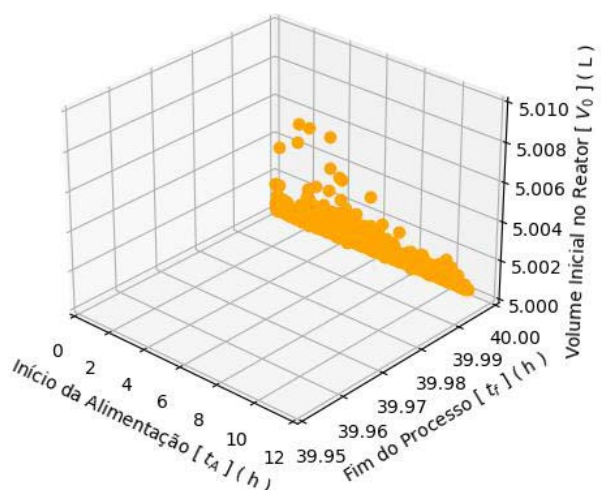
Gráfico 12 - Conjunto ótimo de Pareto do Problema A do Estudo de Caso 1 obtido pelo C-TAEA.



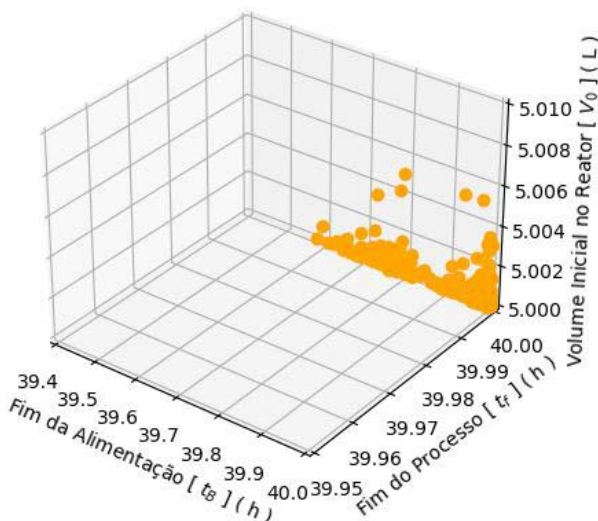
(a) I. Alim. (t_A) \times F. Alim. (t_B) \times F. Proc. (t_f).



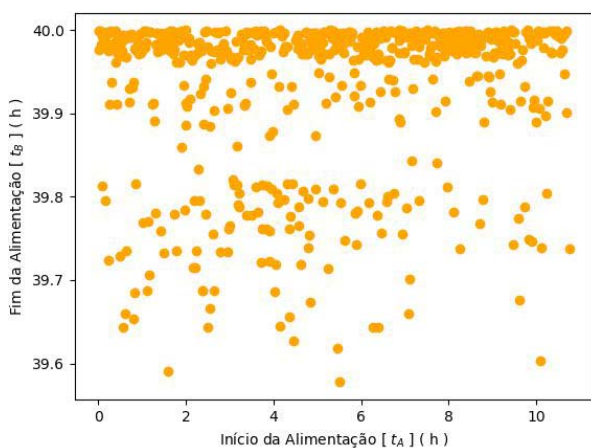
(b) I. Alim. (t_A) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).



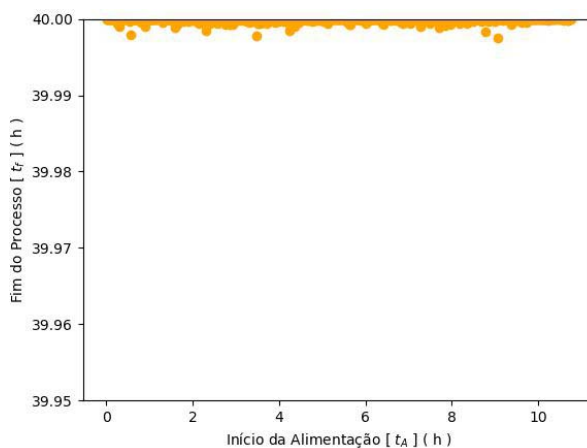
(c) I. Alim. (t_A) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).



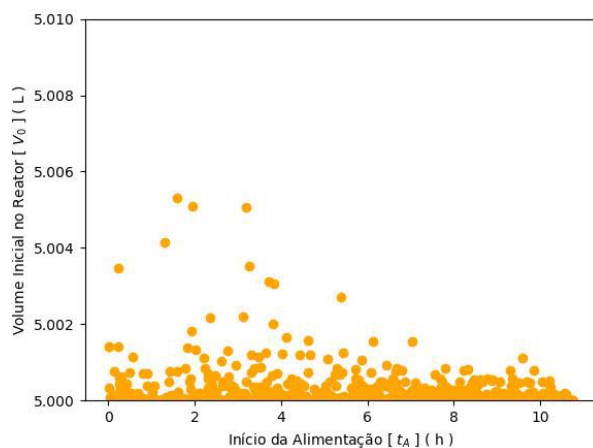
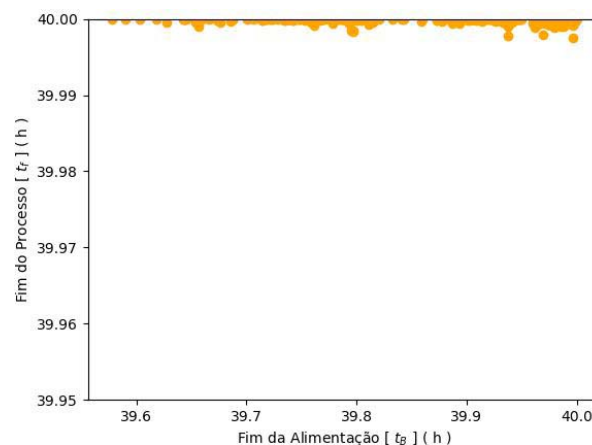
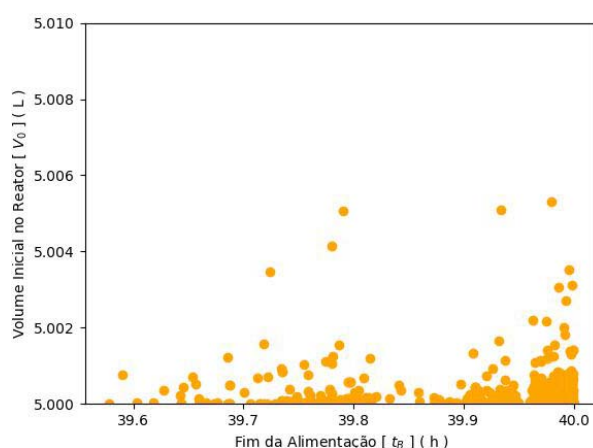
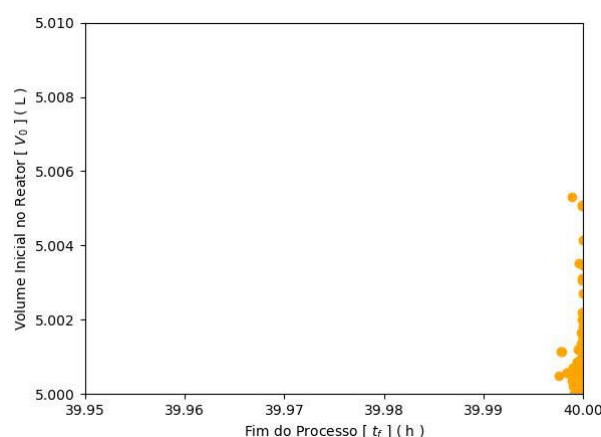
(d) F. Alim. (t_B) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).



(e) I. Alim. (t_A) \times F. Alim. (t_B).



(f) I. Alim. (t_A) \times F. Proc. (t_f).

(g) I. Alim. (t_A) \times V. Ini. Reat. (V_0).(h) F. Alim. (t_B) \times F. Proc. (t_f).(i) F. Alim. (t_B) \times V. Ini. Reat. (V_0).(j) F. Proc. (t_f) \times V. Ini. Reat. (V_0).

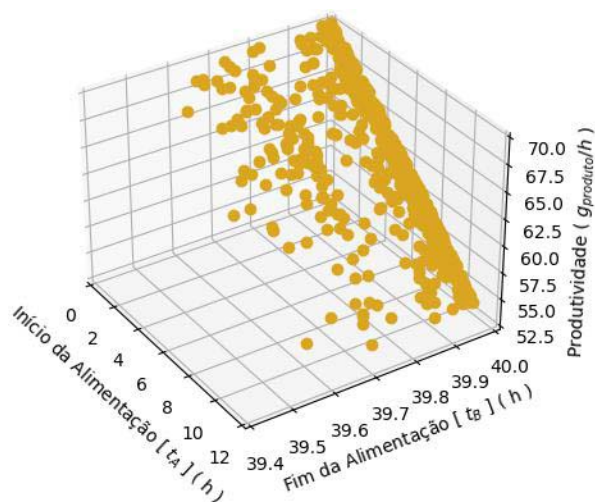
Fonte: O autor, 2023.

8.2.1.3.3 Relação entre os Otimizadores de Pareto e os Objetivos

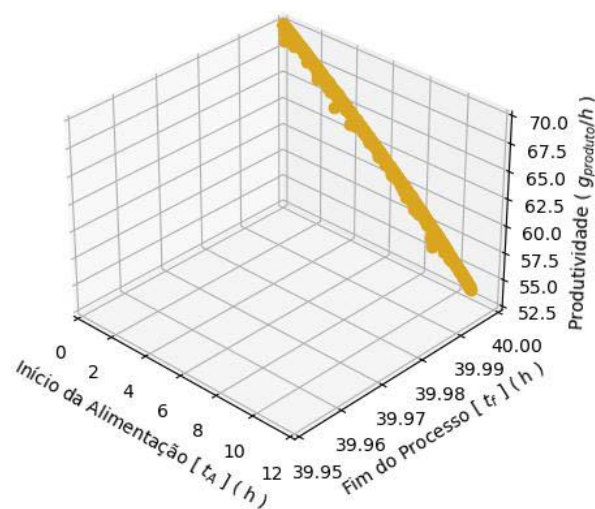
A seguir, serão apresentadas as correspondências dos pontos do conjunto ótimo de Pareto, mostrado no Gráfico 12, com cada um dos objetivos do problema de otimização em questão.

O Gráfico 13 mostra a produtividade relacionada a cada um dos otimizadores do conjunto ótimo de Pareto e o Gráfico 14 enuncia o rendimento relacionado a cada um dos otimizadores de Pareto encontrados.

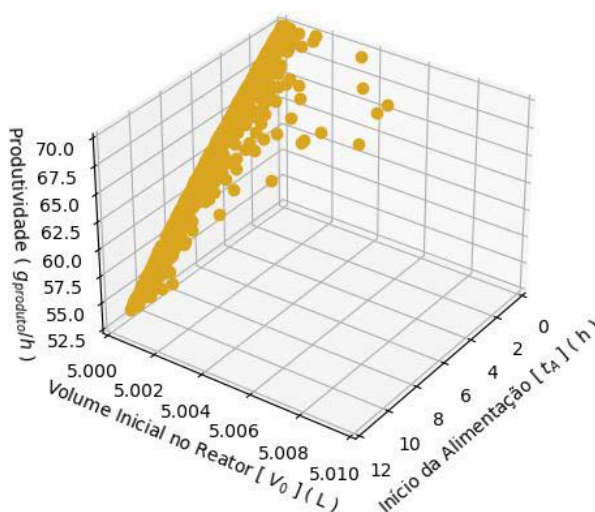
Gráfico 13 - Produtividade correspondente aos otimizadores de Pareto do Problema A do Estudo de Caso 1 obtidos pelo C-TAEA.



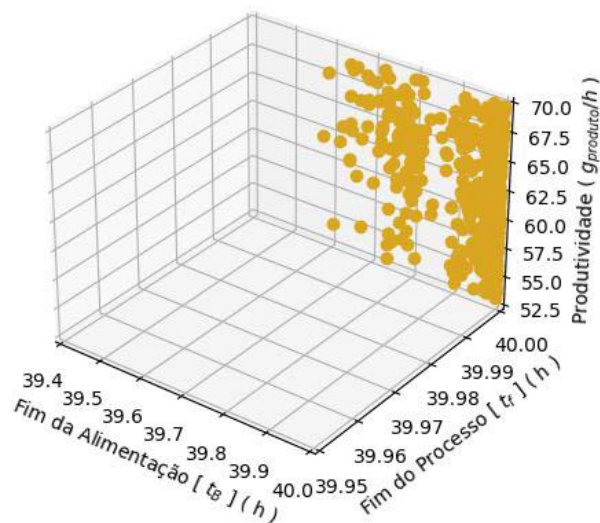
(a) Produtividade ($Prod$) \times I. Alim. (t_A) \times F. Alim. (t_B).



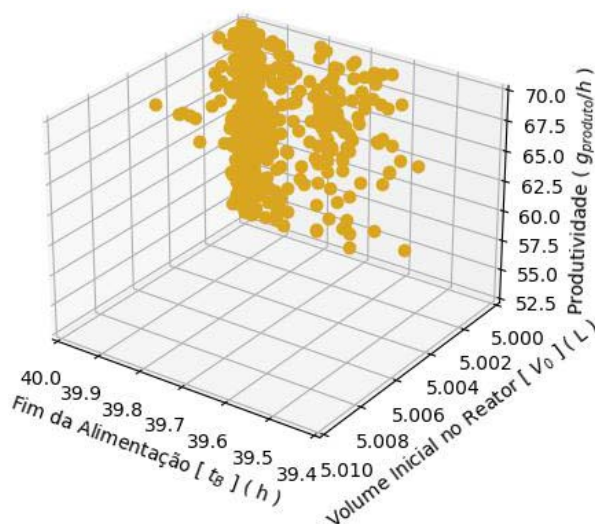
(b) Produtividade ($Prod$) \times I. Alim. (t_A) \times F. Proc. (t_f).



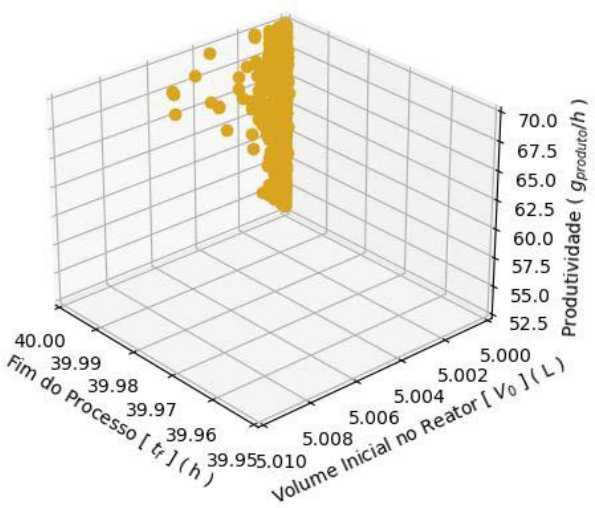
(c) Produtividade ($Prod$) \times I. Alim. (t_A) \times V. Ini. Reat. (V_0).



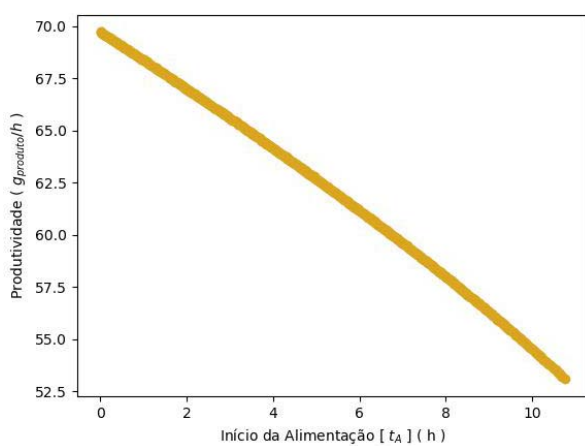
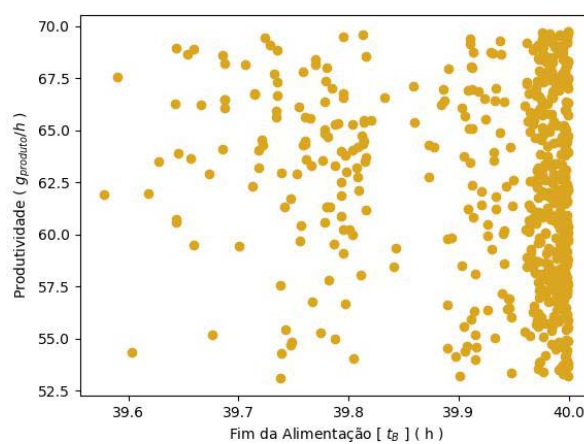
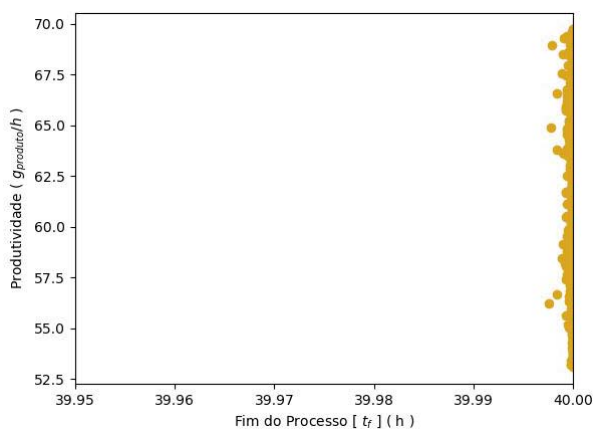
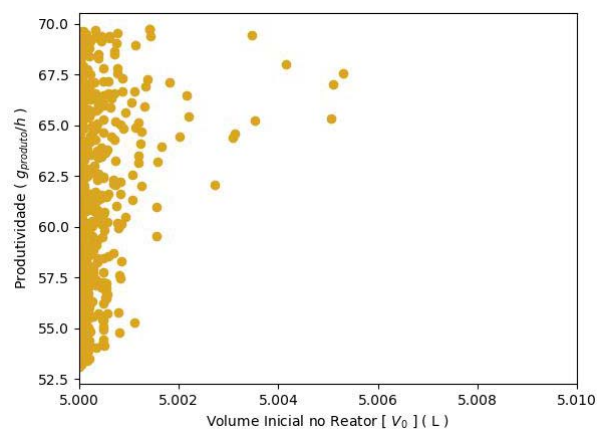
(d) Produtividade ($Prod$) \times F. Alim. (t_B) \times F. Proc. (t_f).



(e) Produtividade ($Prod$) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).

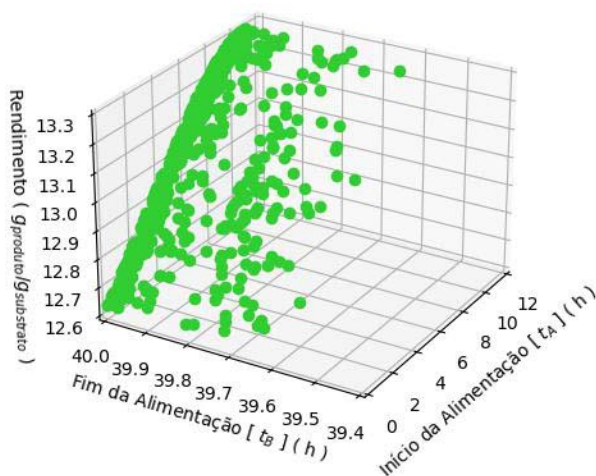


(f) Produtividade ($Prod$) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).

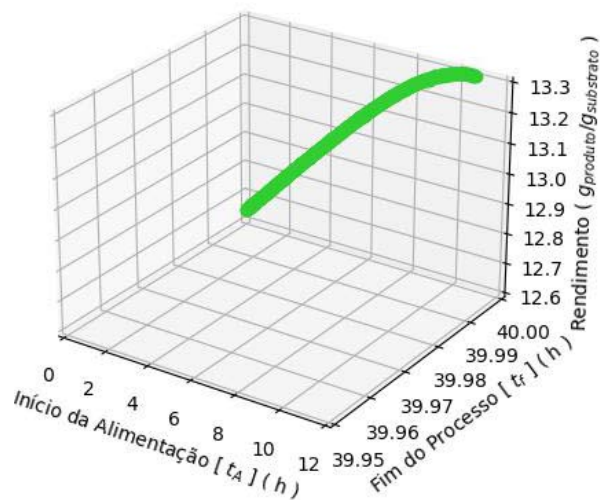
(g) Produtividade ($Prod$) \times I. Alim. (t_A).(h) Produtividade ($Prod$) \times F. Alim. (t_B).(i) Produtividade ($Prod$) \times F. Proc. (t_f).(j) Produtividade ($Prod$) \times V. Ini. Reat. (V_0).

Fonte: O autor, 2023.

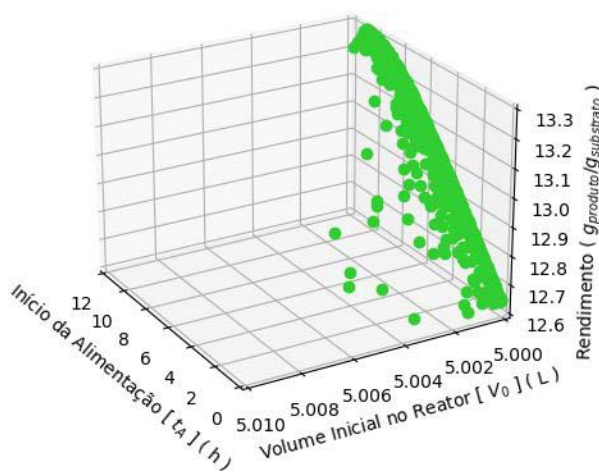
Gráfico 14 - Rendimento correspondente aos otimizadores de Pareto do Problema A do Estudo de Caso 1 obtidos pelo C-TAEA.



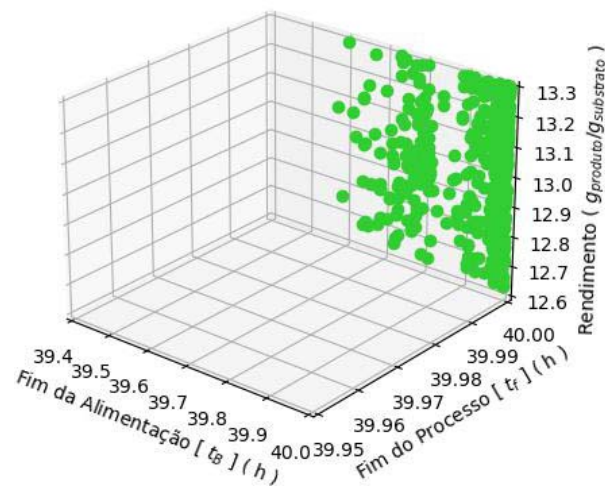
(a) Rendimento ($Rend$) \times I. Alim. (t_A) \times F. Alim. (t_B).



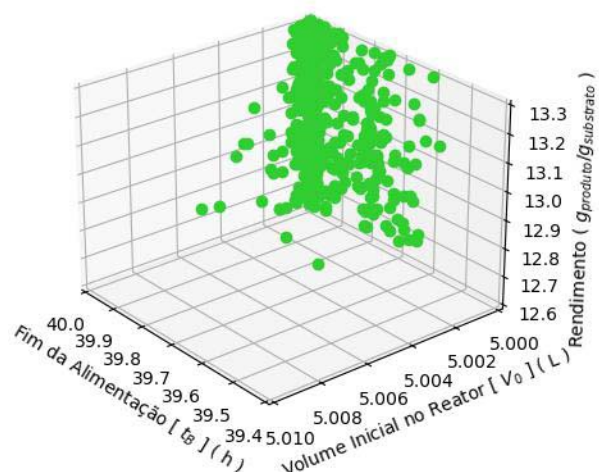
(b) Rendimento ($Rend$) \times I. Alim. (t_A) \times F. Proc. (t_f).



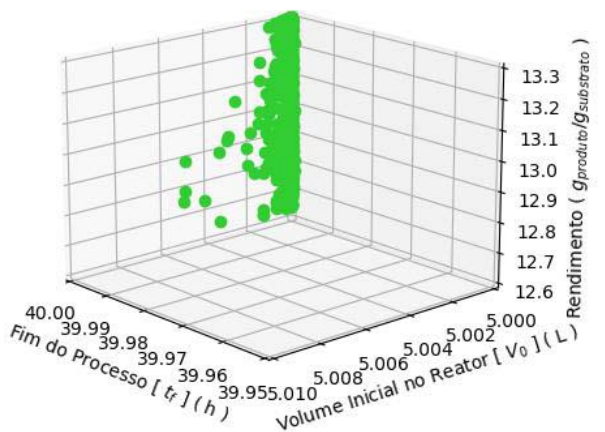
(c) Rendimento ($Rend$) \times I. Alim. (t_A) \times V. Ini. Reat. (V_0).



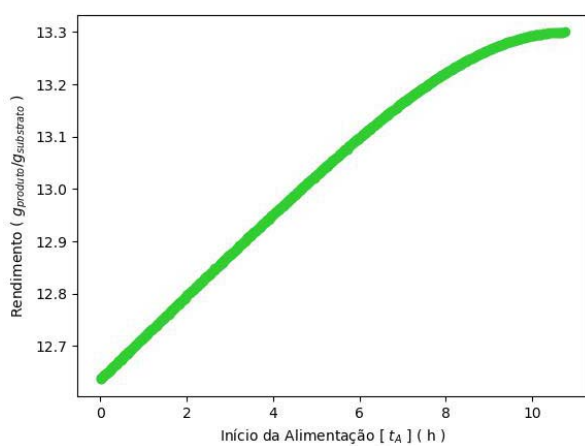
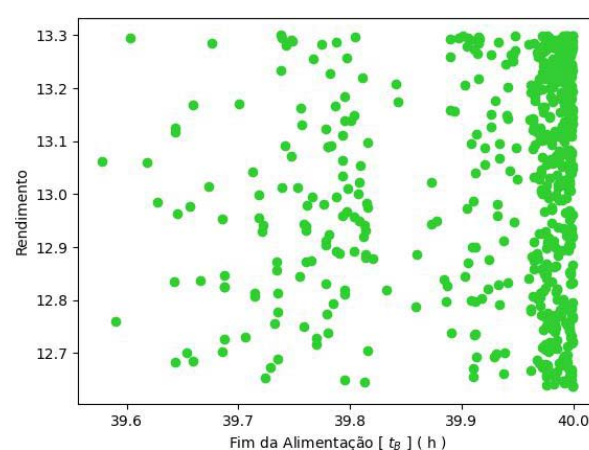
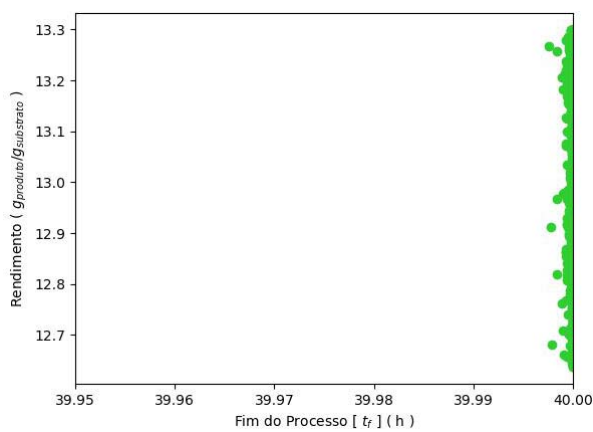
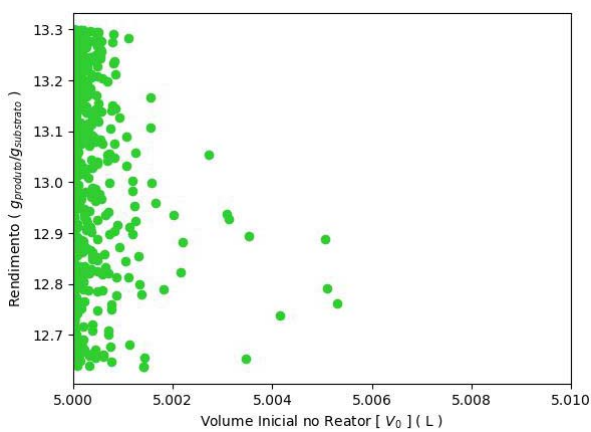
(d) Rendimento ($Rend$) \times F. Alim. (t_B) \times F. Proc. (t_f).



(e) Rendimento ($Rend$) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).



(f) Rendimento ($Rend$) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).

(g) Rendimento ($Rend$) \times I. Alim. (t_A).(h) Rendimento ($Rend$) \times F. Alim. (t_B).(i) Rendimento ($Rend$) \times F. Proc. (t_f).(j) Rendimento ($Rend$) \times V. Ini. Reat. (V_0).

Fonte: O autor, 2023.

8.2.1.4 Soluções Notáveis

Considerando os ótimos de Pareto obtidos em todas as simulações de otimização realizadas sobre este problema, foram elencadas as soluções que apresentam o melhor valor atingido para cada um dos objetivos, bem como, soluções que equilibram ambos os objetivos.

A Tabela 10 expressa cada uma das soluções elencadas, bem como o método usado em sua obtenção.

Tabela 10 - Soluções Notáveis do Problema A do Estudo de Caso 1.

Solução	Variáveis				Objetivos	
	t_A	t_B	t_f	V_0	$Prod(t_f)$	$Rend(t_f)$
AGE-MOEA						
E1P1M1S1	0,00333	39,99805	39,99990	5,00096	69,71117	12,63660
E1P1M1S2	10,76103	39,97957	39,99997	5,00008	53,11779	13,30011
E1P1M1S3	4,24701	39,99879	39,99990	5,00073	63,81185	12,96810
C-TAEA						
E1P1M2S1	0,00410	39,99967	39,99999	5,00140	69,71372	12,63657
E1P1M2S2	10,77509	39,73774	39,99996	5,00000	53,07905	13,30017
E1P1M2S3	4,23840	39,79695	39,99839	5,00056	63,81496	12,96778
NSGA-II						
E1P1M3S1	0,00354	39,99401	39,99971	5,00121	69,71244	12,63656
E1P1M3S2	10,85661	39,99667	39,99995	5,00007	52,91689	13,30009
E1P1M3S3	4,23409	39,94422	39,99973	5,00041	63,81918	12,96777

Fonte: O autor, 2023.

8.2.2 Problema de Otimização B

Nesta subseção, serão apresentadas as informações específicas e os resultados obtidos a partir de cada uma das simulações de otimização feitas sobre o Problema de Otimização B do Estudo de Caso 1.

8.2.2.1 Simulação com o NSGA-II

A simulação que buscou solucionar o Problema de Otimização B através do método NSGA-II levou 7 minutos e 12 segundos para atingir as 50 gerações. O método de otimização convergiu na 31ª geração, encontrando 600 soluções não dominadas.

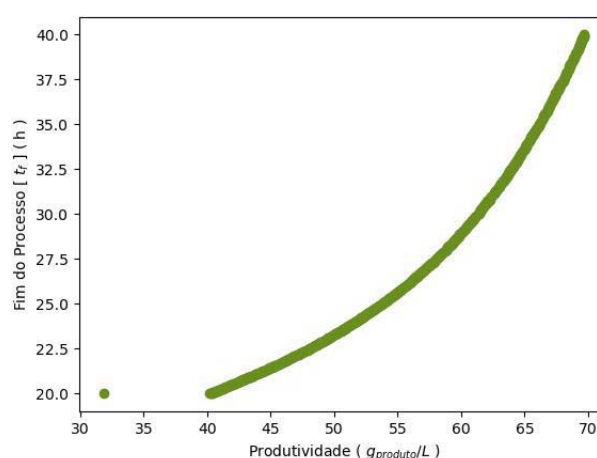
As soluções obtidas nesta simulação estão descritas, detalhadas e ilustradas nas

subseções abaixo.

8.2.2.1.1 Fronteira de Pareto

O Gráfico 15 apresenta a frente de Pareto do problema de otimização em questão, que foi estimada nesta simulação.

Gráfico 15 - Fronteira de Pareto do Problema B do Estudo de Caso 1 obtida pelo NSGA-II.



Fonte: O autor, 2023.

8.2.2.1.2 Conjunto Ótimo de Pareto

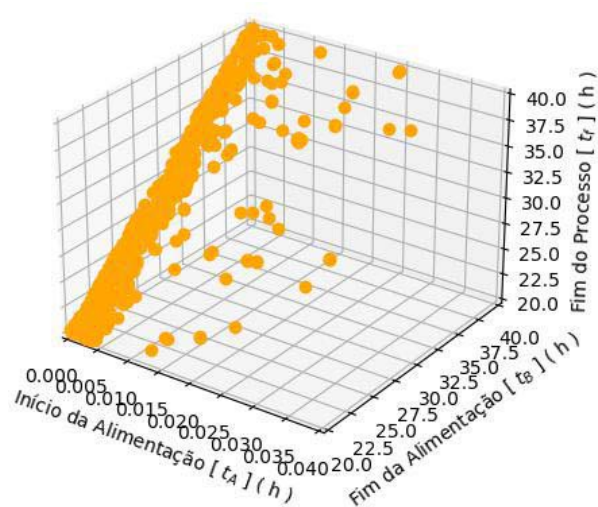
O Gráfico 16 mostra os otimizadores de Pareto encontrados nesta simulação de otimização, nos quais consiste o conjunto ótimo de Pareto estimado.

8.2.2.1.3 Relação entre os Otimizadores de Pareto e os Objetivos

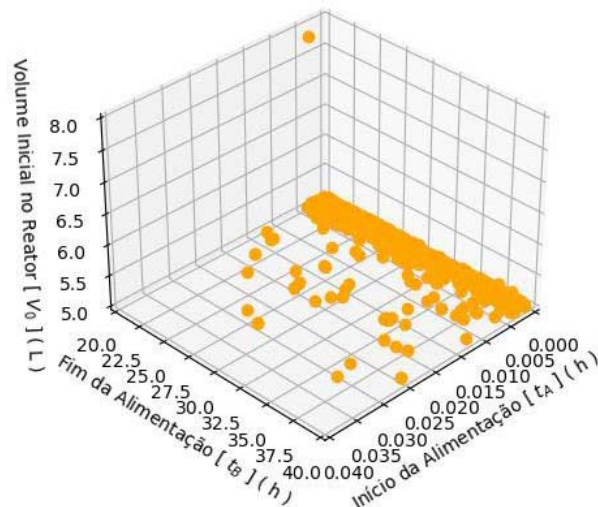
A seguir, serão apresentadas as correspondências dos pontos do conjunto ótimo de Pareto, mostrado no Gráfico 16, com cada um dos objetivos do problema de otimização em questão.

O Gráfico 17 mostra a produtividade relacionada a cada um dos otimizadores do conjunto ótimo de Pareto e o Gráfico 18 enuncia o tempo final relacionado a cada um dos otimizadores de Pareto encontrados.

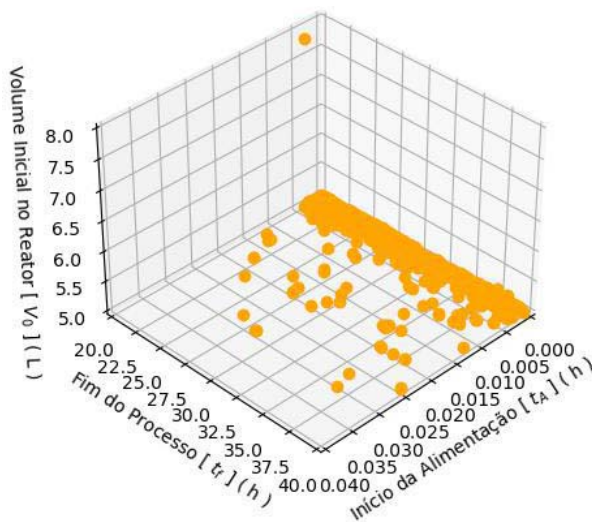
Gráfico 16 - Conjunto ótimo de Pareto do Problema B do Estudo de Caso 1 obtido pelo NSGA-II.



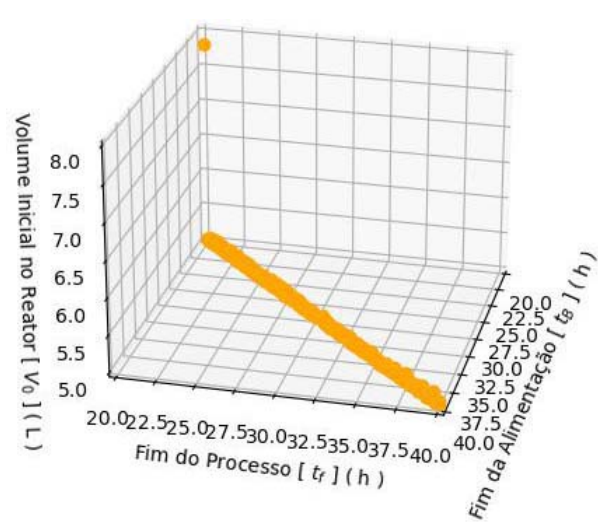
(a) I. Alim. (t_A) \times F. Alim. (t_B) \times F. Proc. (t_f).



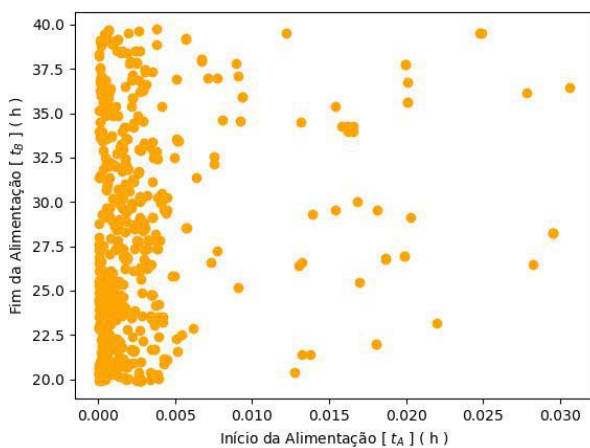
(b) I. Alim. (t_A) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).



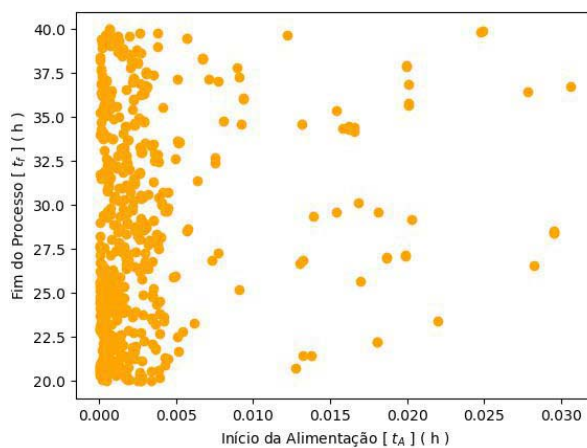
(c) I. Alim. (t_A) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).



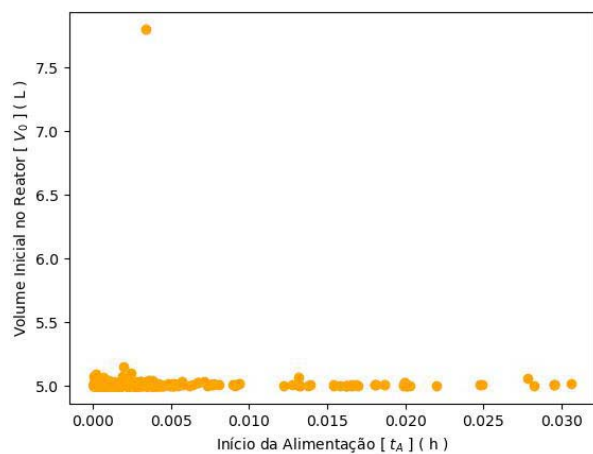
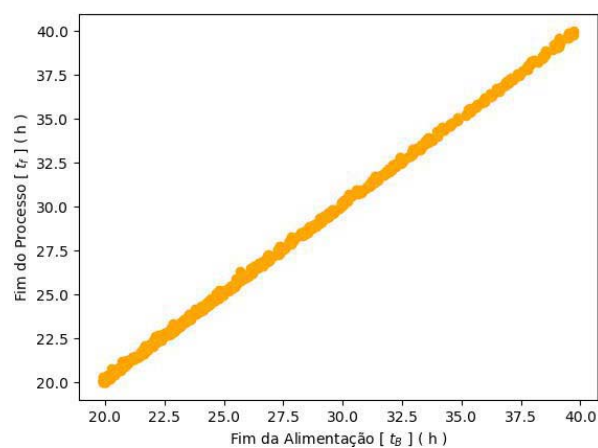
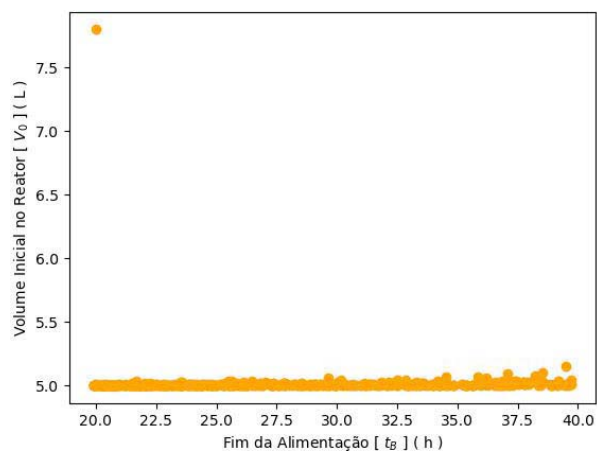
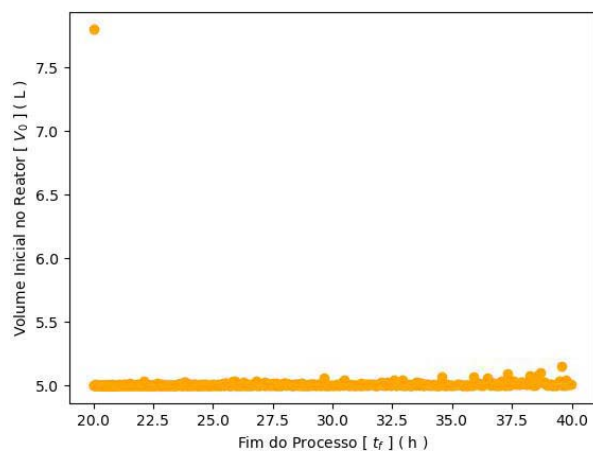
(d) F. Alim. (t_B) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).



(e) I. Alim. (t_A) \times F. Alim. (t_B).

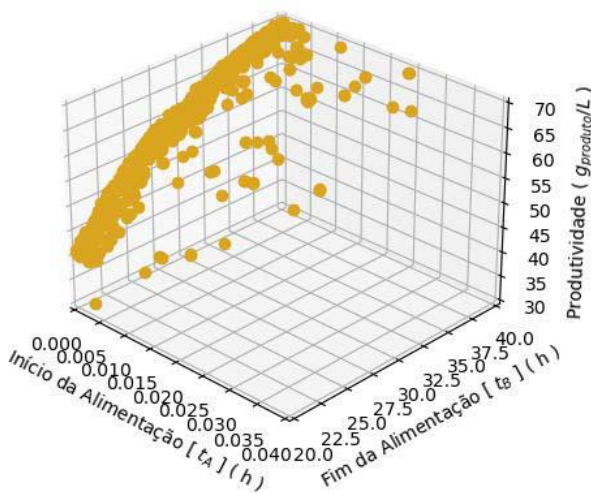


(f) I. Alim. (t_A) \times F. Proc. (t_f).

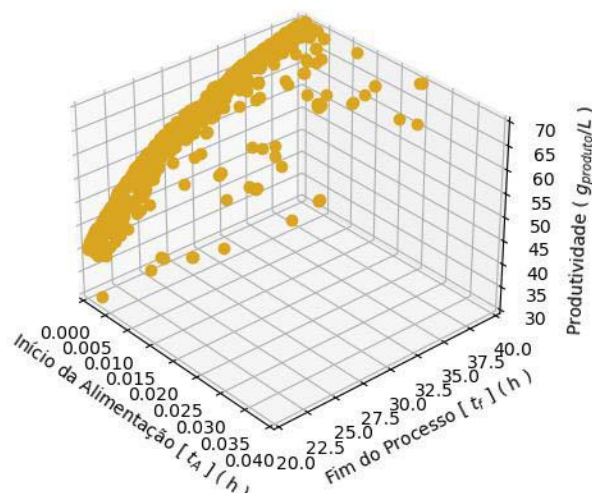
(g) I. Alim. (t_A) \times V. Ini. Reat. (V_0).(h) F. Alim. (t_B) \times F. Proc. (t_f).(i) F. Alim. (t_B) \times V. Ini. Reat. (V_0).(j) F. Proc. (t_f) \times V. Ini. Reat. (V_0).

Fonte: O autor, 2023.

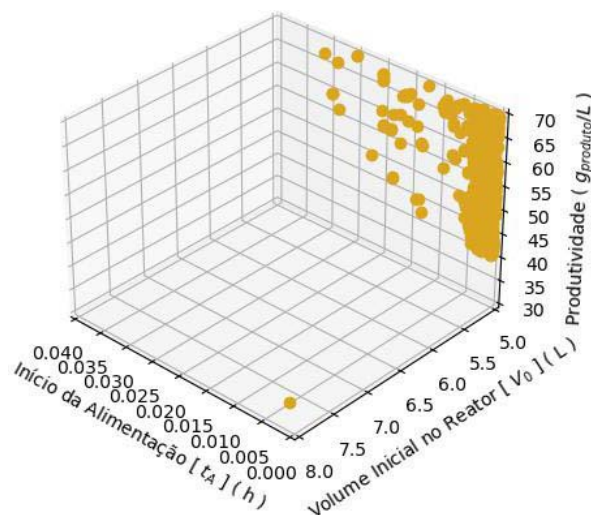
Gráfico 17 - Produtividade correspondente aos otimizadores de Pareto do Problema B do Estudo de Caso 1 obtidos pelo NSGA-II.



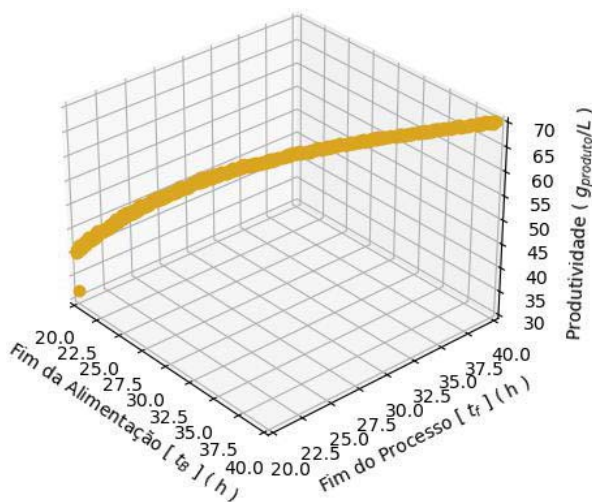
(a) Produtividade ($Prod$) \times I. Alim. (t_A) \times F. Alim. (t_B).



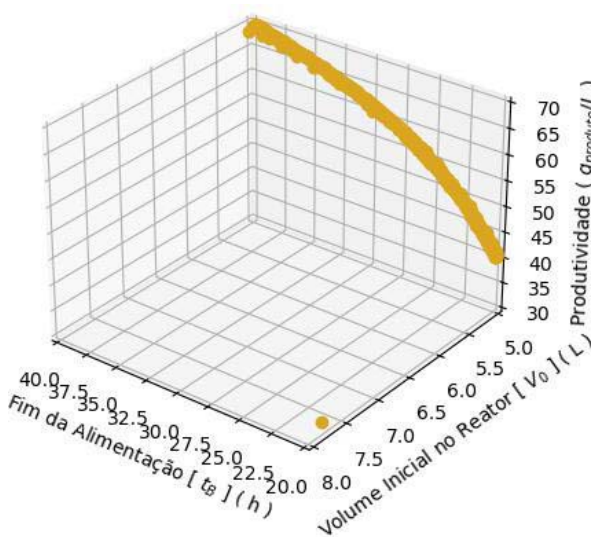
(b) Produtividade ($Prod$) \times I. Alim. (t_A) \times F. Proc. (t_f).



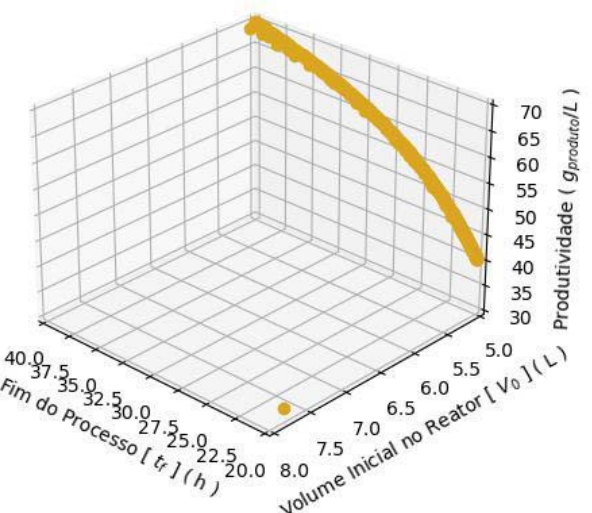
(c) Produtividade ($Prod$) \times I. Alim. (t_A) \times V. Ini. Reat. (V_0).



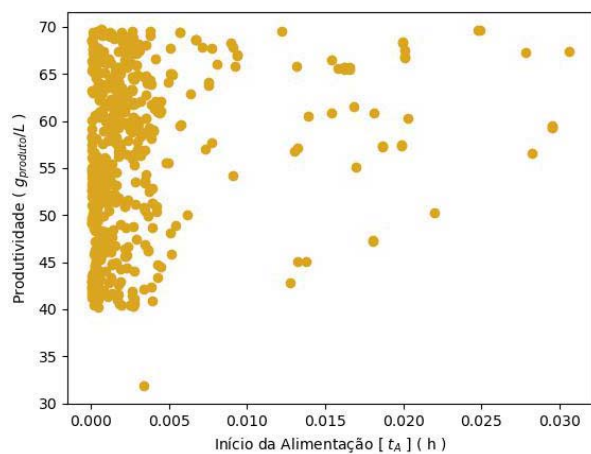
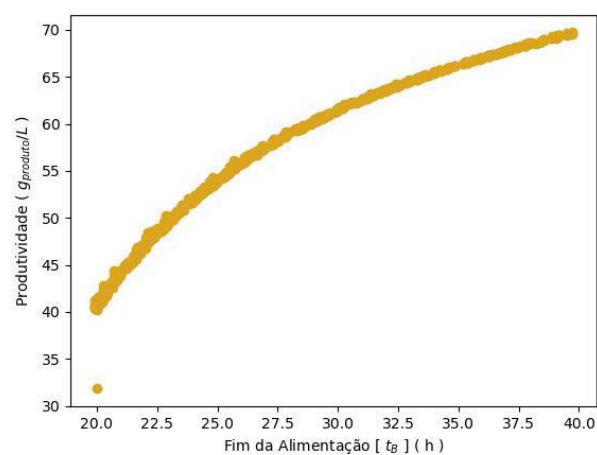
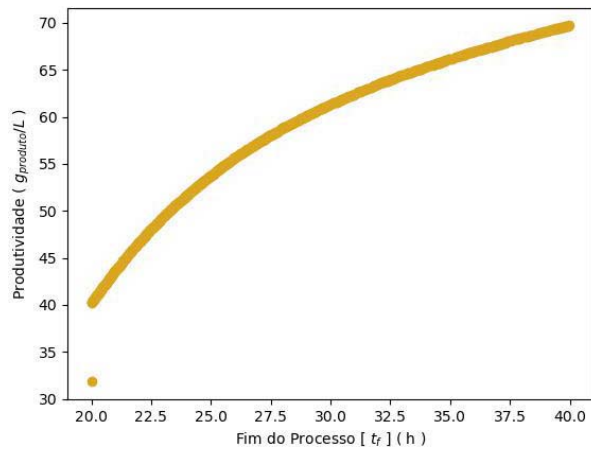
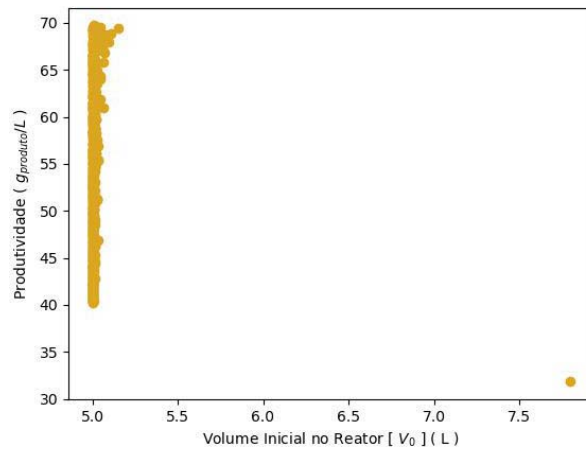
(d) Produtividade ($Prod$) \times F. Alim. (t_B) \times F. Proc. (t_f).



(e) Produtividade ($Prod$) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).

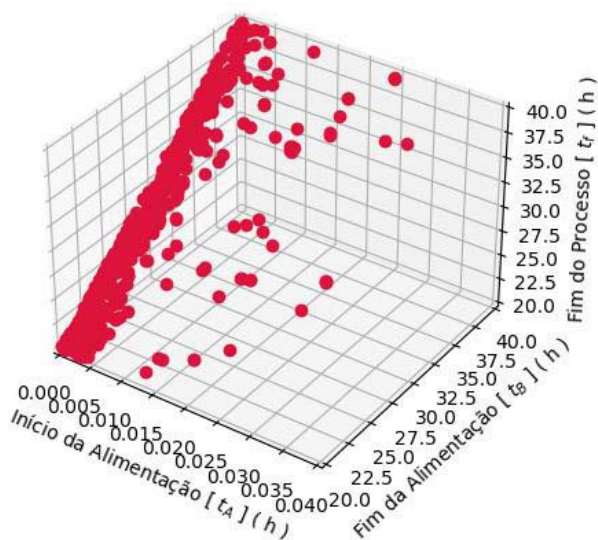


(f) Produtividade ($Prod$) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).

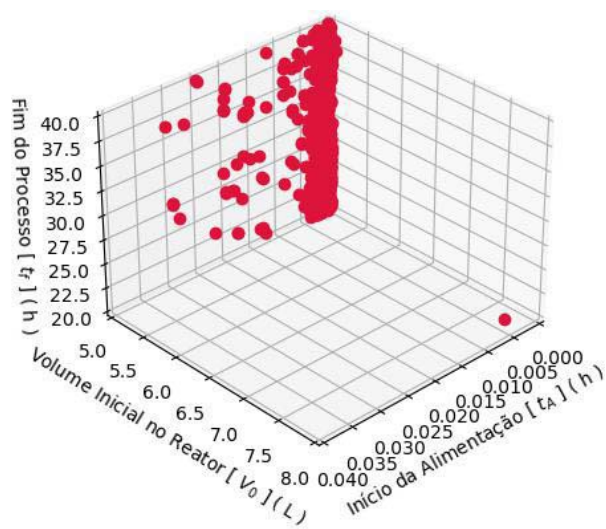
(g) Produtividade ($Prod$) \times I. Alim. (t_A).(h) Produtividade ($Prod$) \times F. Alim. (t_B).(i) Produtividade ($Prod$) \times F. Proc. (t_f).(j) Produtividade ($Prod$) \times V. Ini. Reat. (V_0).

Fonte: O autor, 2023.

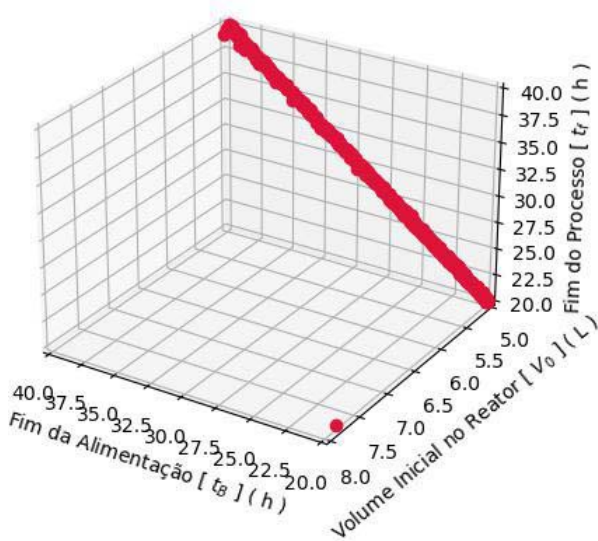
Gráfico 18 - Tempo final correspondente aos otimizadores de Pareto do Problema B do Estudo de Caso 1 obtidos pelo NSGA-II.



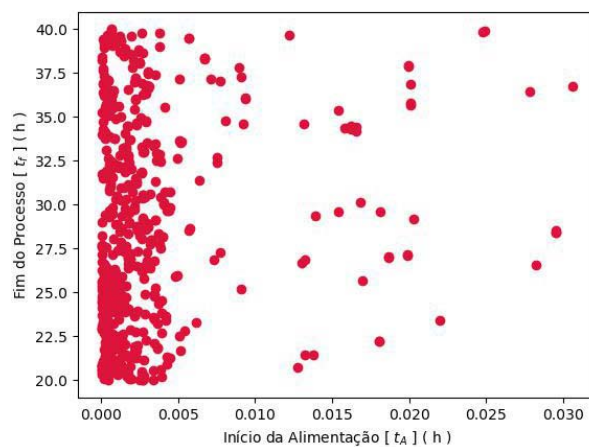
(a) Fim do Proc. (t_f) \times I. Alim. (t_A) \times F. Alim. (t_B).



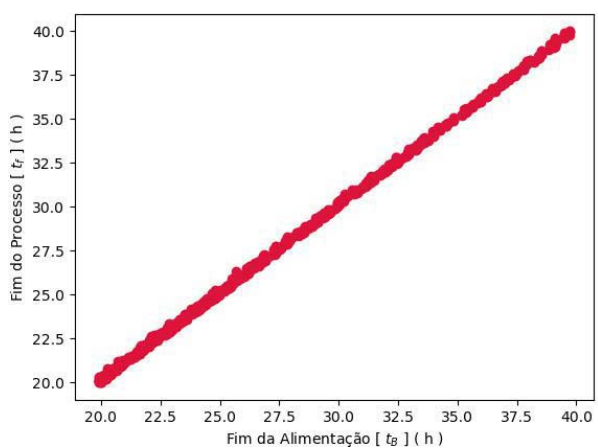
(b) Fim do Proc. (t_f) \times I. Alim. (t_A) \times V. Ini. Reat. (V_0).



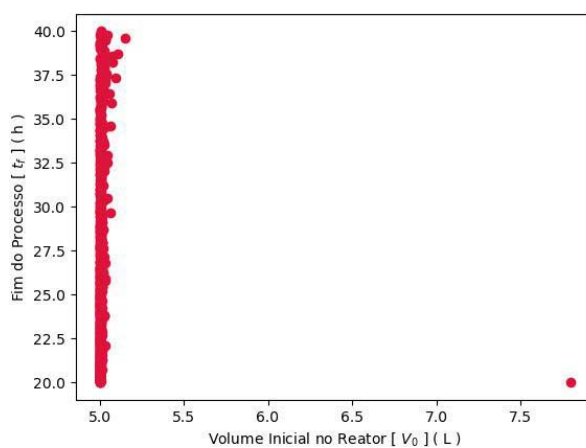
(c) Fim do Proc. (t_f) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).



(d) Fim do Proc. (t_f) \times I. Alim. (t_A).



(e) Fim do Proc. (t_f) \times F. Alim. (t_B).



(f) Fim do Proc. (t_f) \times V. Ini. Reat. (V_0).

8.2.2.2 Simulação com o AGE-MOEA

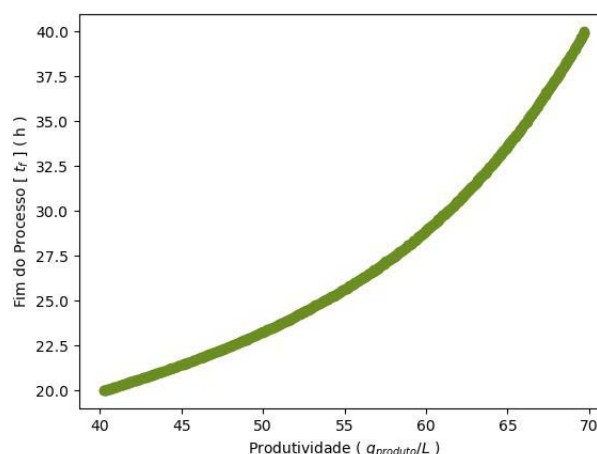
A simulação que buscou solucionar o Problema de Otimização B através do método AGE-MOEA levou 8 minutos e 58 segundos para atingir as 50 gerações. O método de otimização convergiu na 31^a geração, encontrando 600 soluções não dominadas.

As soluções obtidas nesta simulação estão descritas, detalhadas e ilustradas nas subseções abaixo.

8.2.2.2.1 Fronteira de Pareto

O Gráfico 19 apresenta a frente de Pareto do problema de otimização em questão, que foi estimada nesta simulação.

Gráfico 19 - Fronteira de Pareto do Problema B do Estudo de Caso 1 obtida pelo AGE-MOEA.

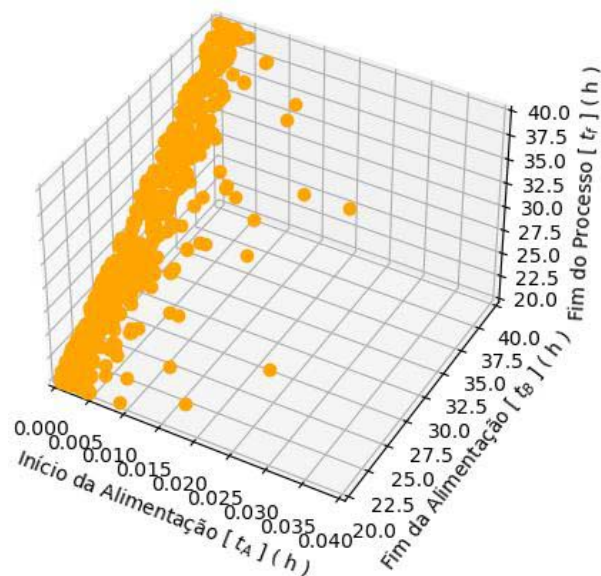


Fonte: O autor, 2023.

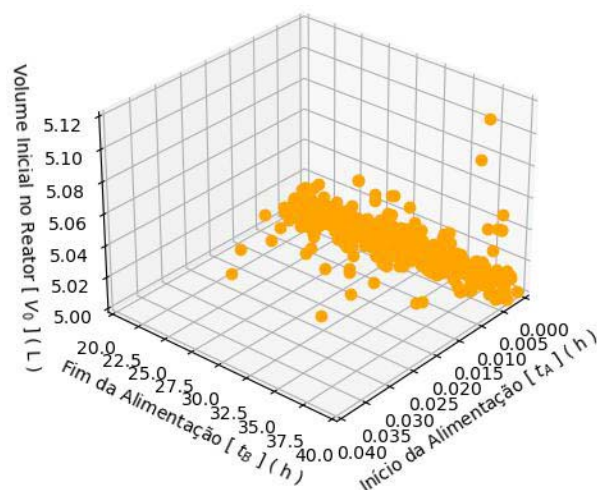
8.2.2.2.2 Conjunto Ótimo de Pareto

O Gráfico 20 mostra os otimizadores de Pareto encontrados nesta simulação de otimização, nos quais consiste o conjunto ótimo de Pareto estimado.

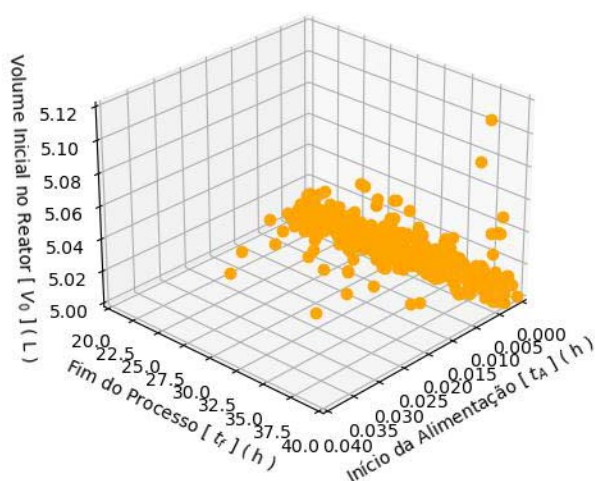
Gráfico 20 - Conjunto ótimo de Pareto do Problema B do Estudo de Caso 1 obtido pelo AGE-MOEA.



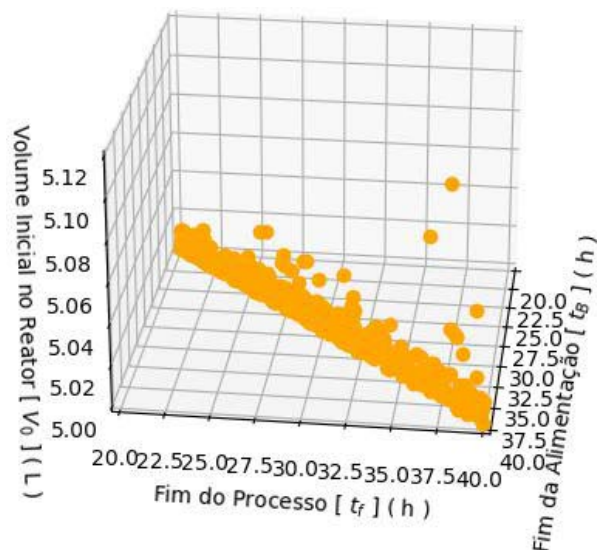
(a) I. Alim. (t_A) \times F. Alim. (t_B) \times F. Proc. (t_f).



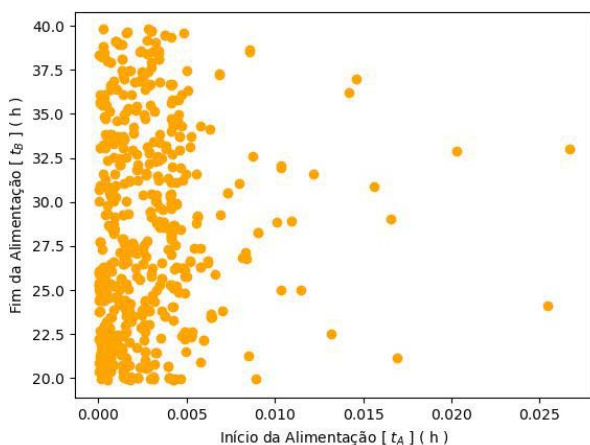
(b) I. Alim. (t_A) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).



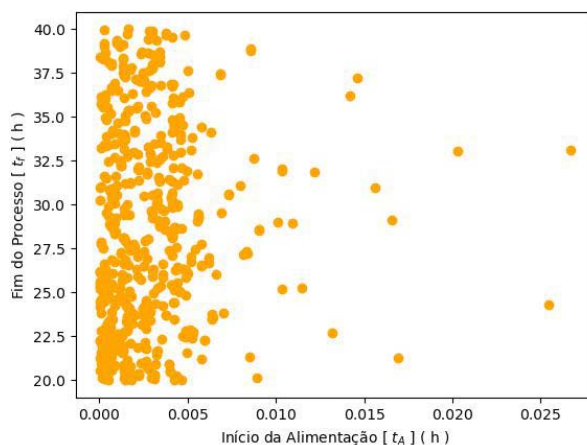
(c) I. Alim. (t_A) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).



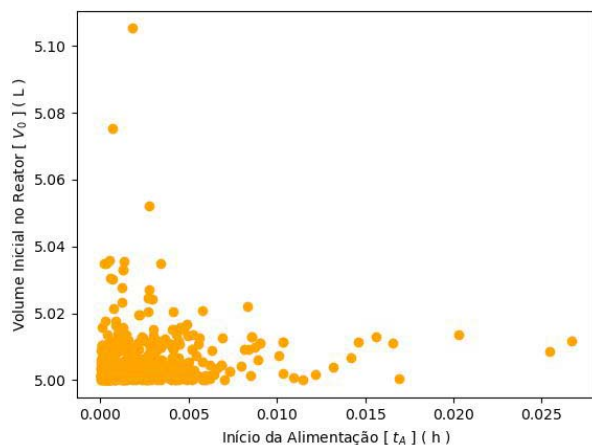
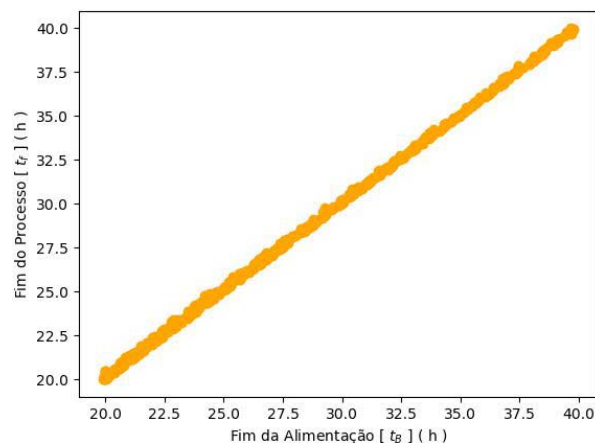
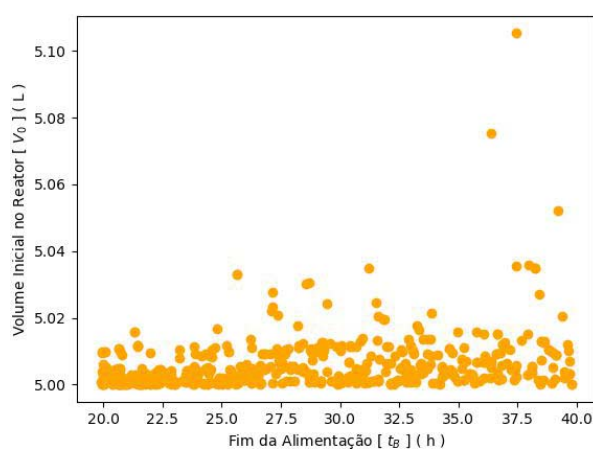
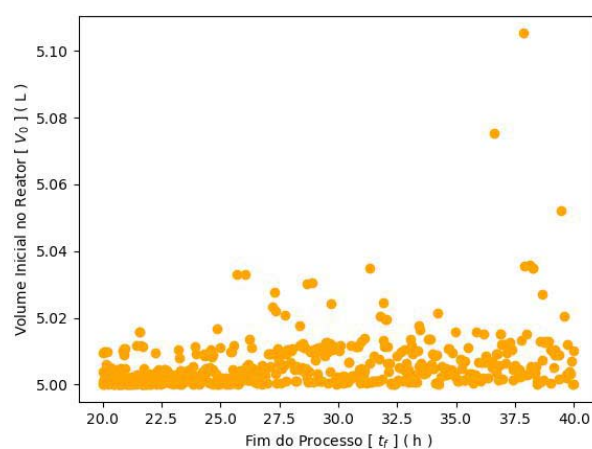
(d) F. Alim. (t_B) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).



(e) I. Alim. (t_A) \times F. Alim. (t_B).



(f) I. Alim. (t_A) \times F. Proc. (t_f).

(g) I. Alim. (t_A) \times V. Ini. Reat. (V_0).(h) F. Alim. (t_B) \times F. Proc. (t_f).(i) F. Alim. (t_B) \times V. Ini. Reat. (V_0).(j) F. Proc. (t_f) \times V. Ini. Reat. (V_0).

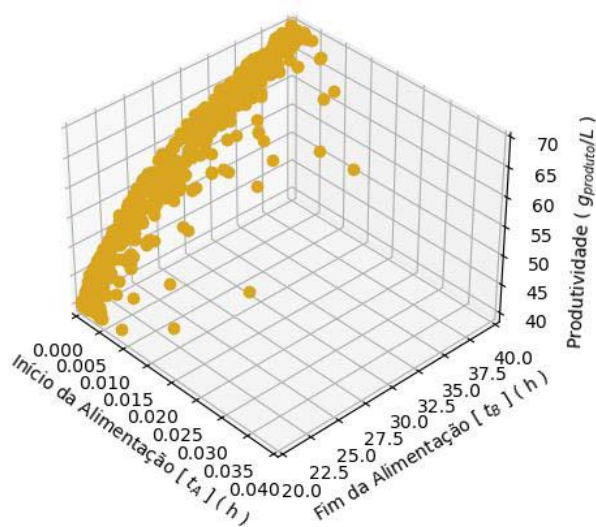
Fonte: O autor, 2023.

8.2.2.2.3 Relação entre os Otimizadores de Pareto e os Objetivos

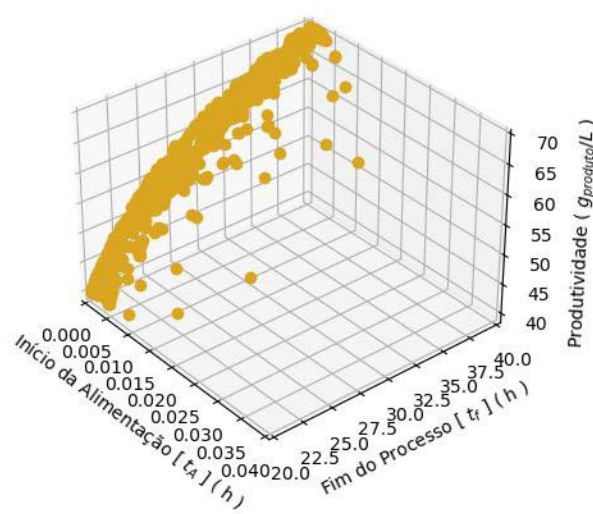
A seguir, serão apresentadas as correspondências dos pontos do conjunto ótimo de Pareto, mostrado no Gráfico 20, com cada um dos objetivos do problema de otimização em questão.

O Gráfico 21 mostra a produtividade relacionada a cada um dos otimizadores do conjunto ótimo de Pareto e o Gráfico 22 enuncia o tempo final relacionado a cada um dos otimizadores de Pareto encontrados.

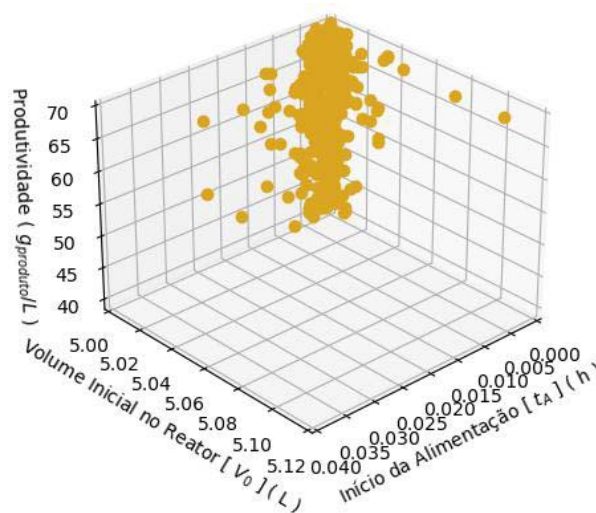
Gráfico 21 - Produtividade correspondente aos otimizadores de Pareto do Problema B do Estudo de Caso 1 obtidos pelo AGE-MOEA.



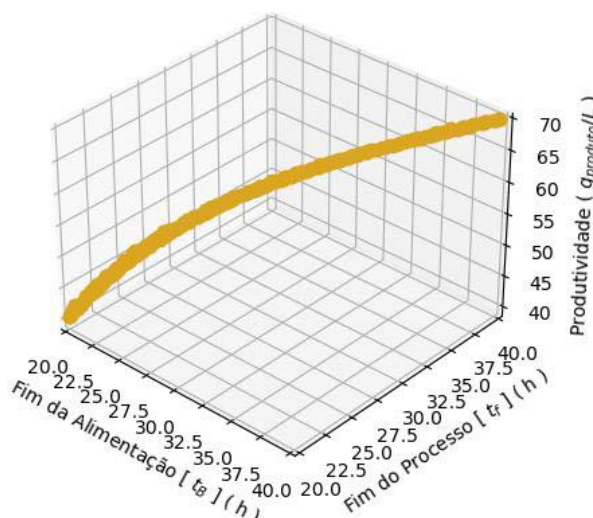
(a) Produtividade ($Prod$) \times I. Alim. (t_A) \times F. Alim. (t_B).



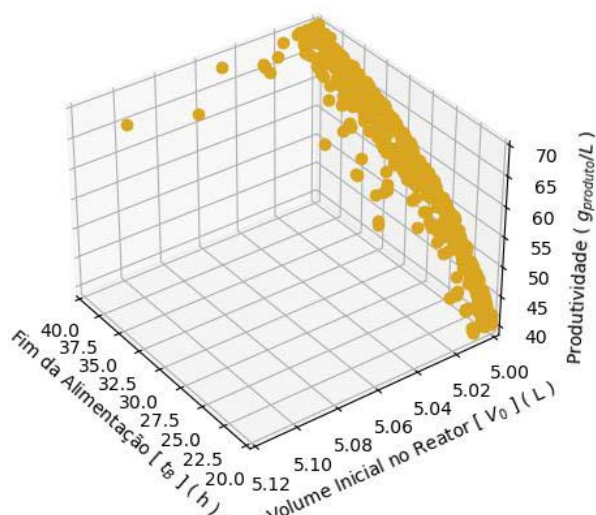
(b) Produtividade ($Prod$) \times I. Alim. (t_A) \times F. Proc. (t_f).



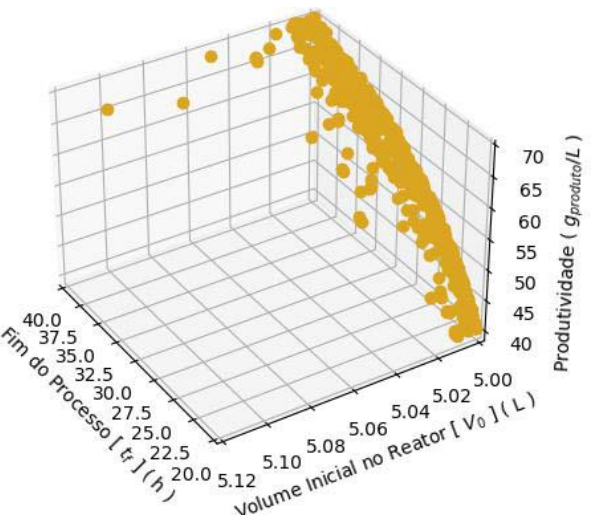
(c) Produtividade ($Prod$) \times I. Alim. (t_A) \times V. Ini. Reat. (V_0).



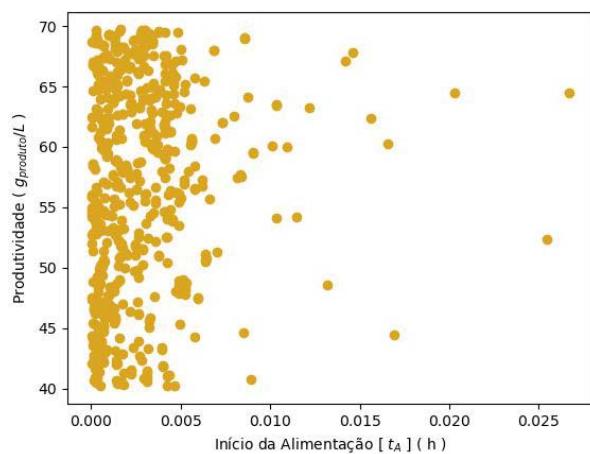
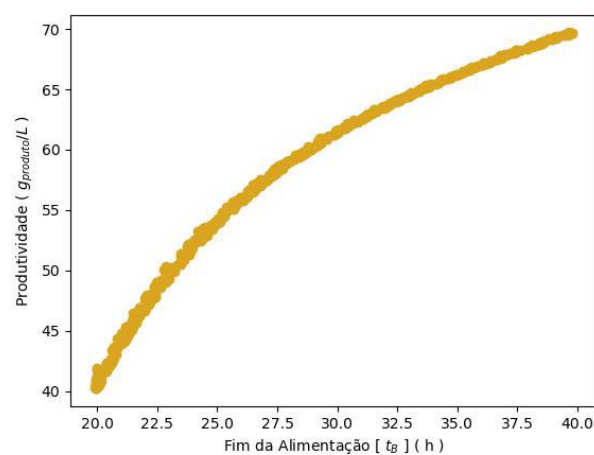
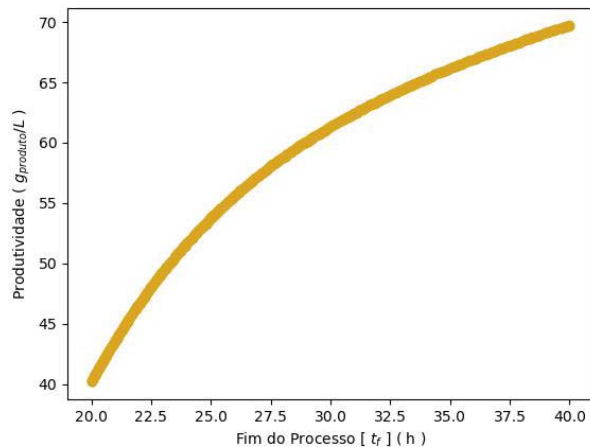
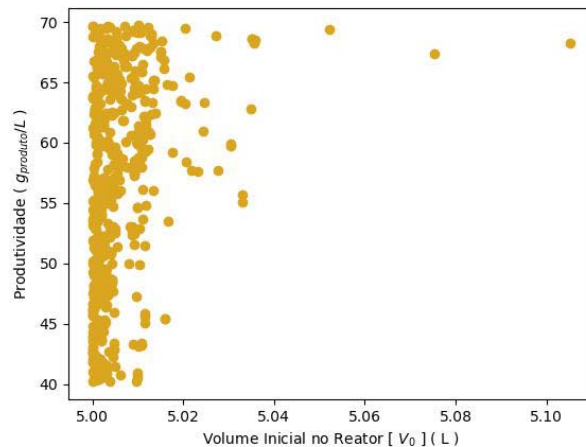
(d) Produtividade ($Prod$) \times F. Alim. (t_B) \times F. Proc. (t_f).



(e) Produtividade ($Prod$) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).

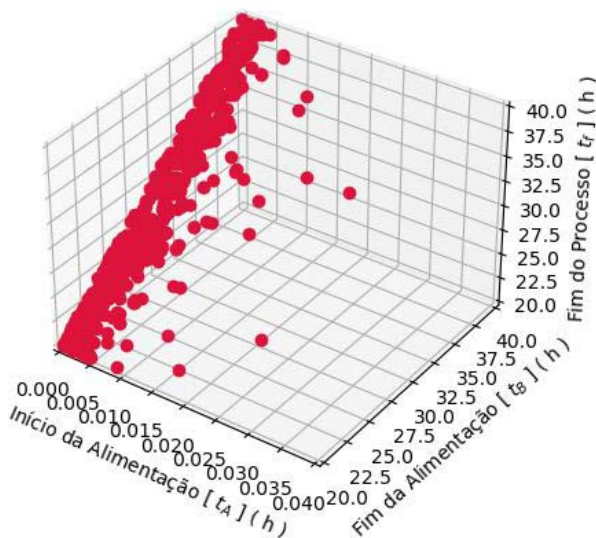


(f) Produtividade ($Prod$) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).

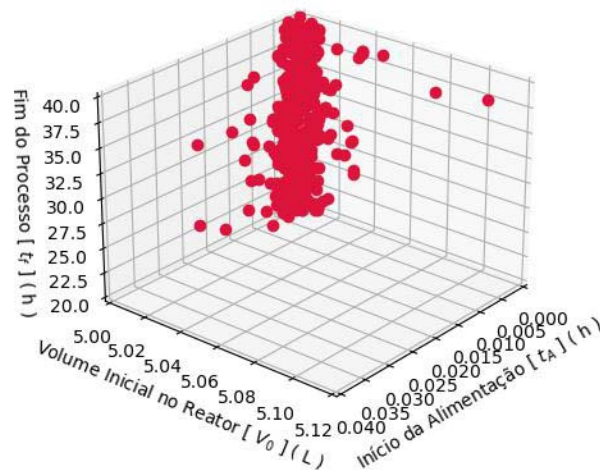
(g) Produtividade ($Prod$) \times I. Alim. (t_A).(h) Produtividade ($Prod$) \times F. Alim. (t_B).(i) Produtividade ($Prod$) \times F. Proc. (t_f).(j) Produtividade ($Prod$) \times V. Ini. Reat. (V_0).

Fonte: O autor, 2023.

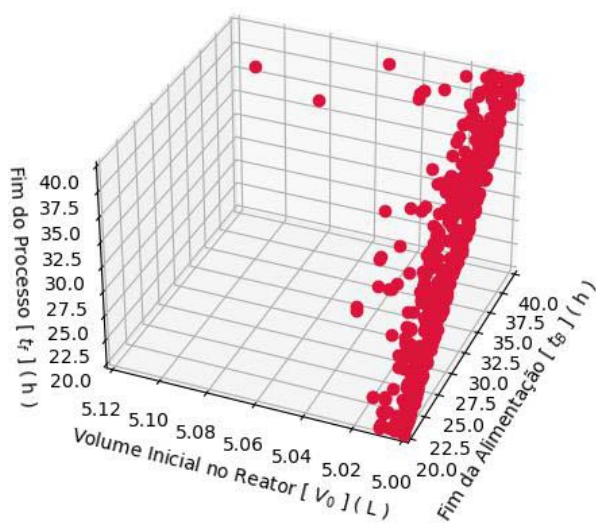
Gráfico 22 - Tempo final correspondente aos otimizadores de Pareto do Problema B do Estudo de Caso 1 obtidos pelo AGE-MOEA.



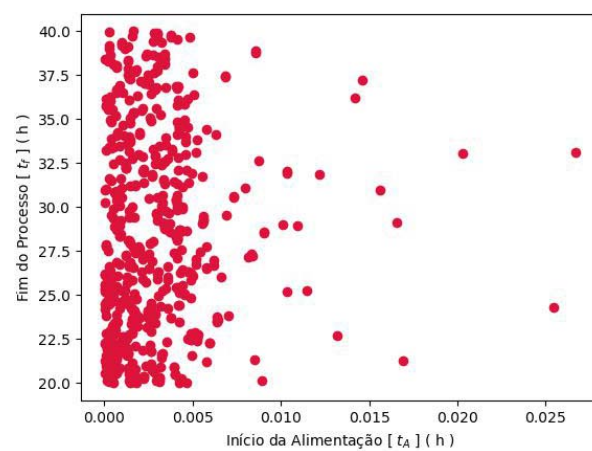
(a) Fim do Proc. (t_f) \times I. Alim. (t_A) \times F. Alim. (t_B).



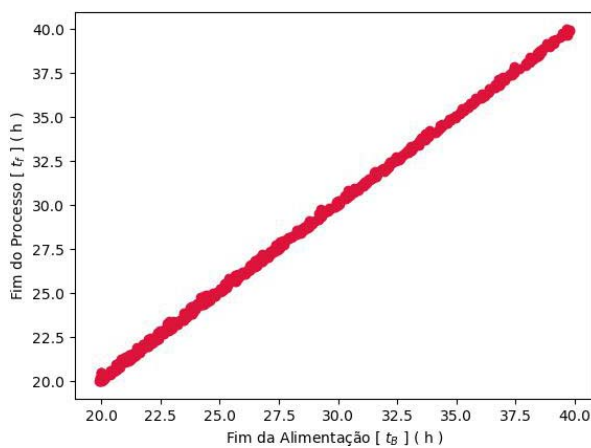
(b) Fim do Proc. (t_f) \times I. Alim. (t_A) \times V. Ini. Reat. (V_0).



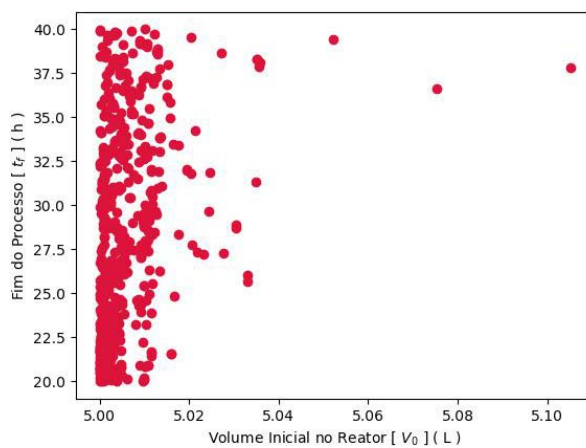
(c) Fim do Proc. (t_f) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).



(d) Fim do Proc. (t_f) \times I. Alim. (t_A).



(e) Fim do Proc. (t_f) \times F. Alim. (t_B).



(f) Fim do Proc. (t_f) \times V. Ini. Reat. (V_0).

8.2.2.3 Simulação com o C-TAEA

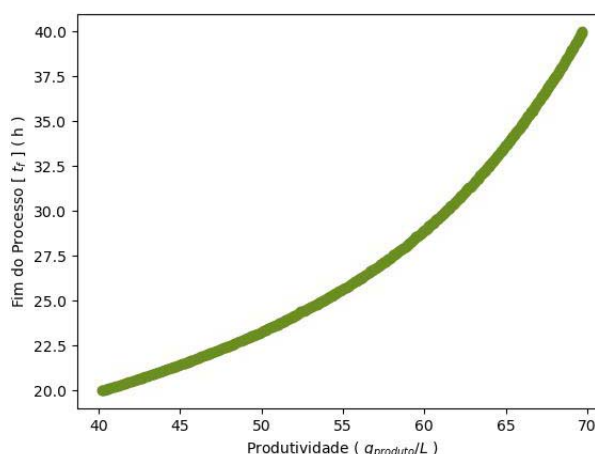
A simulação que buscou solucionar o Problema de Otimização B através do método C-TAEA levou 8 minutos e 48 segundos para atingir as 50 gerações. O método de otimização convergiu na 30^a geração, encontrando 600 soluções não dominadas.

As soluções obtidas nesta simulação estão descritas, detalhadas e ilustradas nas subseções abaixo.

8.2.2.3.1 Fronteira de Pareto

O Gráfico 23 apresenta a frente de Pareto do problema de otimização em questão, que foi estimada nesta simulação.

Gráfico 23 - Fronteira de Pareto do Problema B do Estudo de Caso 1 obtida pelo C-TAEA.

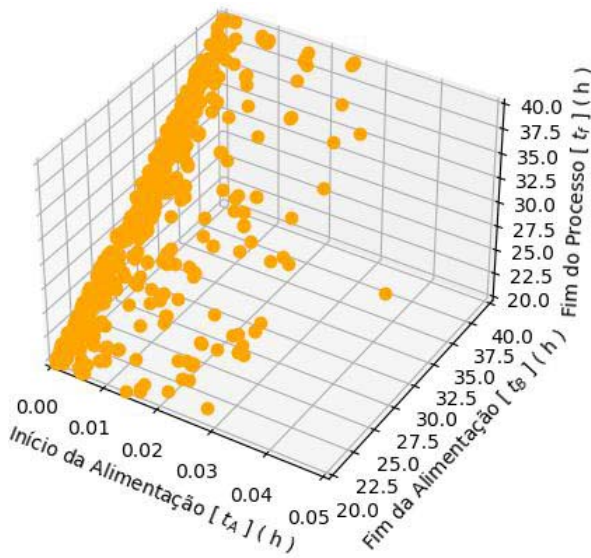


Fonte: O autor, 2023.

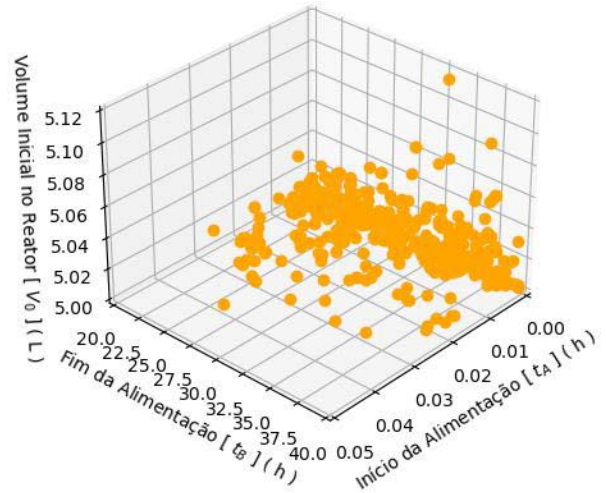
8.2.2.3.2 Conjunto Ótimo de Pareto

O Gráfico 24 mostra os otimizadores de Pareto encontrados nesta simulação de otimização, nos quais consiste o conjunto ótimo de Pareto estimado.

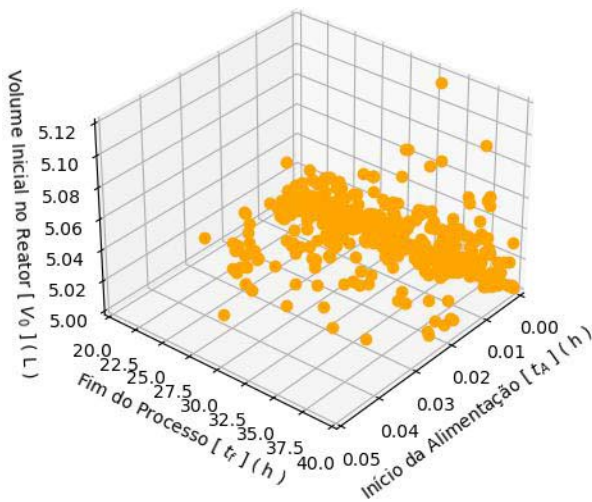
Gráfico 24 - Conjunto ótimo de Pareto do Problema B do Estudo de Caso 1 obtido pelo C-TAEA.



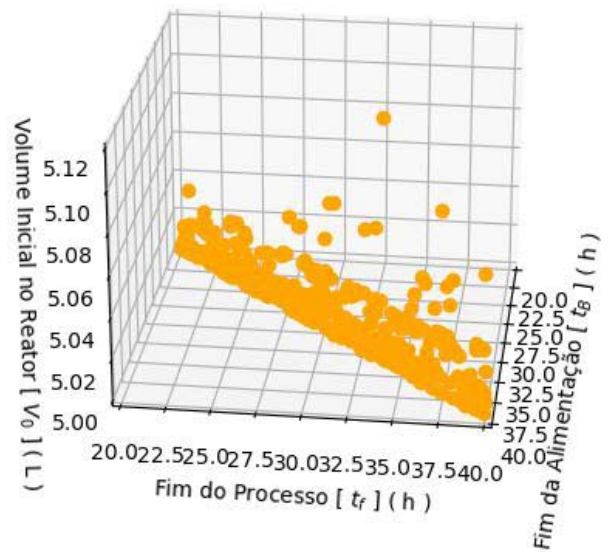
(a) I. Alim. (t_A) \times F. Alim. (t_B) \times F. Proc. (t_f).



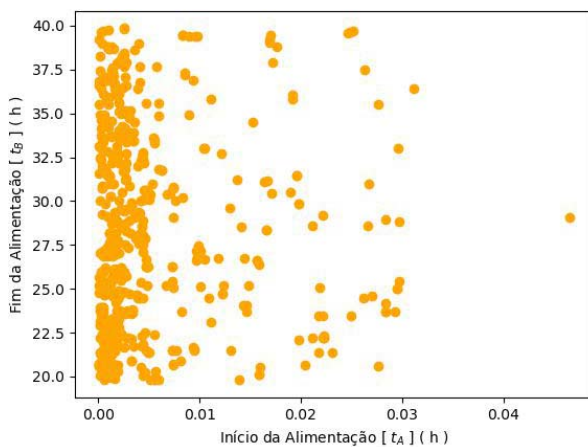
(b) I. Alim. (t_A) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).



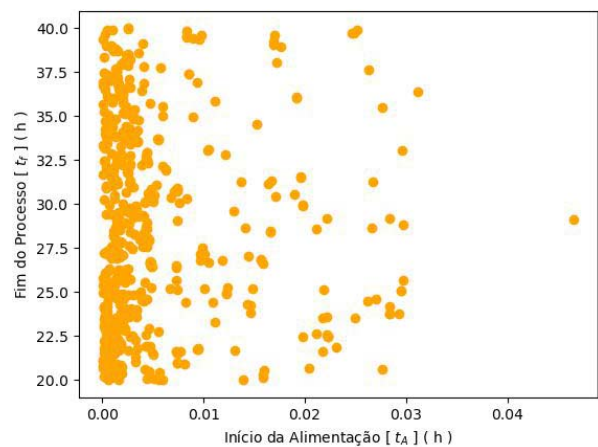
(c) I. Alim. (t_A) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).



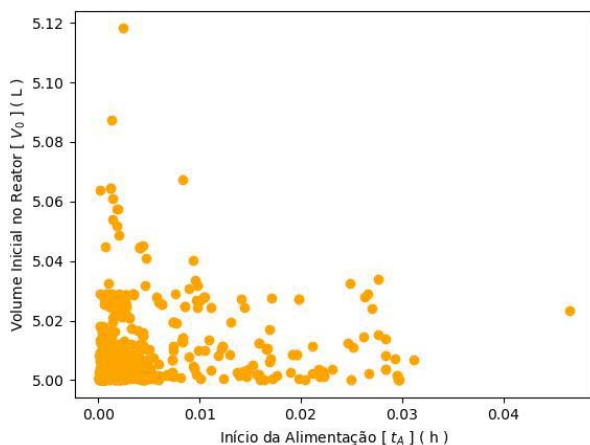
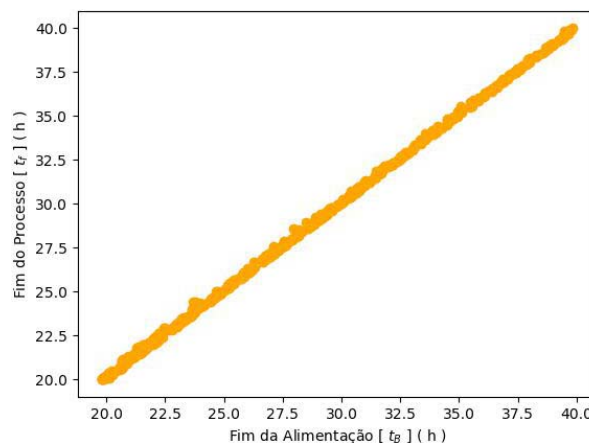
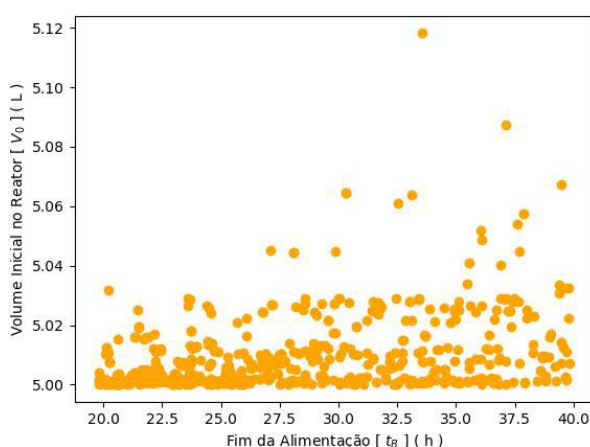
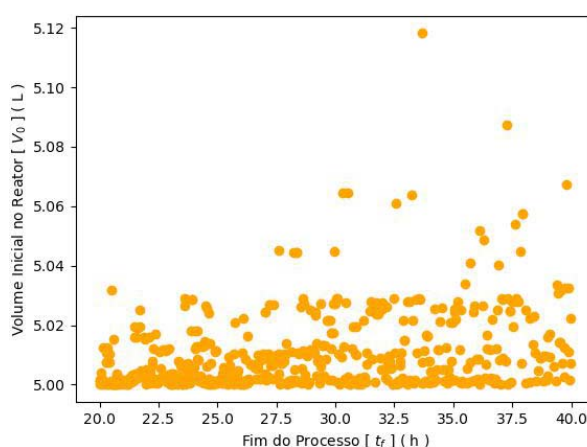
(d) F. Alim. (t_B) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).



(e) I. Alim. (t_A) \times F. Alim. (t_B).



(f) I. Alim. (t_A) \times F. Proc. (t_f).

(g) I. Alim. (t_A) \times V. Ini. Reat. (V_0).(h) F. Alim. (t_B) \times F. Proc. (t_f).(i) F. Alim. (t_B) \times V. Ini. Reat. (V_0).(j) F. Proc. (t_f) \times V. Ini. Reat. (V_0).

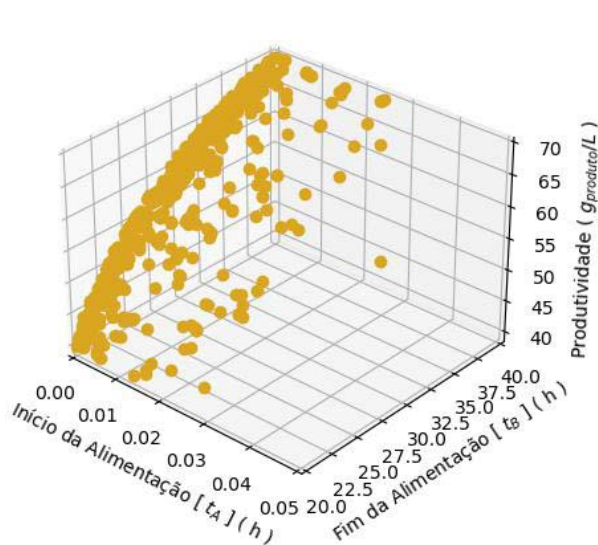
Fonte: O autor, 2023.

8.2.2.3.3 Relação entre os Otimizadores de Pareto e os Objetivos

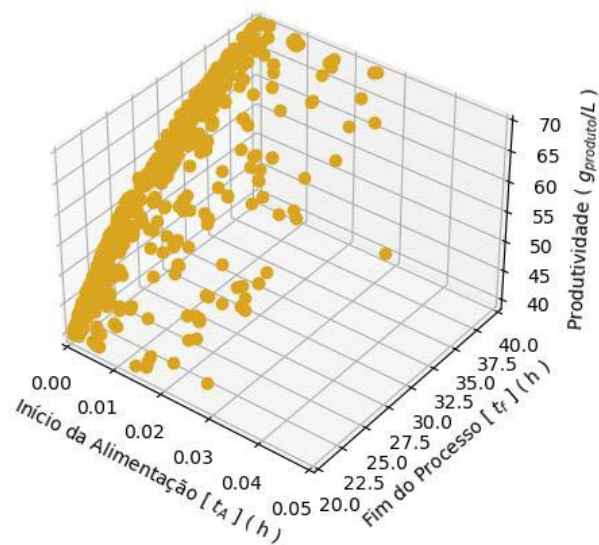
A seguir, serão apresentadas as correspondências dos pontos do conjunto ótimo de Pareto, mostrado no Gráfico 24, com cada um dos objetivos do problema de otimização em questão.

O Gráfico 25 mostra a produtividade relacionada a cada um dos otimizadores do conjunto ótimo de Pareto e o Gráfico 26 enuncia o tempo final relacionado a cada um dos otimizadores de Pareto encontrados.

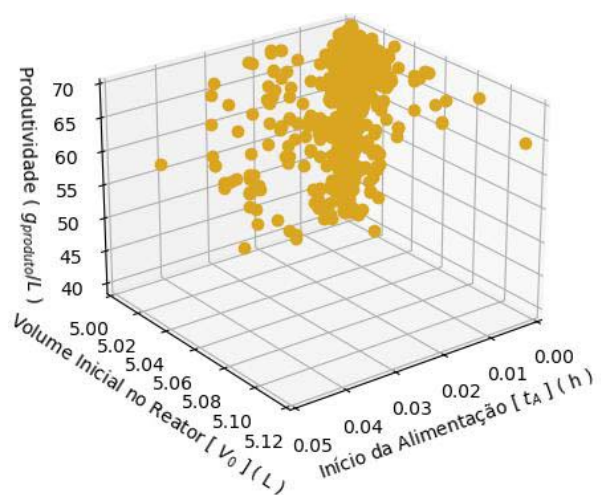
Gráfico 25 - Produtividade correspondente aos otimizadores de Pareto do Problema B do Estudo de Caso 1 obtidos pelo C-TAEA.



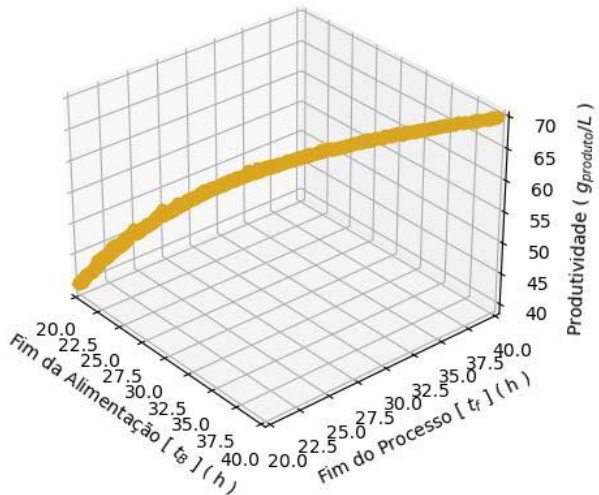
(a) Produtividade ($Prod$) \times I. Alim. (t_A) \times F. Alim. (t_B).



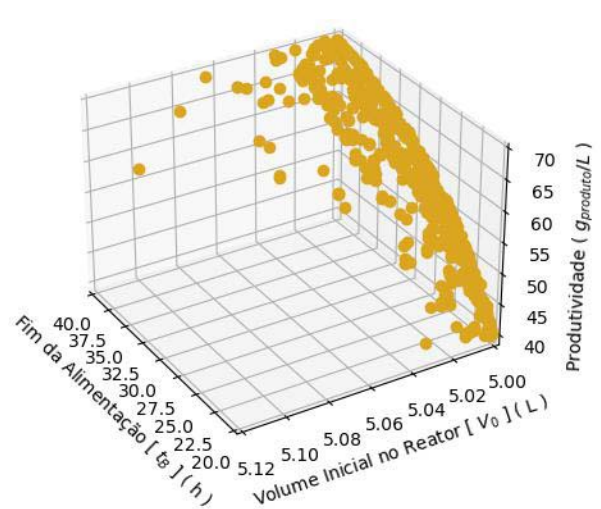
(b) Produtividade ($Prod$) \times I. Alim. (t_A) \times F. Proc. (t_f).



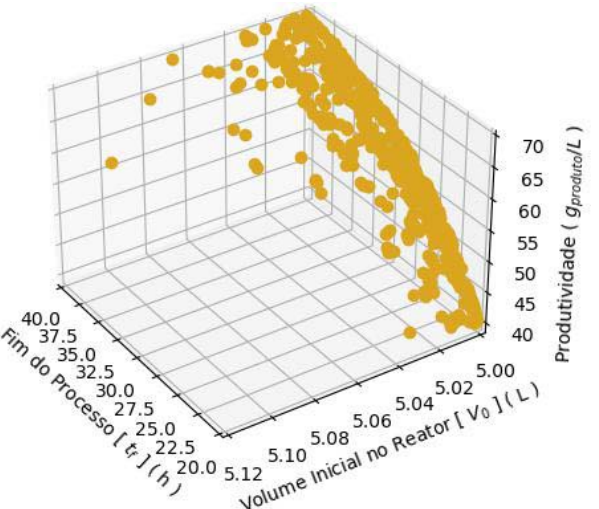
(c) Produtividade ($Prod$) \times I. Alim. (t_A) \times V. Ini. Reat. (V_0).



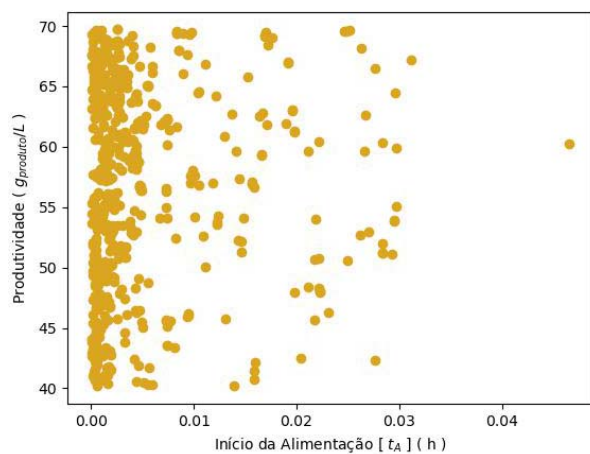
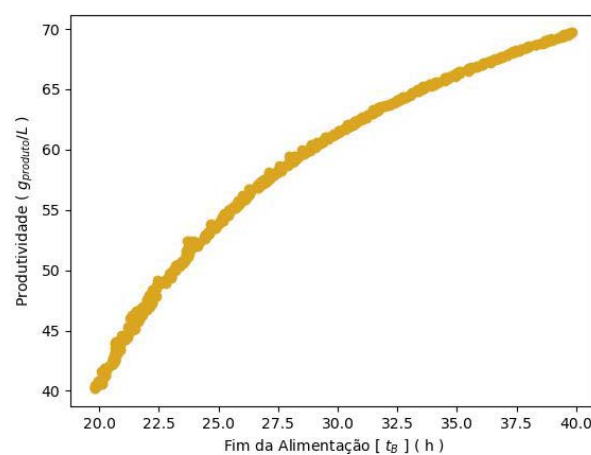
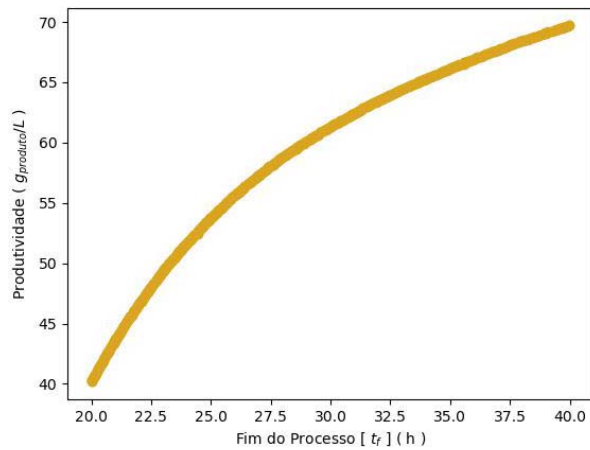
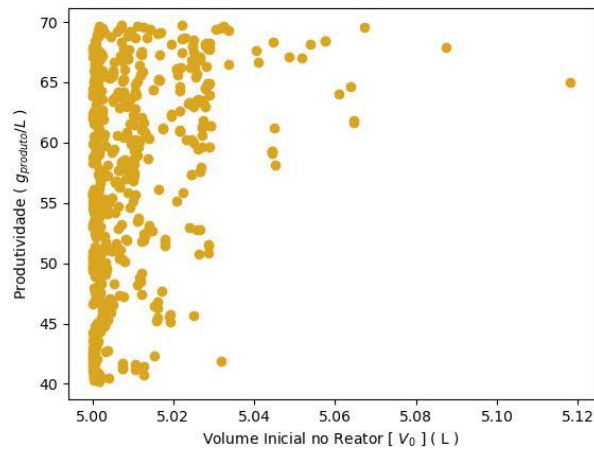
(d) Produtividade ($Prod$) \times F. Alim. (t_B) \times F. Proc. (t_f).



(e) Produtividade ($Prod$) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).

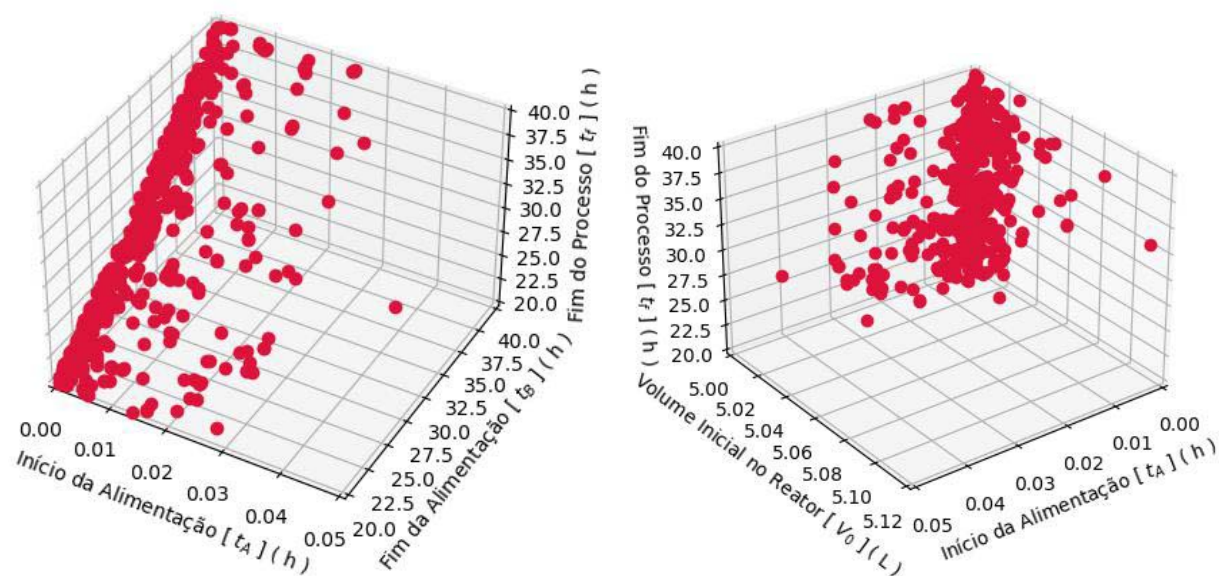


(f) Produtividade ($Prod$) \times F. Proc. (t_f) \times V. Ini. Reat. (V_0).

(g) Produtividade ($Prod$) \times I. Alim. (t_A).(h) Produtividade ($Prod$) \times F. Alim. (t_B).(i) Produtividade ($Prod$) \times F. Proc. (t_f).(j) Produtividade ($Prod$) \times V. Ini. Reat. (V_0).

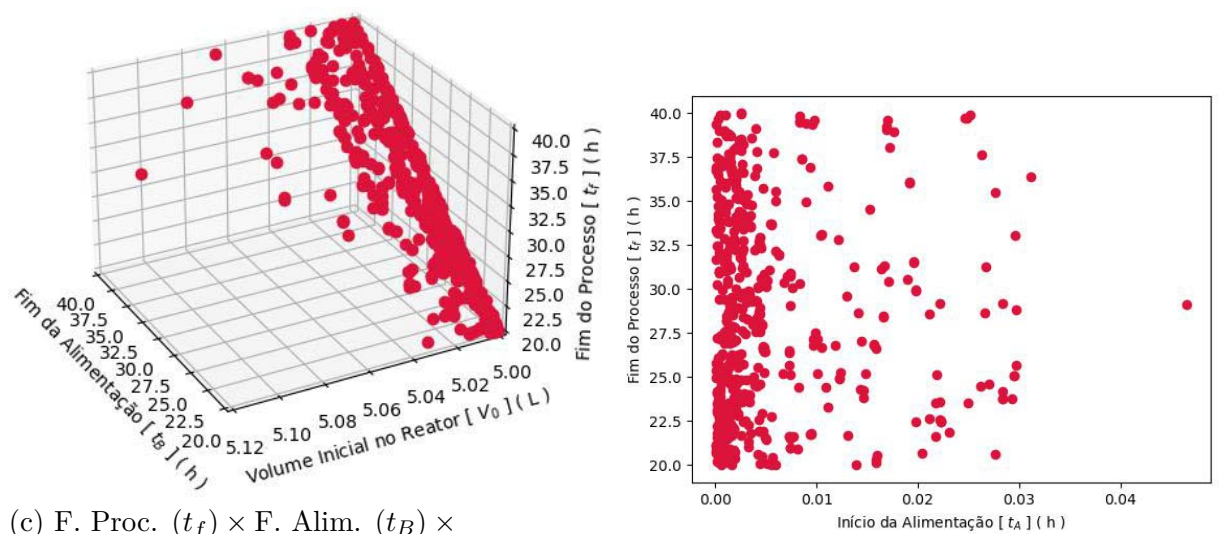
Fonte: O autor, 2023.

Gráfico 26 - Tempo final correspondente aos otimizadores de Pareto do Problema B do Estudo de Caso 1 obtidos pelo C-TAEA.



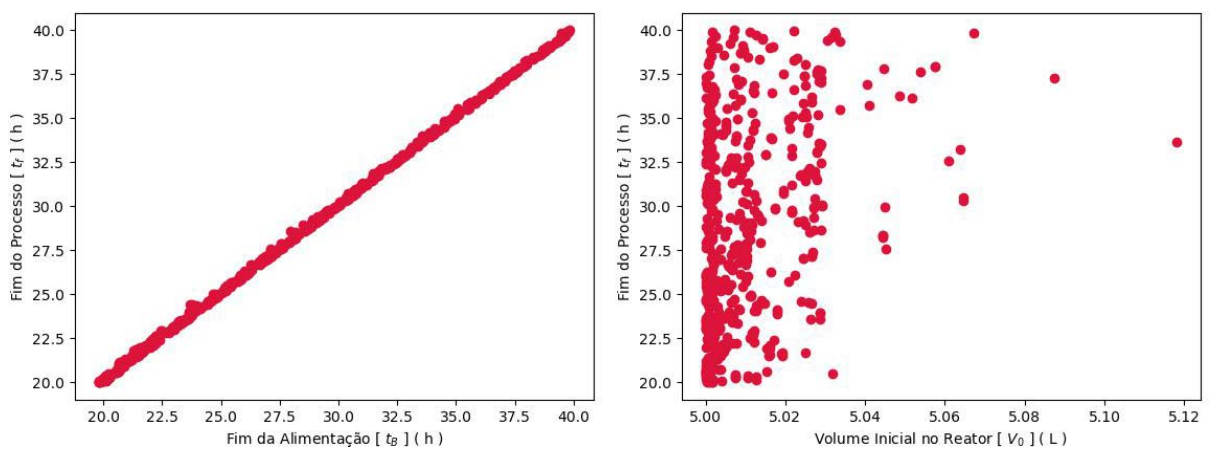
(a) F. Proc. (t_f) \times I. Alim. (t_A) \times F. Alim. (t_B).

(b) F. Proc. (t_f) \times I. Alim. (t_A) \times V. Ini. Reat. (V_0).



(c) F. Proc. (t_f) \times F. Alim. (t_B) \times V. Ini. Reat. (V_0).

(d) F. Proc. (t_f) \times I. Alim. (t_A).



(e) F. Proc. (t_f) \times F. Alim. (t_B).

(f) F. Proc. (t_f) \times V. Ini. Reat. (V_0).

8.2.2.4 Soluções Notáveis

Considerando os ótimos de Pareto obtidos em todas as simulações de otimização realizadas sobre este problema, foram elencadas as soluções que apresentam o melhor valor atingido para cada um dos objetivos, bem como, soluções que equilibram ambos os objetivos.

A Tabela 11 expressa cada uma das soluções elencadas, bem como o método usado em sua obtenção.

Tabela 11 - Soluções Notáveis do Problema B do Estudo de Caso 1.

Solução	Variáveis				Objetivos	
	t_A	t_B	t_f	V_0	$Prod(t_f)$	t_f
AGE-MOEA						
E1P2M1S1	0,00160	39,63237	39,9991	5,01013	69,70793	39,9991
E1P2M1S2	0,00053	19,92817	20,00001	5,00964	40,20843	20,00001
E1P2M1S3	3,20333	25,42198	25,64420	5,00337	54,98301	25,64420
C-TAEA						
E1P2M2S1	0,00250	39,85439	39,99693	5,00725	69,70339	39,99693
E1P2M2S2	0,01389	19,78969	20,00000	5,00157	40,17601	20,00000
E1P2M2S3	0,00042	25,41747	25,61491	5,00401	54,95747	25,61491
NSGA-II						
E1P2M3S1	0,00068	39,70700	39,99267	5,01048	69,70820	39,99267
E1P2M3S2	0,00334	19,99876	20,00000	7,80049	31,82653	20,00000
E1P2M3S3	0,00016	23,43747	23,60059	5,00889	50,79299	23,60059

Fonte: O autor, 2023.

8.3 Análise e Discussão do Estudo de Caso 1

Na presente seção, abordaremos os dados e resultados obtidos através das simulações que foram mostrados nas seções anteriores, realizando análises destas informações e discutindo pontos relevantes que sejam inerentes a elas.

8.3.1 Problema de Otimização A

Na presente subseção, estão relacionadas e descritas as análises e discussões feitas em relação às informações e resultados obtidos a partir das simulações realizadas sobre o Problema de Otimização A deste estudo de caso.

8.3.1.1 Considerações sobre os Métodos de Otimização

Embora não seja o intuito deste trabalho comparar os métodos de otimização usados nas simulações realizadas, cabe constatar que, com relação ao tempo de convergência dos métodos de otimização nas referidas simulações, todos os métodos levaram uma quantidade semelhante de iterações para convergir.

No que se refere ao tempo de execução levado pelos métodos de otimização para completar as 50 gerações, todos os métodos levaram tempos bem próximos uns dos outros, não existindo uma diferença relevante a ser ressaltada.

No que diz respeito à quantidade de soluções encontradas, todos os métodos encontraram 600 soluções não dominadas.

Todos os métodos de otimização empregados convergiram para resultados muito próximos uns dos outros. Deste modo, não existe razão para analisar e discutir separadamente os resultados de cada um, uma vez que todos corroboraram entre si.

8.3.1.2 Considerações sobre os Resultados

Levando em conta os resultados das simulações de otimização realizadas sobre o Problema A deste estudo de caso, constata-se que não existe vantagem para nenhum dos objetivos quando o volume inicial no reator é ampliado.

Este fato provavelmente se deve ao aumento no volume do reator não ser acompanhado proporcionalmente pela quantidade de substrato, remetendo a ineficiência em aumentar o volume do reator com algum material sem açúcares.

Dessa forma, é mais vantajoso manter o volume inicial do reator próximo ao mínimo, cujos açúcares iniciais são proporcionais, e enchê-lo com mosto açucarado durante a alimentação.

O tempo final de operação foi mantido próximo do máximo em todas as soluções. Essa condição se deu possivelmente para que houvesse o maior tempo possível para a geração de lisina, sendo esta geração expressiva ao ponto de compensar positivamente o tempo gasto, uma vez que a produtividade leva em conta ambos os fatores.

Em todas as soluções, o instante de tempo em que a alimentação do reator é finalizada está muito próximo do instante em que o processo termina, possivelmente para permitir que a maior quantidade de suco contendo substrato seja injetada no reator, permitindo a maior geração de lisina possível. O controle da quantidade de substrato inserido no reator ficou a cargo do instante de tempo em que se inicia a alimentação deste.

A variável que apresentou maior influência na relação entre objetivos foi o instante de tempo de início da alimentação do reator. Quanto mais rápido se inicia a alimentação

do reator, mais açúcares são inseridos no sistema, levando a uma maior geração de lisina pelas bactérias, elevando assim a produtividade. Entretanto, o rendimento tende a ser menor em virtude da maior quantidade de substratos injetados no sistema.

Em contrapartida, quanto mais tempo levar para se iniciar a alimentação, menor será a quantidade total de substrato inserido no sistema, fazendo com que o rendimento aumente, uma vez que haverá um menor consumo de substrato em relação a lisina gerada, e conseqüentemente, que a produtividade caia, pois menos lisina será gerada no mesmo intervalo de tempo.

Desse modo, estabelece-se uma relação conflitiosa entre os objetivos, de forma que, quanto maior a produtividade obtida no processo, menor é o rendimento alcançado e vice-versa.

8.3.1.3 Análise de Cenários

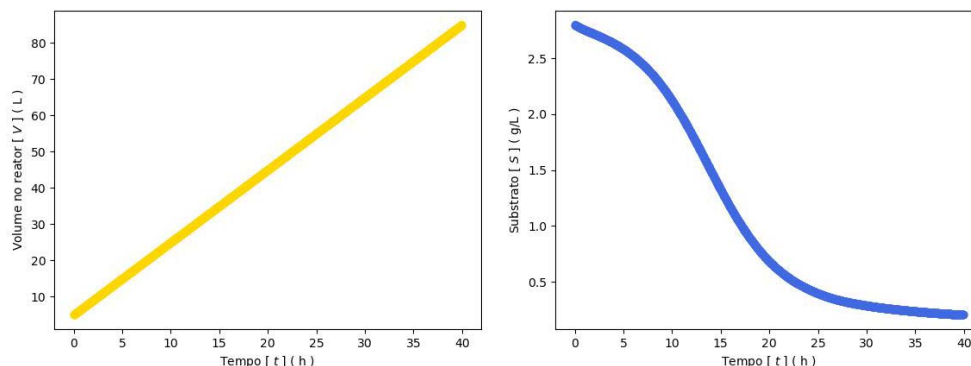
Levando em conta as considerações apresentadas e os resultados obtidos, podemos destacar alguns cenários para serem analisados e discutidos.

8.3.1.3.1 Cenário 1

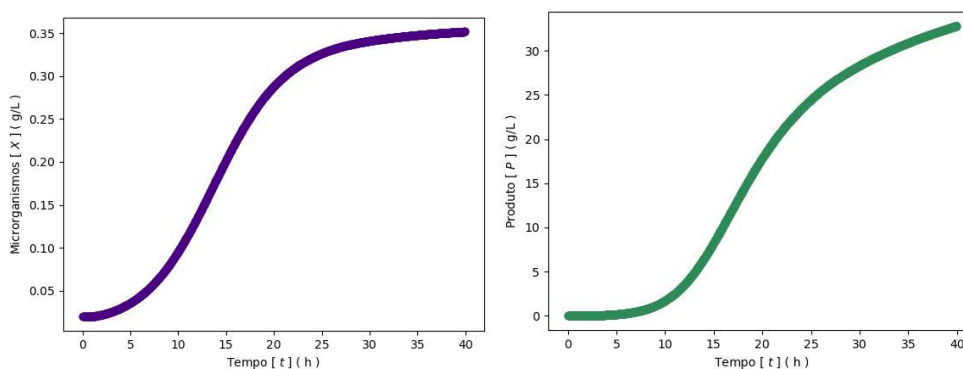
O primeiro cenário elencado para este problema considerou o otimizador E1P1M2S1, mostrado na Tabela 10, que leva à maior produtividade dentre os otimizadores encontrados.

O Gráfico 27 apresenta os comportamentos das concentrações de cada um dos componentes ao longo do tempo no reator de fermentação para o processo que considera o presente otimizador de Pareto.

Gráfico 27 - Concentrações ao longo do tempo no processo que adota a solução dada no Cenário 1 levantado para o Problema A do Estudo de Caso 1.



(a) Volume no reator no processo que adota o otimizador deste cenário. (b) Concentração de substrato no reator no processo que adota o otimizador deste cenário.



(c) Concentração de microrganismos no reator no processo que adota o otimizador deste cenário. (d) Concentração de produto no reator no processo que adota o otimizador deste cenário.

Fonte: O autor, 2023.

No presente cenário, ao término do processo, o volume ocupado no reator foi de 84,99446 L, tendo sido consumidos 2,59632 g/L de substrato, gerado 32,80856 g/L de lisina, restando 0,35167 g/L de bactérias no reator.

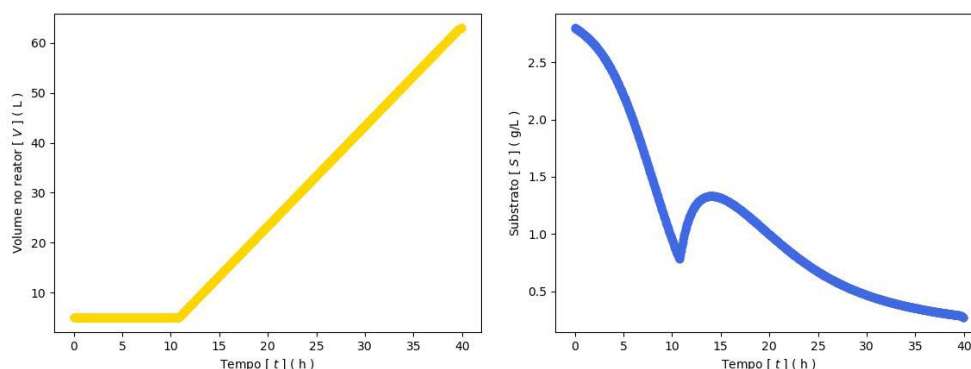
Em valores absolutos, houve o consumo de 220,67281 g de substrato, formação de 2.788,54 g de lisina, restando 29,89 g de bactérias no reator.

8.3.1.3.2 Cenário 2

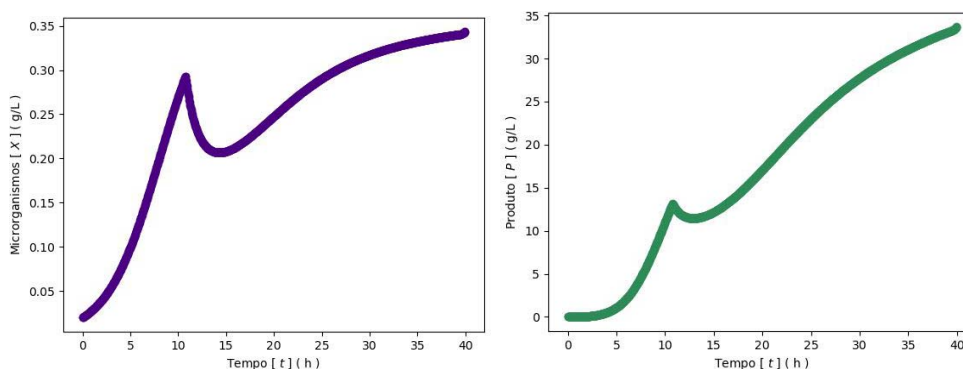
O segundo cenário elencado para este problema considerou a solução E1P1M2S2, mostrada na Tabela 10, que leva ao maior rendimento dentre todas as soluções encontrados.

O Gráfico 28 apresenta os comportamentos das concentrações de cada um dos componentes ao longo do tempo no reator de fermentação para o processo que considera o presente otimizador de Pareto.

Gráfico 28 - Concentrações ao longo do tempo no processo que adota a solução dada no Cenário 2 levantado para o Problema A do Estudo de Caso 1.



- (a) Volume no reator no processo que adota o otimizador deste cenário. (b) Concentração de substrato no reator no processo que adota o otimizador deste cenário.



- (c) Concentração de microrganismos no reator no processo que adota o otimizador deste cenário. (d) Concentração de produto no reator no processo que adota o otimizador deste cenário.

Fonte: O autor, 2023.

No presente cenário, ao término do processo, o volume ocupado no reator foi de 62,96595 L, tendo sido consumidos 2,53527 g/L de substrato, gerado 33,71944 g/L de lisina, restando 0.34384 g/L de bactérias no reator.

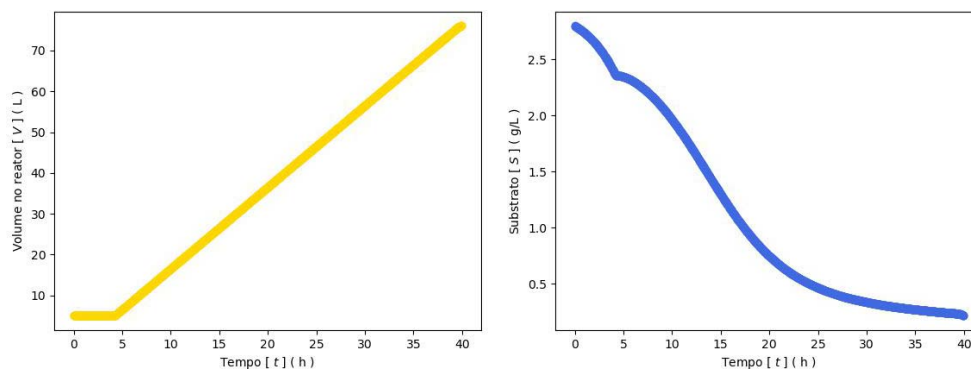
Em valores absolutos, houve o consumo de 159,63568 g de substrato, formação de 2.123,17 g de lisina, restando 21,65021 g de bactérias no reator.

8.3.1.3.3 Cenário 3

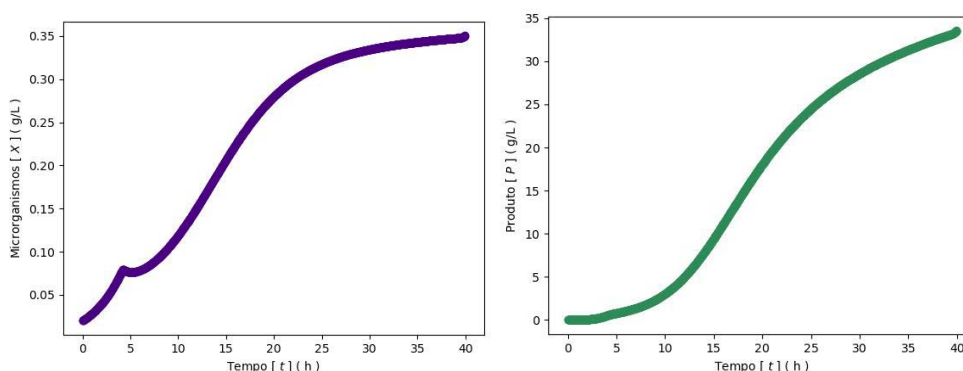
O terceiro cenário elencado para este problema considerou a solução E1P1M2S3, mostrada na Tabela 10, que equilibra a produtividade e o rendimento obtidos no processo de otimização.

O Gráfico 29 apresenta os comportamentos das concentrações de cada um dos componentes ao longo do tempo no reator de fermentação para o processo que considera o presente otimizador de Pareto.

Gráfico 29 - Concentrações ao longo do tempo no processo que adota a solução dada no Cenário 3 levantado para o Problema A do Estudo de Caso 1.



- (a) Volume no reator no processo que adota o otimizador deste cenário. (b) Concentração de substrato no reator no processo que adota o otimizador deste cenário.



- (c) Concentração de microrganismos no reator no processo que adota o otimizador deste cenário. (d) Concentração de produto no reator no processo que adota o otimizador deste cenário.

Fonte: O autor, 2023.

No presente cenário, ao término do processo, o volume ocupado no reator foi de 76,03335 L, tendo sido consumidos 2,58879 g/L de substrato, gerado 33,57079 g/L de lisina, restando 0,35079 g/L de bactérias no reator.

Em valores absolutos, houve o consumo de 196,83437 g de substrato, formação de 2.552,49 g de lisina, restando 26,67173 g de bactérias no reator.

8.3.2 Problema de Otimização B

Na presente subseção, estão relacionadas e descritas as análises e discussões feitas em relação às informações e resultados obtidos a partir das simulações realizadas sobre o Problema de Otimização B deste estudo de caso.

8.3.2.1 Considerações sobre os Métodos de Otimização

Mesmo sem fins de comparação de métodos, verifica-se que, com relação ao tempo de convergência dos métodos de otimização nas referidas simulações, todos os métodos levaram uma quantidade semelhante de iterações para convergir.

No que se refere ao tempo de execução levado pelos métodos de otimização para completar as 50 gerações, todos os métodos levaram tempos bem próximos uns dos outros, não existindo uma diferença relevante a ser ressaltada.

No que diz respeito à quantidade de soluções encontradas, todos os métodos encontraram 600 soluções não dominadas.

Todos os métodos de otimização empregados convergiram para resultados muito próximos uns dos outros, corroborando entre si.

8.3.2.2 Considerações sobre os Resultados

Considerando os resultados das simulações de otimização realizadas sobre o Problema B deste estudo de caso, assim como no problema anterior, percebe-se que ampliar o volume inicial no reator não traz vantagem para nenhum dos objetivos, pelas mesmas razões descritas anteriormente.

Em todas as soluções encontradas, o início da alimentação do reator ocorre o mais rápido possível, bem próximo ao início do processo de fermentação em batelada alimentada.

Provavelmente, esse rápido início da alimentação visa garantir o máximo de substrato no sistema o mais rápido possível, a fim de garantir a maior conversão de açúcares em lisina no menor tempo possível. Como neste problema, o rendimento não é objetivo, maiores quantidades de açúcares não constituem empecilho para nenhuma das soluções.

O instante de término da alimentação do reator e o instante em que o processo

termina foram as variáveis que apresentaram maior influência na relação entre os objetivos, apresentando uma forte correlação uma com a outra, uma vez que instante de término da alimentação do reator está sempre muito próximo ao instante em que o processo termina.

Desse modo, quanto maior o tempo de operação, maior a produtividade, uma vez que existe um tempo maior de alimentação, que acarreta em mais substrato no sistema, gerando maior quantidade de lisina em relação ao tempo de operação.

Por outro lado, quanto menor o tempo, menor a produtividade, uma vez que há menor quantidade de substrato no sistema e conseqüentemente menor formação de lisina no processo fermentativo.

Embora uma primeira percepção nos leve a ver os objetivos do presente problema como colaborativos, uma vez que quanto menor o tempo de operação, menor o denominador da fórmula da produtividade, dando a entender que esta seria então maior.

Entretanto, neste caso, percebe-se que o numerador da fórmula da produtividade, isto é, a quantidade de lisina produzida, apresenta um impacto expressivamente maior, levando assim à relação conflituosa entre os objetivos.

8.3.2.3 Análise de Cenários

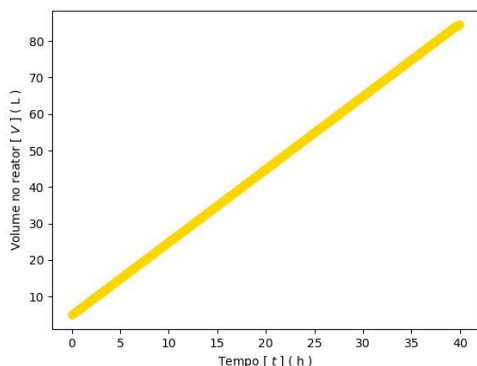
Tomando as considerações apresentadas e os resultados obtidos, podemos elencar alguns cenários para serem analisados e discutidos.

8.3.2.3.1 Cenário 1

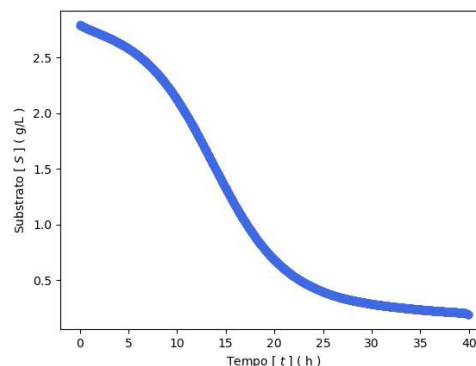
O primeiro cenário elencado para este problema considerou o otimizador E1P2M3S1, mostrado na Tabela 11, que leva à maior produtividade dentre os otimizadores encontrados.

O Gráfico 30 apresenta os comportamentos das concentrações de cada um dos componentes ao longo do tempo no reator de fermentação para o processo que considera o presente otimizador de Pareto.

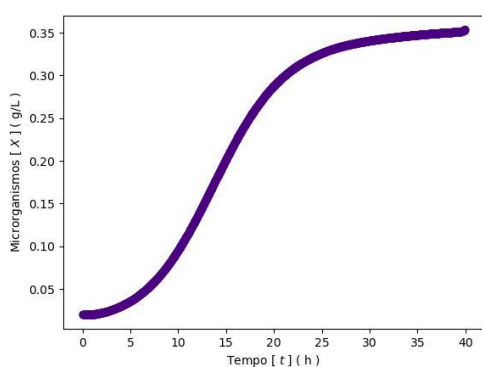
Gráfico 30 - Concentrações ao longo do tempo no processo que adota a solução dada no Cenário 1 levantado para o Problema B do Estudo de Caso 1.



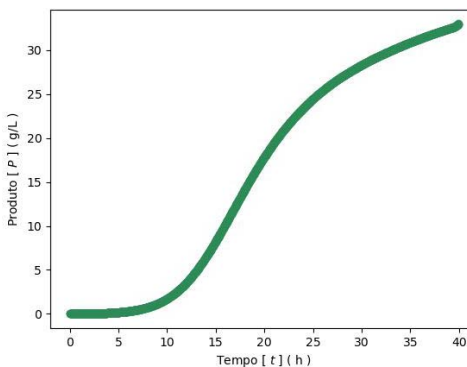
(a) Volume no reator no processo que adota o otimizador deste cenário.



(b) Concentração de substrato no reator no processo que adota o otimizador deste cenário.



(c) Concentração de microrganismos no reator no processo que adota o otimizador deste cenário.



(d) Concentração de produto no reator no processo que adota o otimizador deste cenário.

Fonte: O autor, 2023.

No presente cenário, ao término do processo, o volume ocupado no reator foi de 84,44146 L, tendo sido consumidos 2,61321 g/L de substrato, gerado 33,01477 g/L de lisina, restando 0,35391 g/L de bactérias no reator.

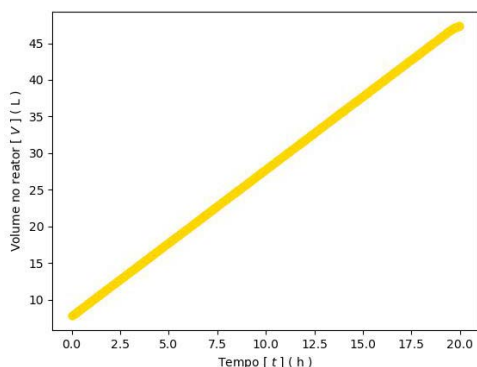
Em valores absolutos, houve o consumo de 220,66326 g de substrato, formação de 2.787,81 g de lisina, restando 29,88467 g de bactérias no reator.

8.3.2.3.2 Cenário 2

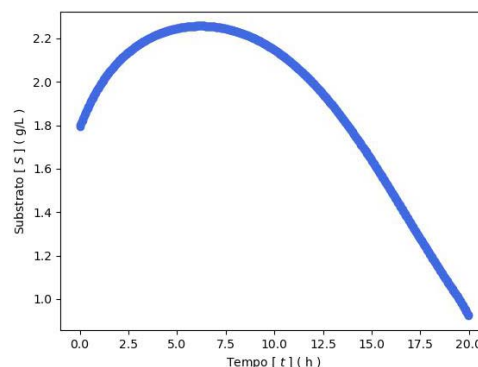
O segundo cenário elencado para este problema considerou a solução E1P2M3S2, mostrada na Tabela 11, que leva ao menor tempo de operação dentre todas as soluções encontrados.

O Gráfico 31 apresenta os comportamentos das concentrações de cada um dos componentes ao longo do tempo no reator de fermentação para o processo que considera o presente otimizador de Pareto.

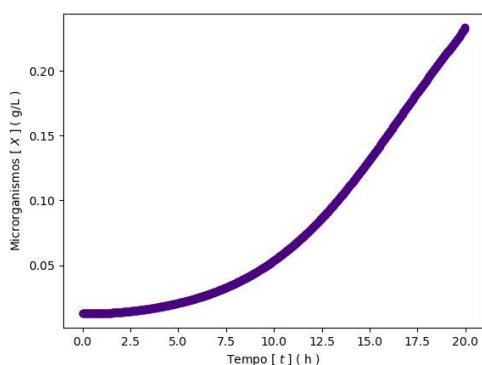
Gráfico 31 - Concentrações ao longo do tempo no processo que adota a solução dada no Cenário 2 levantado para o Problema B do Estudo de Caso 1.



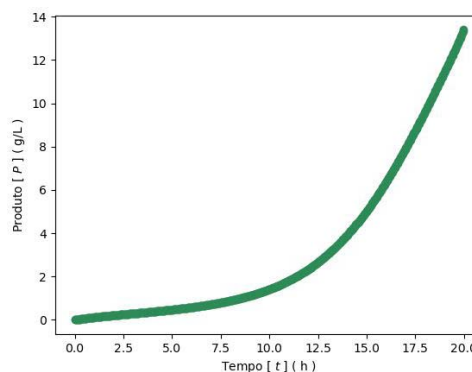
(a) Volume no reator no processo que adota o otimizador deste cenário.



(b) Concentração de substrato no reator no processo que adota o otimizador deste cenário.



(c) Concentração de microrganismos no reator no processo que adota o otimizador deste cenário.



(d) Concentração de produto no reator no processo que adota o otimizador deste cenário.

Fonte: O autor, 2023.

No presente cenário, ao término do processo, o volume ocupado no reator foi de 47,38135 L, tendo sido consumidos 1,76208 g/L de substrato, gerado 13,43421 g/L de lisina, restando 0,23360 g/L de bactérias no reator.

Em valores absolutos, houve o consumo de 83,48972 g de substrato, formação de 636,53100 g de lisina, restando 11,06828 g de bactérias no reator.

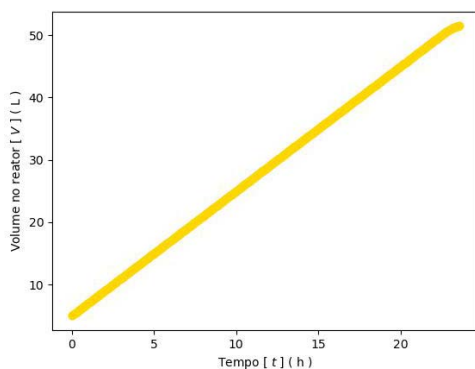
Esta foi a única solução que apresentou um volume inicial superior ao mínimo, que embora tenha mostrado alto rendimento, em virtude do baixo consumo de substrato, apresentou baixa geração de lisina quando comparada aos demais cenários.

8.3.2.3.3 Cenário 3

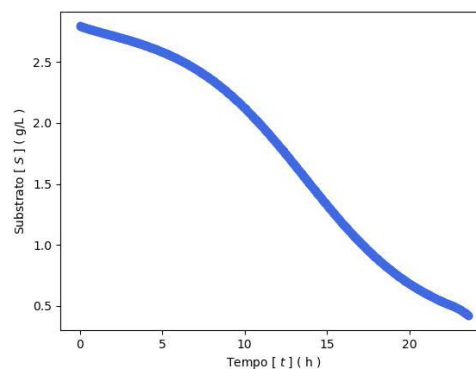
O terceiro cenário elencado para este problema considerou a solução E1P2M3S3, mostrada na Tabela 11, que equilibra a produtividade e o tempo de operação obtidos no processo de otimização.

O Gráfico 32 apresenta os comportamentos das concentrações de cada um dos componentes ao longo do tempo no reator de fermentação para o processo que considera o presente otimizador de Pareto.

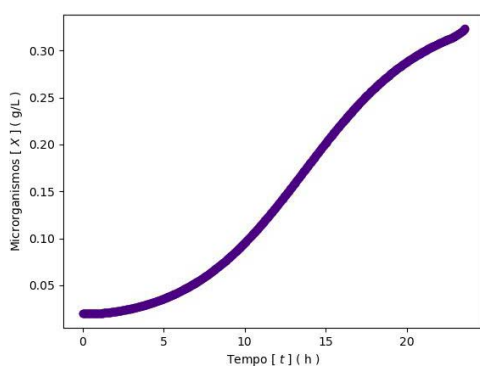
Gráfico 32 - Concentrações ao longo do tempo no processo que adota a solução dada no Cenário 3 levantado para o Problema B do Estudo de Caso 1.



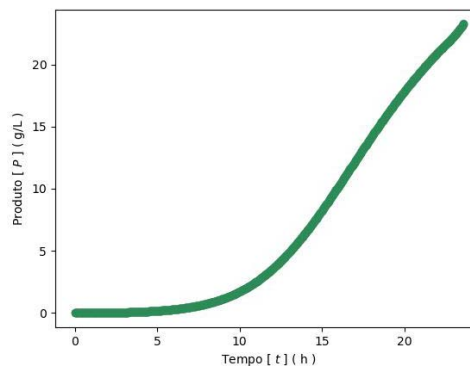
(a) Volume no reator no processo que adota o otimizador deste cenário.



(b) Concentração de substrato no reator no processo que adota o otimizador deste cenário.



(c) Concentração de microrganismos no reator no processo que adota o otimizador deste cenário.



(d) Concentração de produto no reator no processo que adota o otimizador deste cenário.

Fonte: O autor, 2023.

No presente cenário, ao término do processo, o volume ocupado no reator foi de 51,47154 L, tendo sido consumidos 2,38258 g/L de substrato, gerado 23,28929 g/L de lisina, restando 0,32352 g/L de bactérias no reator.

Em valores absolutos, houve o consumo de 122,63506 g de substrato, formação de 1.198,73 g de lisina, restando 16,65207 g de bactérias no reator.

8.3.3 Comparação dos Resultados Obtidos com os Resultados Originais

Baseando-se nos resultados obtidos, analisados e discutidos; e comparando-os com os resultados originais obtidos por Ani, Gujarathi e Vakili-Nezhaad (2021), constata-se que ambos convergiram para os mesmos valores, apresentando comportamentos similares.

8.4 Simulações e Resultados do Estudo de Caso 2

Neste segundo estudo de caso, os métodos, configurações e parâmetros usados nas simulações foram os mesmos empregados no primeiro estudo de caso desta tese, bem como a quantidade de simulações realizadas. Naturalmente, aqui, estas simulações foram feitas sobre os problemas propostos neste estudo de caso.

Entretanto, por considerar um processo fermentativo industrial real, este estudo de caso realizou uma simulação inicial do processo fermentativo com seus parâmetros originais através do Método de Runge-Kutta de 2ª Ordem (RK-2), sem qualquer ajuste ou otimização, a fim de conhecer os resultados do processo não otimizado.

Nesta simulação inicial, especificou-se 360 instantes de tempo igualmente espaçados dentro do intervalo $[t_I, t_H]$ nos quais as funções incógnitas do problema de valor inicial dado pela Equação (95) foram estimadas pelo método de resolução de equações diferenciais ordinárias.

Nas simulações de otimização, para o método de solução de equações diferenciais ordinárias, foram especificados 705 instantes de tempo igualmente espaçados dentro do intervalo $[t_I, t_f^{max}]$, onde $t_f^{max} = t_D^{max} + 1,5 + 2,0 = 23,5$, pois o tempo máximo de início da drenagem do reator, mais 1,5h de drenagem, mais 2h de higienização, resulta em 23,5h, que é tempo máximo de operação do reator. Este é o intervalo em que as funções incógnitas do problema de valor inicial dado pela Equação (95) são estimadas.

Nesta próximas subseções, serão elencados todos os resultados e informações referentes à execução das simulações de otimização sobre os problemas A e B propostos, que foram anteriormente especificadas.

8.4.1 Processo Não Otimizado

Nesta subseção, serão apresentados os resultados da simulação do processo fermentativo, apresentado na Seção 7.5, em sua forma originalmente especificada, adotando os valores da referida seção para todos parâmetros, sem realizar qualquer otimização ou alteração nestes, refletindo assim, o processo real que ocorre atualmente na planta de fermentação da qual os dados foram coletados.

8.4.1.1 Simulação com o RK-2

A simulação que objetivou solucionar o problema de valor inicial dado pela Equação (95) através do método RK-2 gerou as soluções que estão descritas e ilustradas a seguir.

O Gráfico 33 apresenta as concentrações de cada um dos componentes envolvidos no processo fermentativo, nos diferentes instantes de tempo do ciclo de fermentação em batelada alimentada.

8.4.1.2 Concentrações Obtidas ao Final do Processo

As concentrações de cada um dos componentes no vinho delevedurado ao final do ciclo de fermentação, antes da drenagem do reator, estão enunciadas na Tabela 12.

Tabela 12 - Concentrações dos componentes ao final do processo fermentativo original considerado no Estudo de Caso 2.

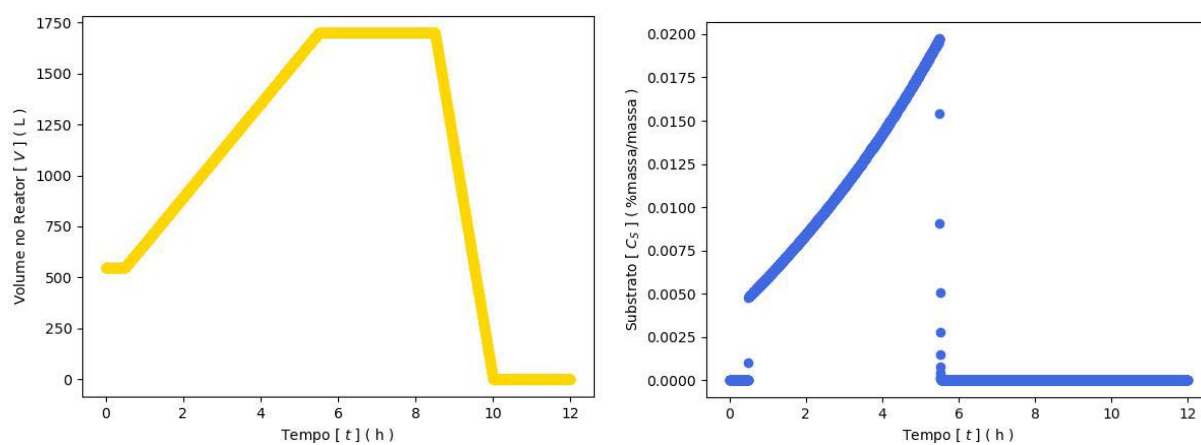
Componente	Concentração	Unidade
$C_S(t_H)$	0,00000	%massa/massa
$C_X(t_H)$	31,20690	kg/m ³
$C_P(t_H)$	8,23728	%vol/vol

Fonte: O autor, 2023.

Assim, o processo fermentativo em suas configurações originais, consome 58,7931 kg/m³ de células de levedura e apresenta conversão máxima de açúcares, produzindo 8,23728 %vol/vol de etanol, que consiste em 65,01685 kg/m³.

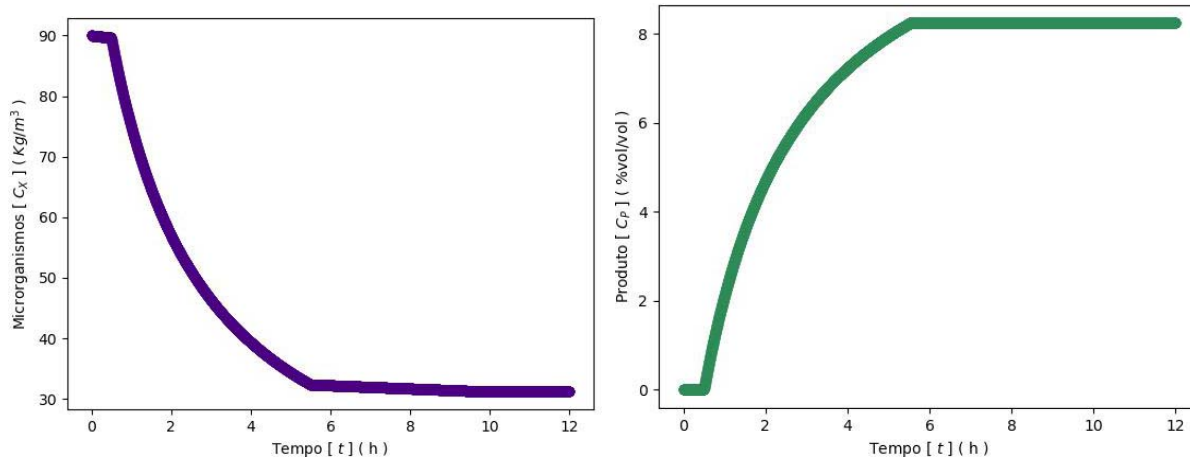
Considerando o reator em questão, que apresenta capacidade de 1.699 m³, sendo completamente preenchido no processo, gera-se, na presente configuração, um total de

Gráfico 33 - Concentrações dos componentes no decorrer do processo fermentativo original considerado no Estudo de Caso 2.



(a) Volume ocupado no reator.

(b) Concentração de substrato.



(c) Concentração de microrganismos.

(d) Concentração de produto.

Fonte: O autor, 2023.

110.463,62 kg de etanol e consome-se 99.889,47 kg de células de biomassa.

8.4.2 Problema de Otimização A

Nesta subseção, serão apresentadas as informações específicas e os resultados obtidos a partir de cada uma das simulações de otimização feitas sobre o Problema de Otimização A.

8.4.2.1 Simulação com o NSGA-II

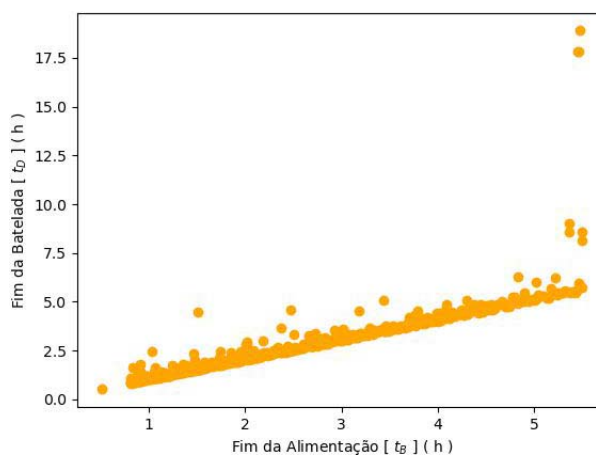
A simulação que buscou solucionar o Problema de Otimização A através do método NSGA-II levou 14 horas e 58 minutos e 49 segundos para atingir as 50 gerações. O método de otimização convergiu na 9ª geração, encontrando 600 soluções não dominadas.

As soluções obtidas nesta simulação estão descritas, detalhadas e ilustradas nas subseções abaixo.

8.4.2.1.1 Conjunto Ótimo de Pareto

O Gráfico 34 mostra os otimizadores de Pareto encontrados nesta simulação de otimização, nos quais consiste o conjunto ótimo de Pareto estimado.

Gráfico 34 - Conjunto ótimo de Pareto do Problema A do Estudo de Caso 2 obtido pelo NSGA-II.

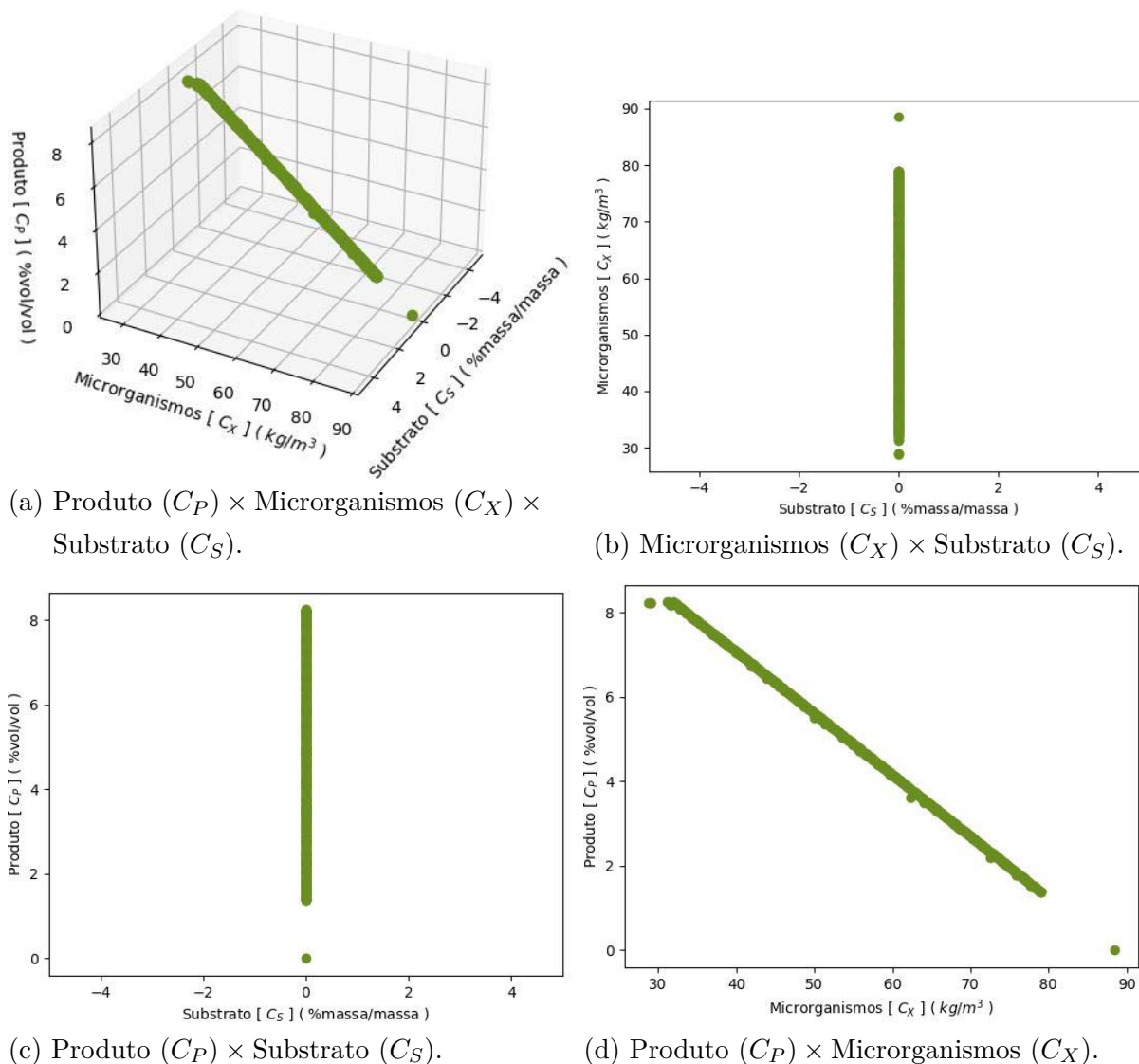


Fonte: O autor, 2023.

8.4.2.1.2 Fronteira de Pareto

O Gráfico 35 apresenta a frente de Pareto do problema de otimização em questão, que foi estimada nesta simulação, acompanhada de suas respectivas projeções.

Gráfico 35 - Fronteira de Pareto do Problema A do Estudo de Caso 2 obtida pelo NSGA-II e suas projeções.



Fonte: O autor, 2023.

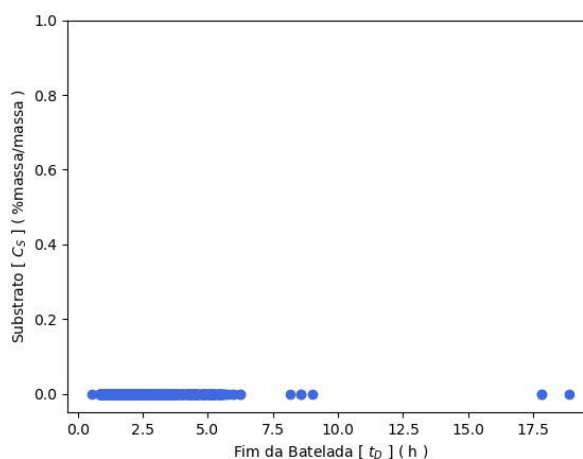
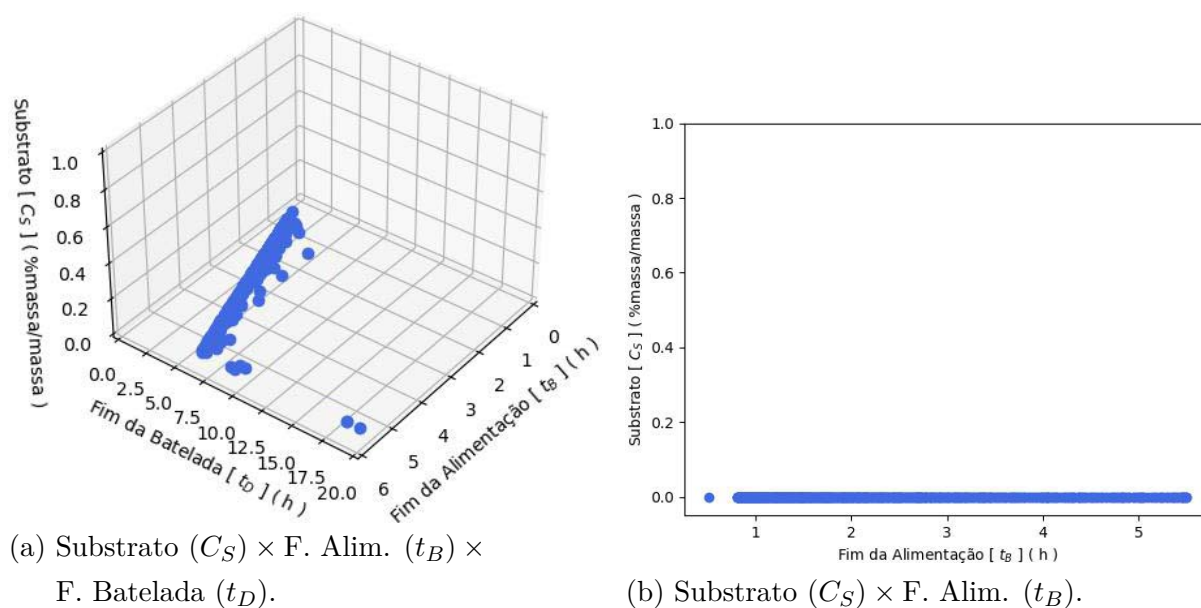
8.4.2.1.3 Relação entre os Otimizadores de Pareto e os Objetivos

A seguir, serão apresentadas as correspondências dos pontos do conjunto ótimo de Pareto, mostrado no Gráfico 34, com cada um dos objetivos do problema de otimização

em questão.

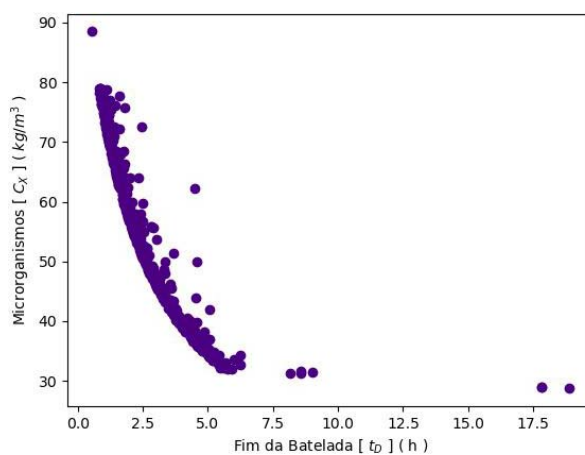
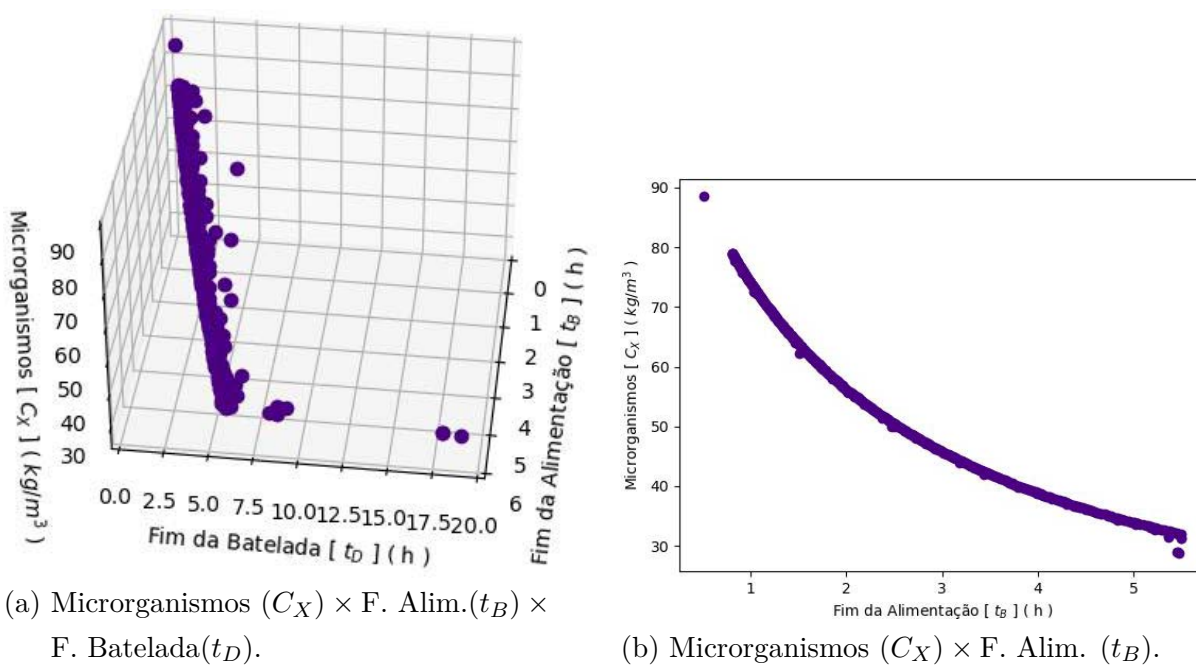
O Gráfico 36 mostra a concentração de substrato relacionada a cada um dos otimizadores do conjunto ótimo de Pareto, o Gráfico 37 enuncia a concentração de células de biomassa relacionada a cada um dos otimizadores de Pareto encontrados e o Gráfico 38 ilustra a concentração de produto relacionada a cada um dos elementos do conjunto ótimo de Pareto.

Gráfico 36 - Concentrações de açúcares correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo NSGA-II.



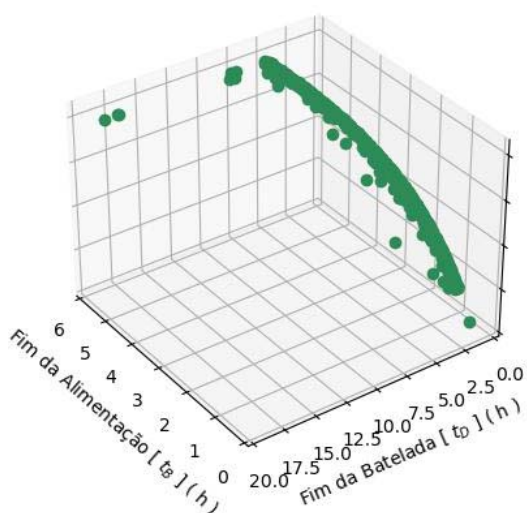
Fonte: O autor, 2023.

Gráfico 37 - Concentrações de levedura correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo NSGA-II.

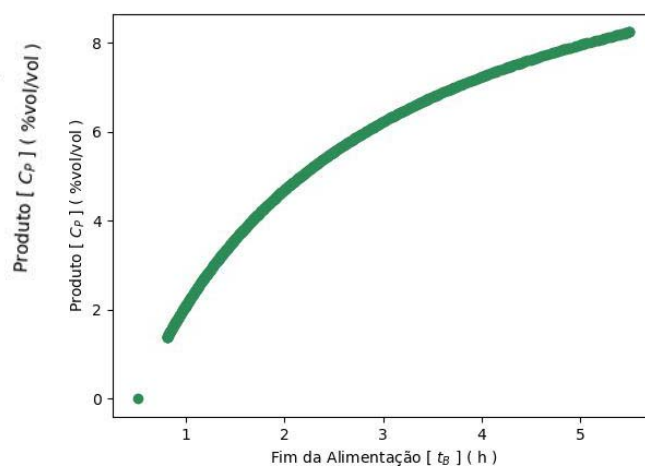


Fonte: O autor, 2023.

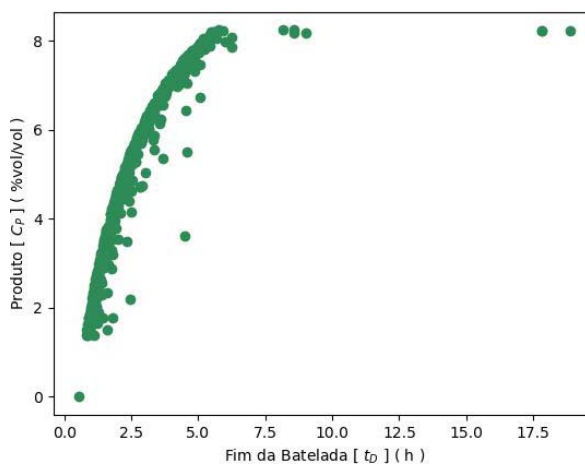
Gráfico 38 - Concentrações de etanol correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo NSGA-II.



(a) Produto (C_P) \times F. Alim. (t_B) \times F. Batelada (t_D).



(b) Produto (C_P) \times F. Alim. (t_B).



(c) Produto (C_P) \times F. Batelada (t_D).

Fonte: O autor, 2023.

8.4.2.2 Simulação com o AGE-MOEA

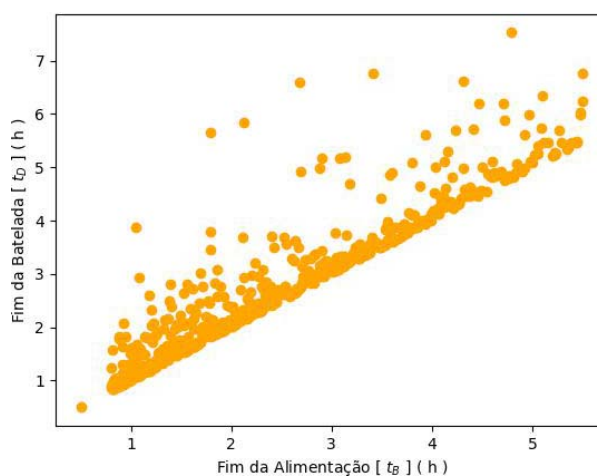
A simulação que buscou solucionar o Problema de Otimização A através do método AGE-MOEA teve uma duração de 8 horas e 22 minutos e 2 segundos para atingir a 50ª geração. O algoritmo convergiu na 9ª geração, encontrando 600 soluções não dominadas.

As soluções obtidas nesta simulação estão descritas, detalhadas e ilustradas nas subseções a seguir.

8.4.2.2.1 Conjunto Ótimo de Pareto

O Gráfico 39 ilustra os otimizadores de Pareto encontrados na presente simulação de otimização, nos quais consiste o conjunto ótimo de Pareto estimado.

Gráfico 39 - Conjunto ótimo de Pareto do Problema A do Estudo de Caso 2 obtido pelo AGE-MOEA.

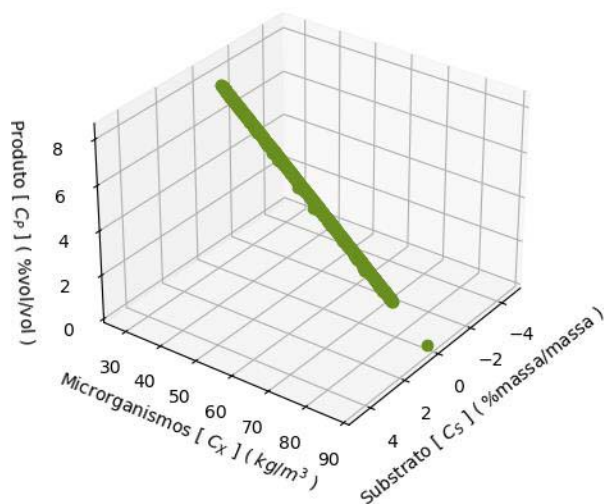


Fonte: O autor, 2023.

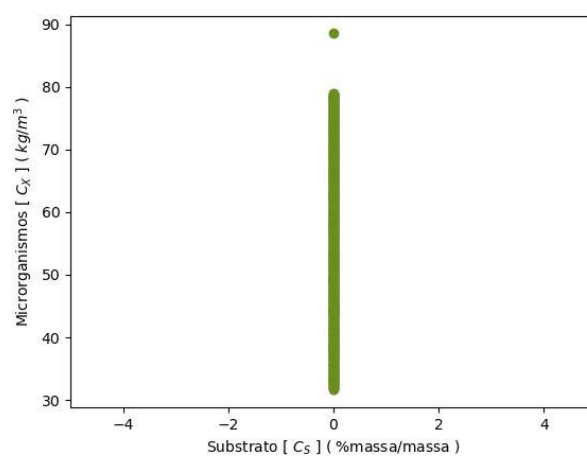
8.4.2.2.2 Fronteira de Pareto

O Gráfico 40 apresenta a fronteira de Pareto do problema de otimização em questão, que foi estimada nesta simulação, acompanhada de suas respectivas projeções.

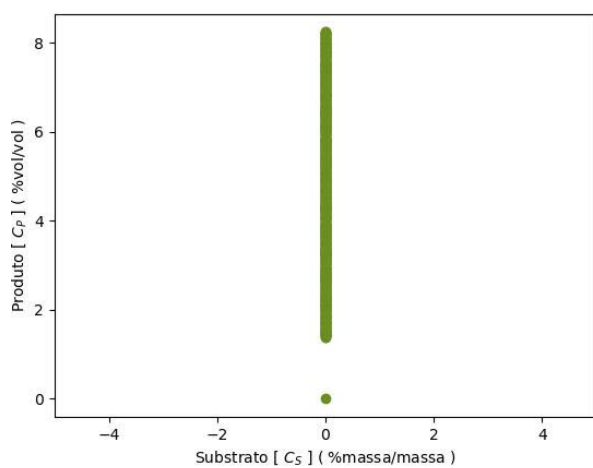
Gráfico 40 - Fronteira de Pareto do Problema A do Estudo de Caso 2 obtida pelo AGE-MOEA e suas projeções.



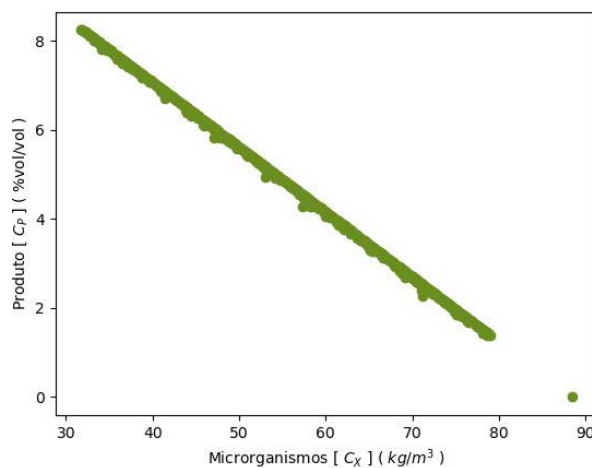
(a) Produto (C_P) \times Microorganismos (C_X) \times Substrato (C_S).



(b) Microorganismos (C_X) \times Substrato (C_S).



(c) Produto (C_P) \times Substrato (C_S).



(d) Produto (C_P) \times Microorganismos (C_X).

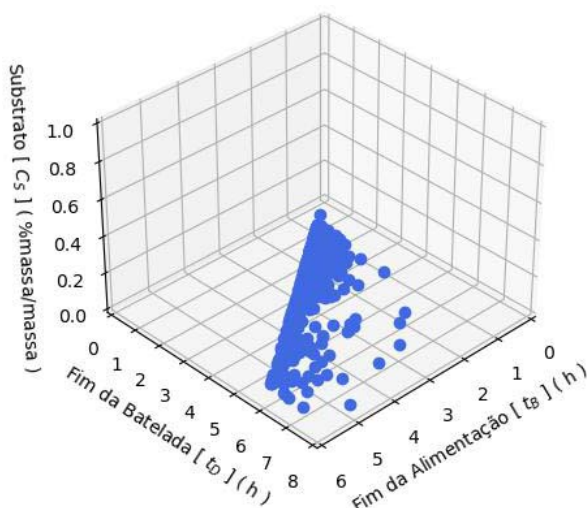
Fonte: O autor, 2023.

8.4.2.2.3 Relação entre os Otimizadores de Pareto e os Objetivos

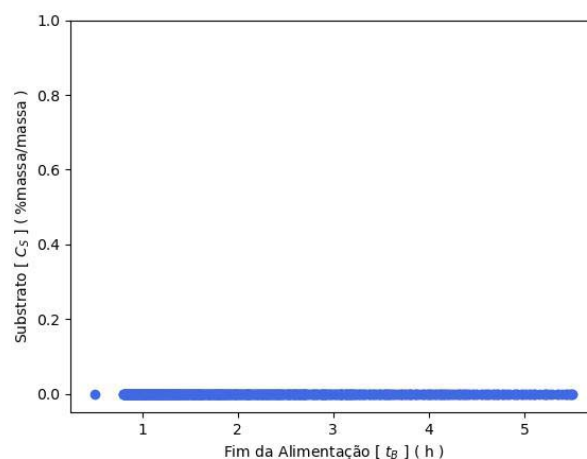
A seguir, serão apresentadas as correspondências dos otimizadores de Pareto, mostrados no Gráfico 39, com cada um dos objetivos do problema de otimização aqui considerado.

O Gráfico 41 apresenta a concentração de substrato relacionada a cada um dos pontos do conjunto ótimo de Pareto, o Gráfico 42 enuncia a concentração de células de biomassa relacionada a cada um dos otimizadores de Pareto encontrados e o Gráfico 43 ilustra a concentração de produto relacionada a cada um dos elementos do conjunto ótimo de Pareto.

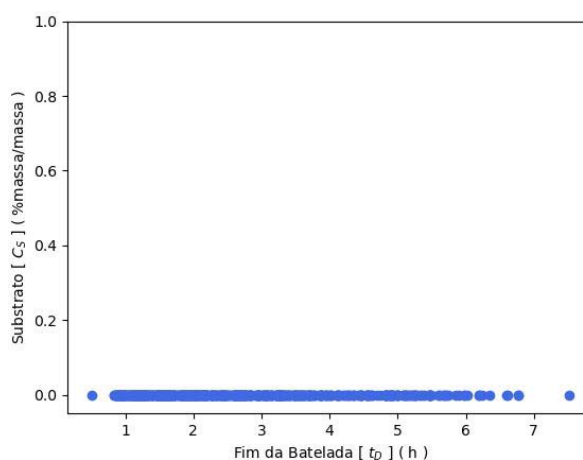
Gráfico 41 - Concentrações de açúcares correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo AGE-MOEA.



(a) Substrato (C_S) \times F. Alim. (t_B) \times F. Batelada (t_D).



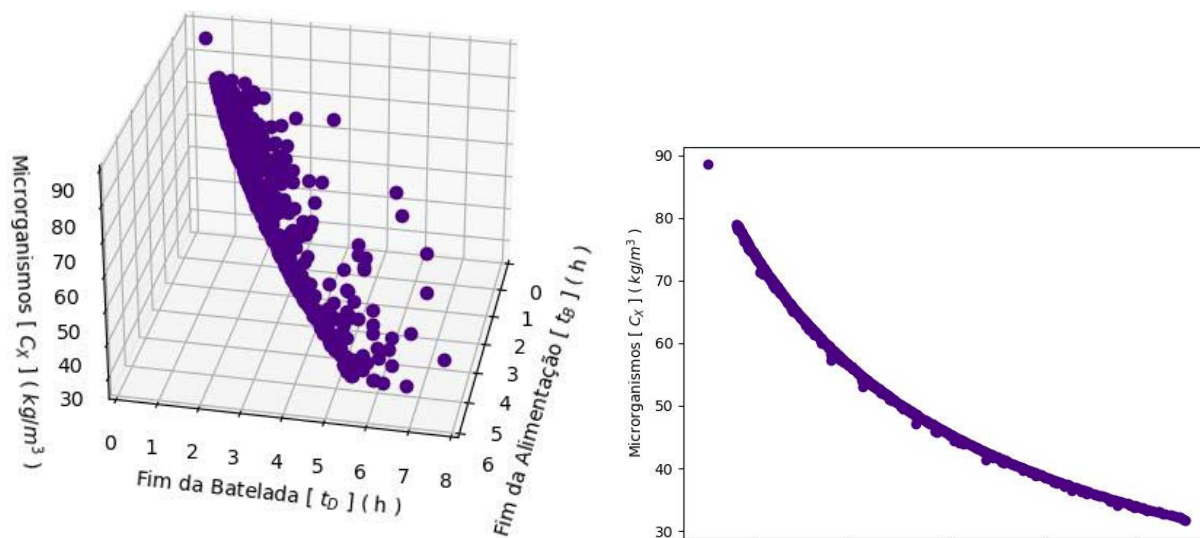
(b) Substrato (C_S) \times F. Alim. (t_B).



(c) Substrato (C_S) \times F. Batelada (t_D).

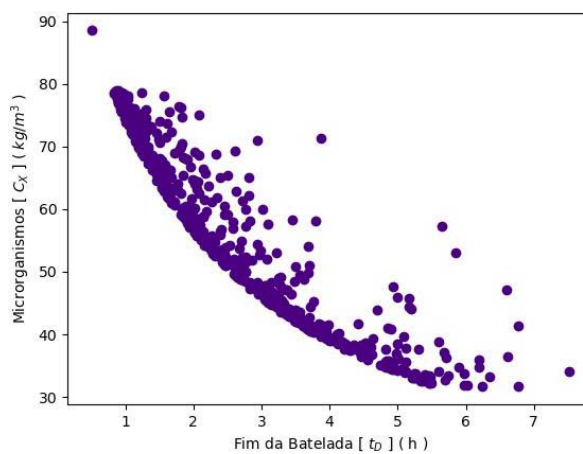
Fonte: O autor, 2023.

Gráfico 42 - Concentrações de levedura correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo AGE-MOEA.



(a) Microrganismos(C_X) \times F. Alim.(t_B) \times F. Batelada(t_D).

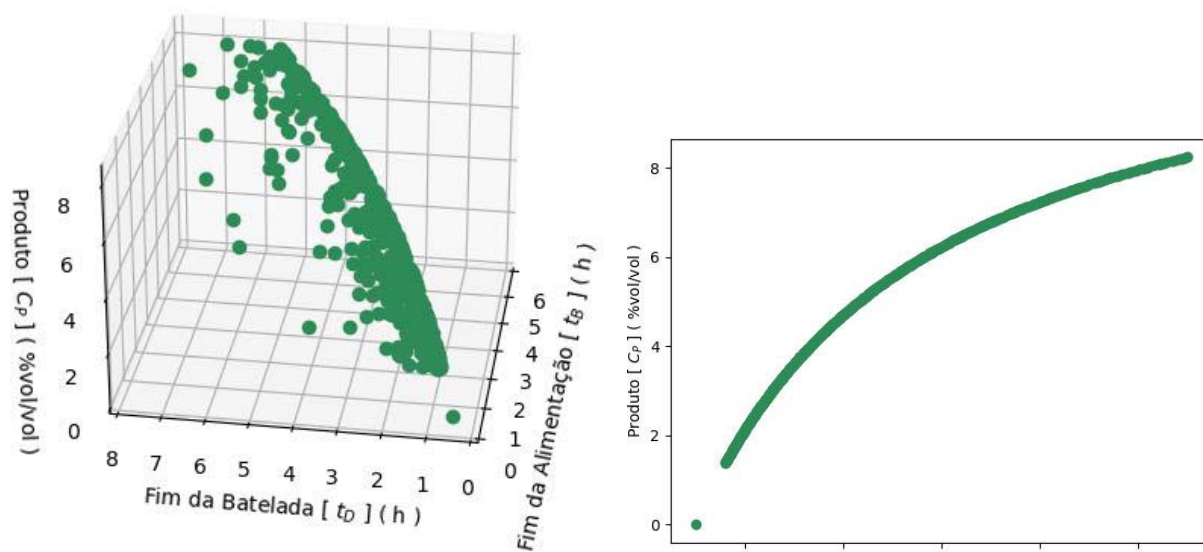
(b) Microrganismos (C_X) \times F. Alim. (t_B).



(c) Microrganismos (C_X) \times F. Batelada (t_D).

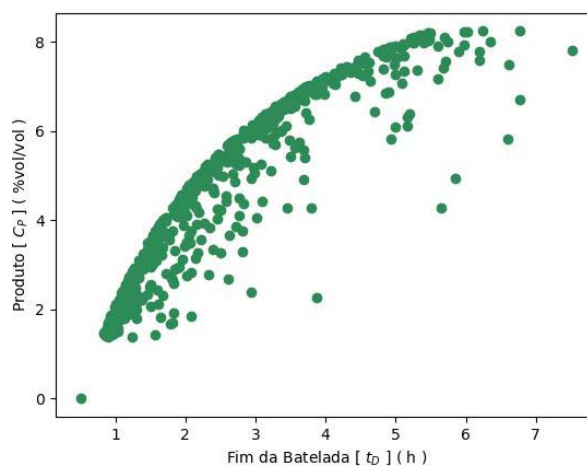
Fonte: O autor, 2023.

Gráfico 43 - Concentrações de etanol correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo AGE-MOEA.



(a) Produto (C_P) \times F. Alim. (t_B) \times
F. Batelada (t_D).

(b) Produto (C_P) \times F. Alim. (t_B).



(c) Produto (C_P) \times F. Batelada (t_D).

Fonte: O autor, 2023.

8.4.2.3 Simulação com o C-TAEA

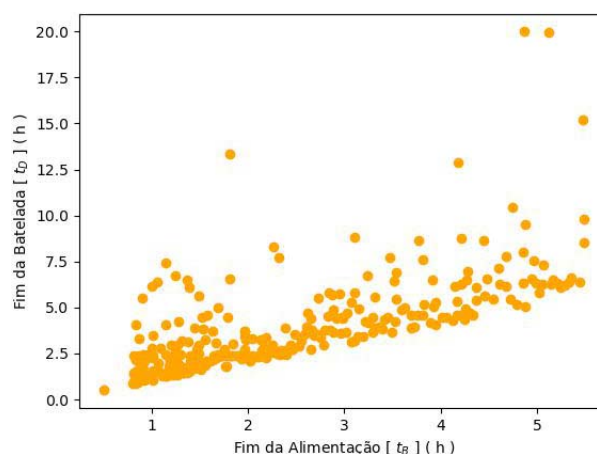
A simulação que buscou solucionar o Problema de Otimização A através do método C-TAEA demorou 7 horas e 36 minutos e 2 segundos para atingir as 50 gerações. O algoritmo convergiu na 50^a geração, encontrando 298 soluções não dominadas.

As soluções obtidas nesta simulação estão descritas, detalhadas e ilustradas nas subseções a seguir.

8.4.2.3.1 Conjunto Ótimo de Pareto

O Gráfico 44 ilustra os otimizadores de Pareto encontrados na presente simulação de otimização, nos quais consiste o conjunto ótimo de Pareto estimado.

Gráfico 44 - Conjunto ótimo de Pareto do Problema A do Estudo de Caso 2 obtido pelo C-TAEA.

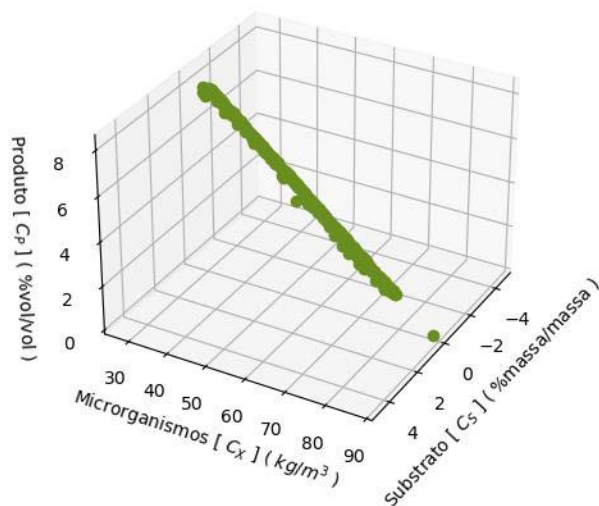


Fonte: O autor, 2023.

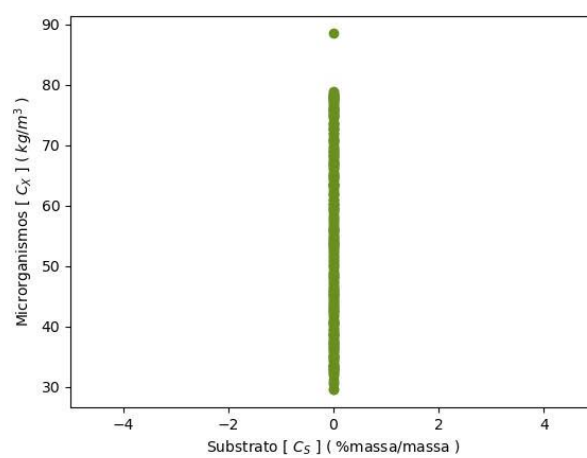
8.4.2.3.2 Fronteira de Pareto

O Gráfico 45 ilustra a fronteira de Pareto do problema de otimização em questão, que foi estimada nesta simulação, acompanhada de suas respectivas projeções.

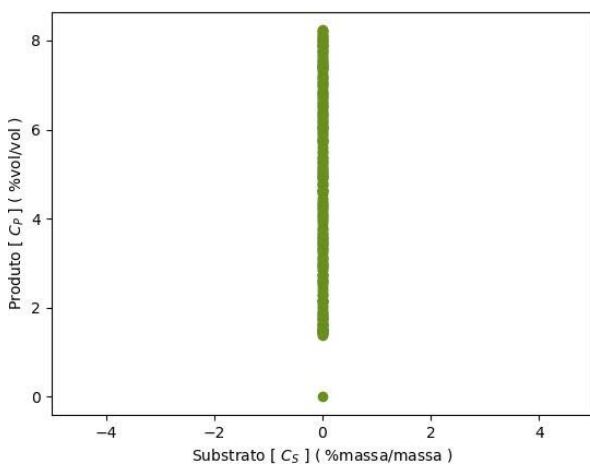
Gráfico 45 - Fronteira de Pareto do Problema A do Estudo de Caso 2 obtida pelo C-TAEA e suas projeções.



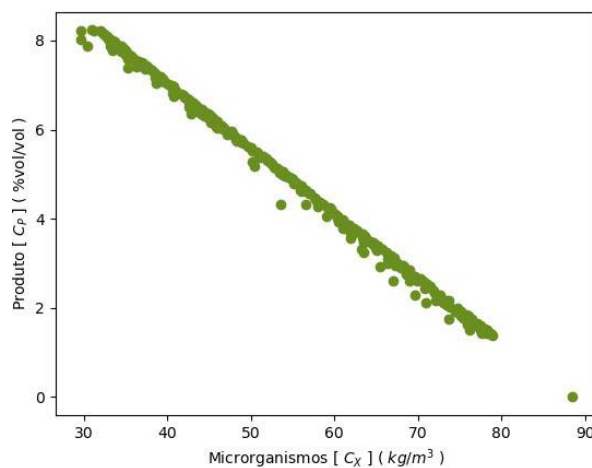
(a) Produto (C_P) \times Microorganismos (C_X) \times Substrato (C_S).



(b) Microorganismos (C_X) \times Substrato (C_S).



(c) Produto (C_P) \times Substrato (C_S).



(d) Produto (C_P) \times Microorganismos (C_X).

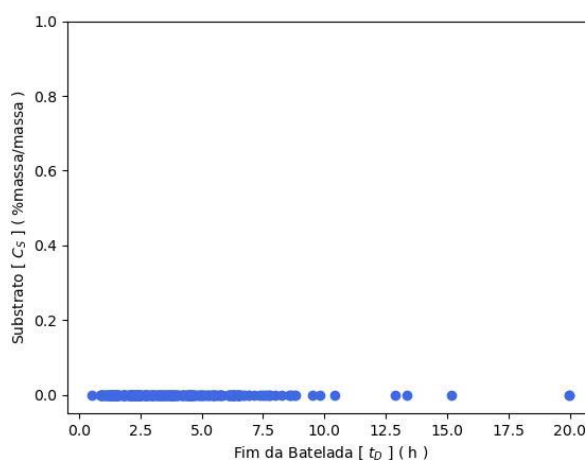
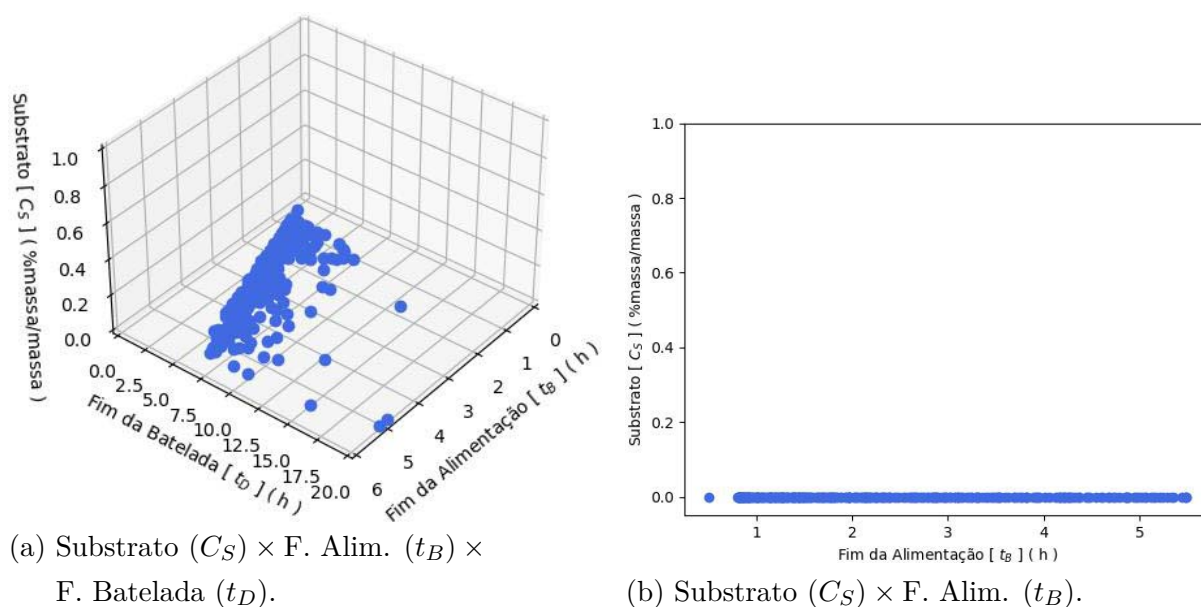
Fonte: O autor, 2023.

8.4.2.3.3 Relação entre os Otimizadores de Pareto e os Objetivos

Abaixo, estão mostradas as correspondências dos pontos do conjunto ótimo de Pareto, mostrado no Gráfico 44, com cada um dos objetivos do problema de otimização considerado.

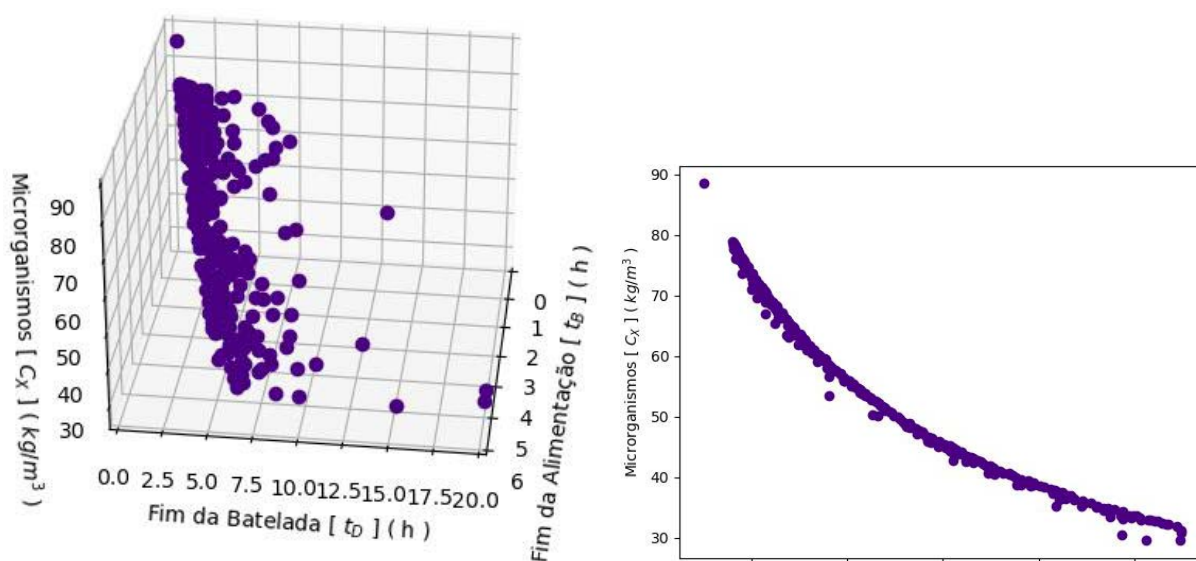
O Gráfico 46 mostra a concentração de substrato relacionada a cada um dos otimizadores de Pareto encontrados, o Gráfico 47 enuncia a concentração de células de biomassa relacionada a cada um dos otimizadores de Pareto encontrados e o Gráfico 48 ilustra a concentração de produto relacionada a cada um dos elementos do conjunto ótimo de Pareto.

Gráfico 46 - Concentrações de açúcares correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo C-TAEA.



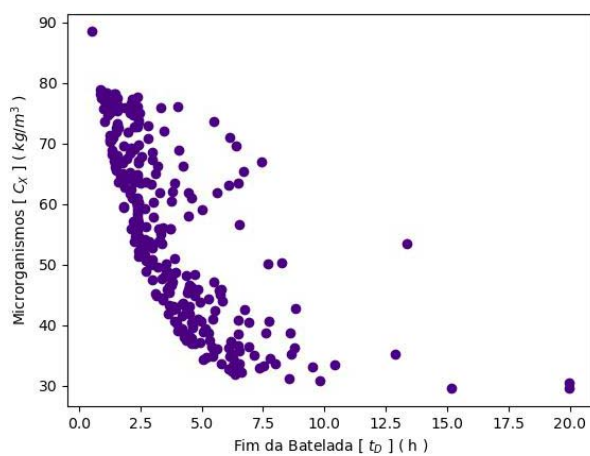
Fonte: O autor, 2023.

Gráfico 47 - Concentrações de levedura correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo C-TAEA.



(a) $\text{Microorganismos}(C_X) \times \text{F. Alim.}(t_B) \times \text{F. Batelada}(t_D)$.

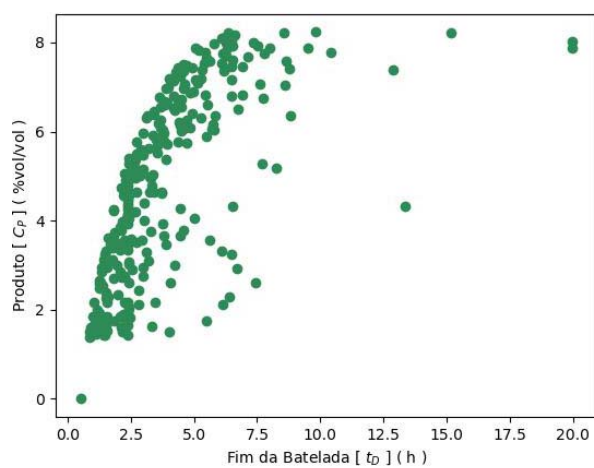
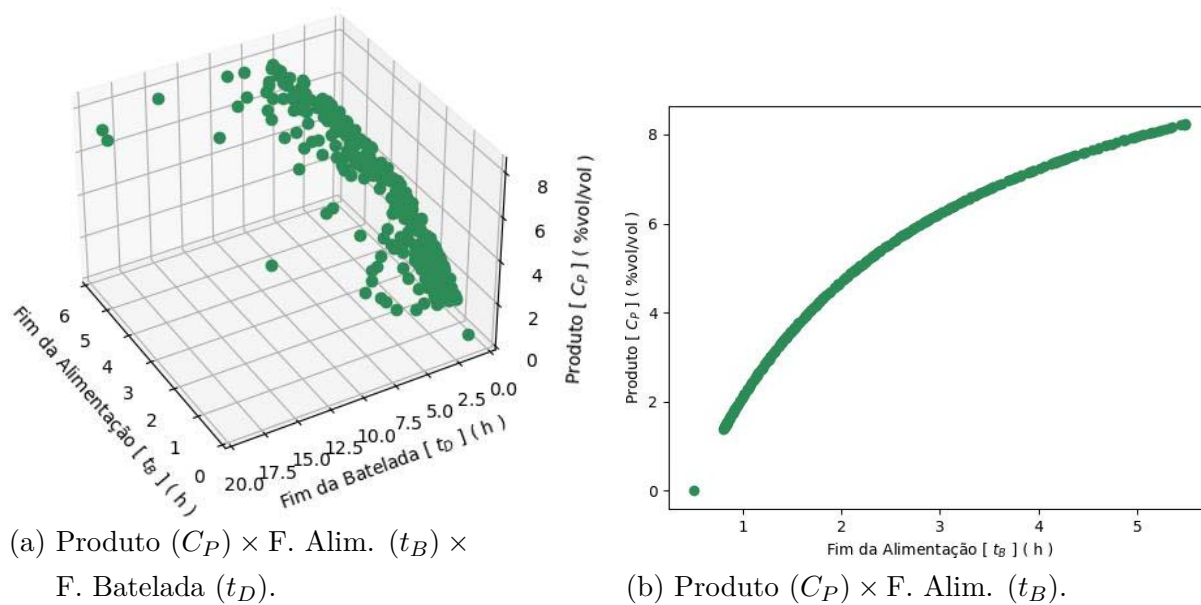
(b) $\text{Microorganismos}(C_X) \times \text{F. Alim.}(t_B)$.



(c) $\text{Microorganismos}(C_X) \times \text{F. Batelada}(t_D)$.

Fonte: O autor, 2023.

Gráfico 48 - Concentrações de etanol correspondentes aos otimizadores de Pareto do Problema A do Estudo de Caso 2 obtidos pelo C-TAEA.



Fonte: O autor, 2023.

8.4.2.4 Soluções Notáveis

Considerando os ótimos de Pareto obtidos em todas as simulações de otimização realizadas sobre este problema, foram elencadas as soluções que apresentam o melhor valor atingido para cada um dos objetivos, bem como, soluções que equilibram todos os objetivos.

A Tabela 13 expressa cada uma das soluções elencadas, bem como o método usado em sua obtenção.

Tabela 13 - Soluções Notáveis do Problema A do Estudo de Caso 2.

Solução	Variáveis		Objetivos		
	t_B	t_D	$C_S(t_H)$	$C_X(t_H)$	$C_P(t_H)$
AGE-MOEA					
E2P1M1S1	0,50005	0,50025	0,00000	88,51814	0,00000
E2P1M1S2	5,50276	6,76941	0,00000	31,64833	8,23885
E2P1M1S3	1,72389	1,88808	0,00000	60,14030	4,12747
C-TAEA					
E2P1M2S1	0,50000	0,50023	0,00000	88,51816	0,00000
E2P1M2S2	5,49189	9,83125	0,00000	30,89006	8,23303
E2P1M2S2	1,72403	2,38839	0,00000	59,88885	4,12781
NSGA-II					
E2P1M3S1	0,50468	0,51827	0,00000	88,50490	0,00000
E2P1M3S2	5,50096	8,58063	0,00000	31,18210	8,23789
E2P1M3S3	1,72024	1,79382	0,00000	60,24128	4,11934

Fonte: O autor, 2023.

8.4.3 Problema de Otimização B

Nesta subseção, são mostradas as informações específicas e os resultados obtidos a partir das simulações de otimização que foram realizadas sobre o Problema de Otimização B.

8.4.3.1 Simulação com o NSGA-II

A simulação que buscou solucionar o Problema de Otimização B através do método NSGA-II demorou 12 horas e 06 minutos e 22 segundos para atingir as 50 gerações. O algoritmo convergiu na 13^a geração, encontrando 600 soluções não dominadas.

As soluções obtidas nesta simulação estão descritas, detalhadas e ilustradas nas subseções a seguir.

8.4.3.1.1 Fronteira de Pareto

O Gráfico 49 apresenta a curva de Pareto do problema de otimização em questão, que foi estimada nesta simulação, acompanhada de suas respectivas projeções.

8.4.3.1.2 Conjunto Ótimo de Pareto

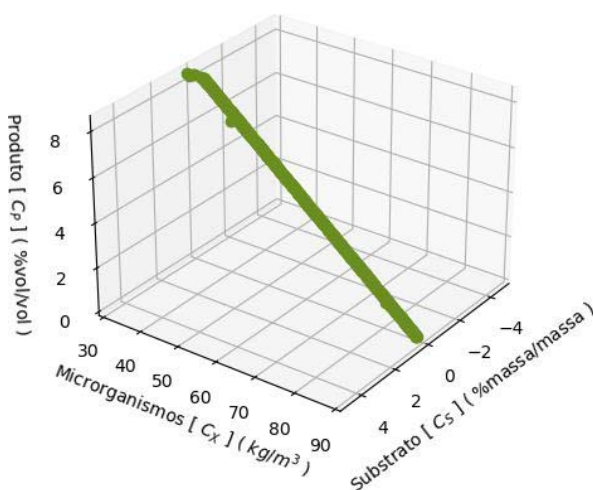
O Gráfico 50 ilustra os otimizadores de Pareto encontrados na presente simulação de otimização, nos quais consiste o conjunto ótimo de Pareto estimado.

8.4.3.1.3 Relação entre os Otimizadores de Pareto e os Objetivos

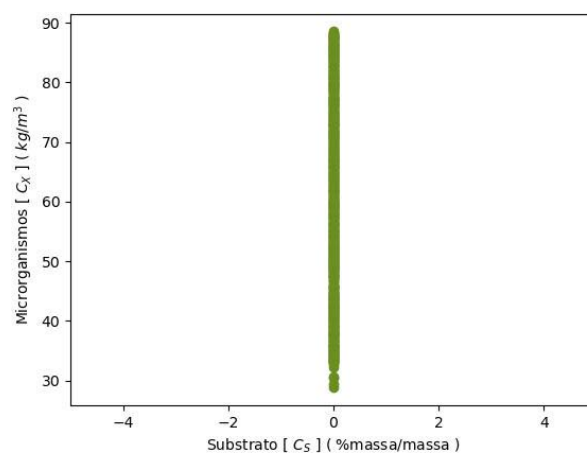
Nesta seção, serão apresentadas as correspondências dos pontos do conjunto ótimo de Pareto mostrado no Gráfico 50 com cada um dos objetivos do problema de otimização considerado.

O Gráfico 51 apresenta a concentração de substrato relacionada a cada um dos pontos do conjunto ótimo de Pareto, o Gráfico 52 enuncia a concentração de células de biomassa relacionada a cada um dos otimizadores de Pareto encontrados e o Gráfico 53 ilustra a concentração de produto relacionada a cada um dos elementos do conjunto ótimo de Pareto.

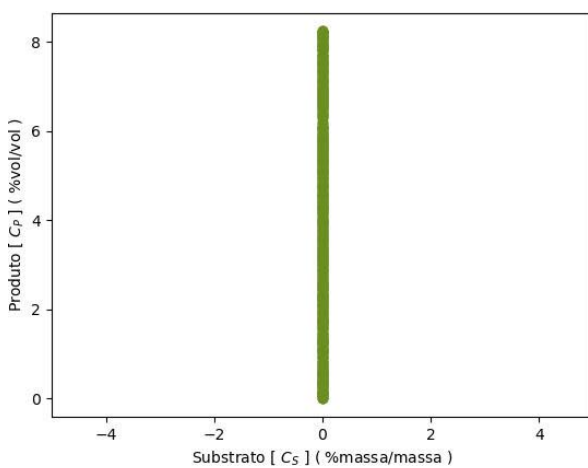
Gráfico 49 - Fronteira de Pareto do Problema B do Estudo de Caso 2 obtida pelo NSGA-II e suas projeções.



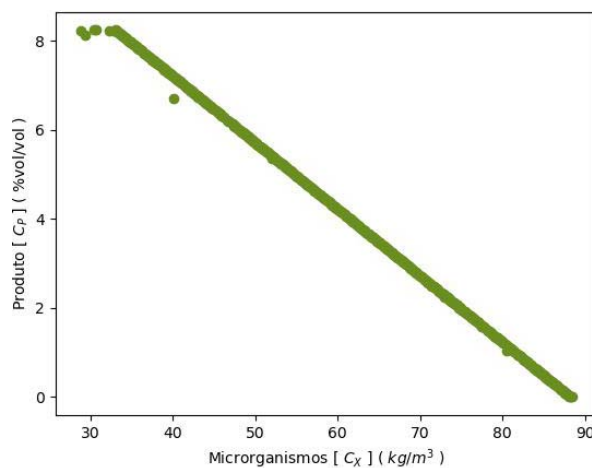
(a) Produto (C_P) \times Microorganismos (C_X) \times Substrato (C_S).



(b) Microorganismos (C_X) \times Substrato (C_S).



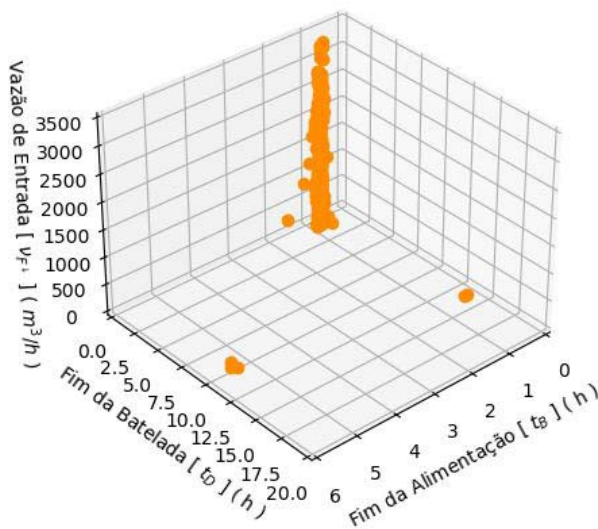
(c) Produto (C_P) \times Substrato (C_S).



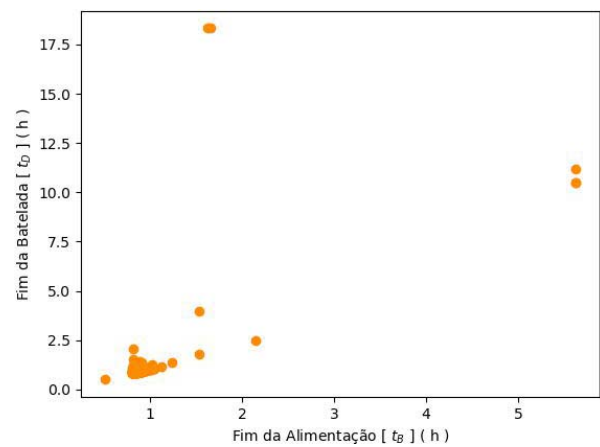
(d) Produto (C_P) \times Microorganismos (C_X).

Fonte: O autor, 2023.

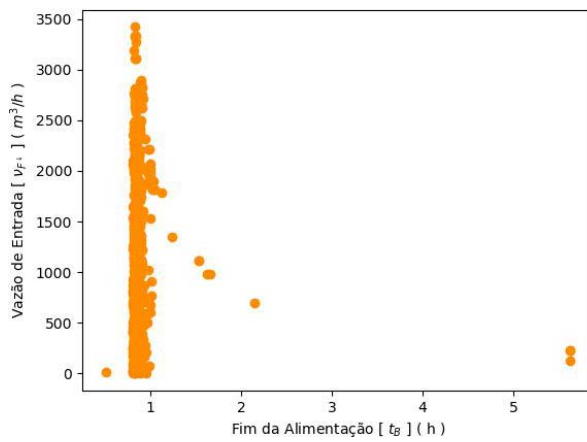
Gráfico 50 - Conjunto ótimo de Pareto do Problema B do Estudo de Caso 2 obtido pelo NSGA-II.



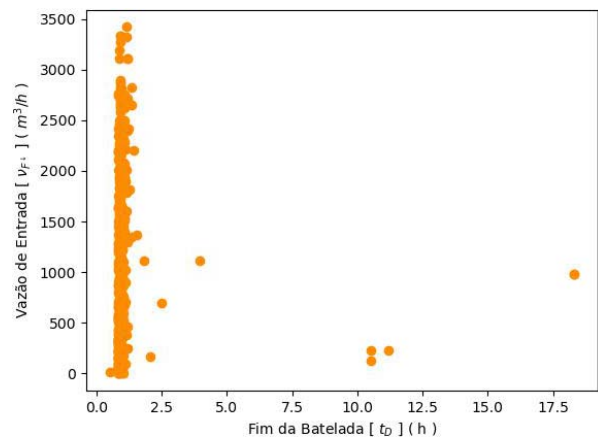
(a) V. Entrada ($\nu_{F\downarrow}$) \times F. Alim. (t_B) \times F. Batelada (t_D).



(b) F. Alim. (t_B) \times F. Batelada (t_D).



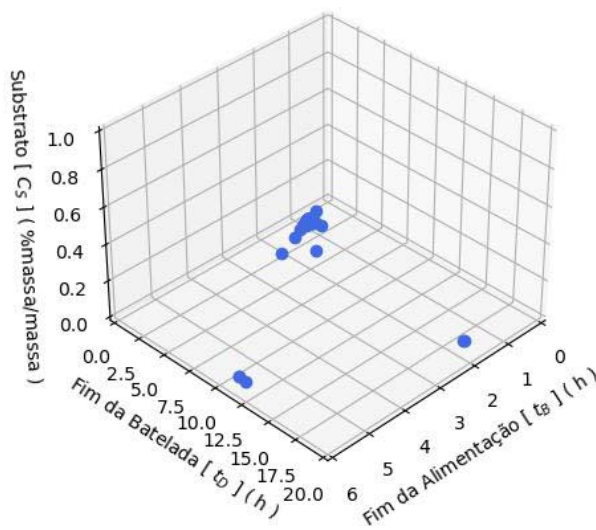
(c) V. Entrada ($\nu_{F\downarrow}$) \times F. Alim. (t_B).



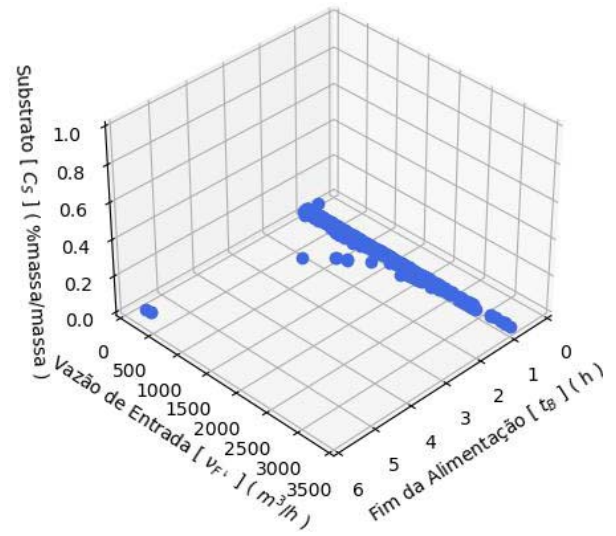
(d) V. Entrada ($\nu_{F\downarrow}$) \times F. Batelada (t_D).

Fonte: O autor, 2023.

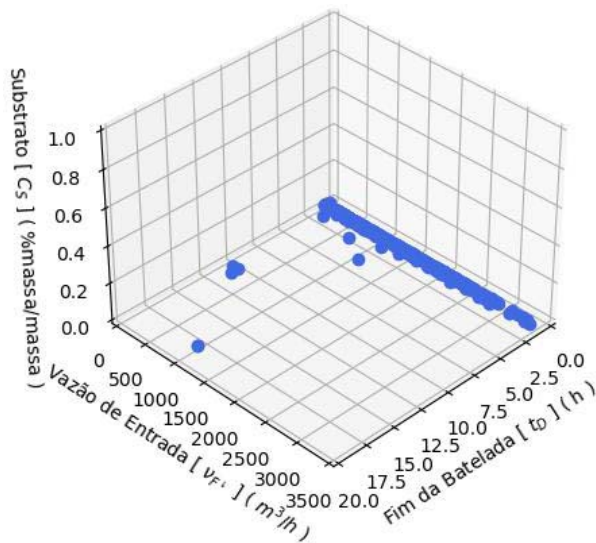
Gráfico 51 - Concentrações de açúcares correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo NSGA-II.



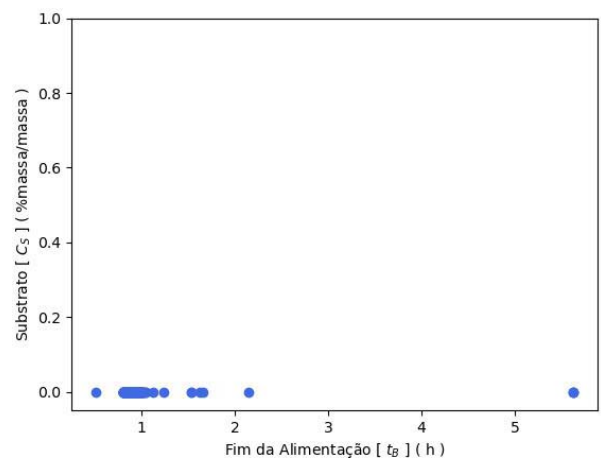
(a) Substrato (C_S) \times F. Alim. (t_B) \times F. Batelada (t_D).



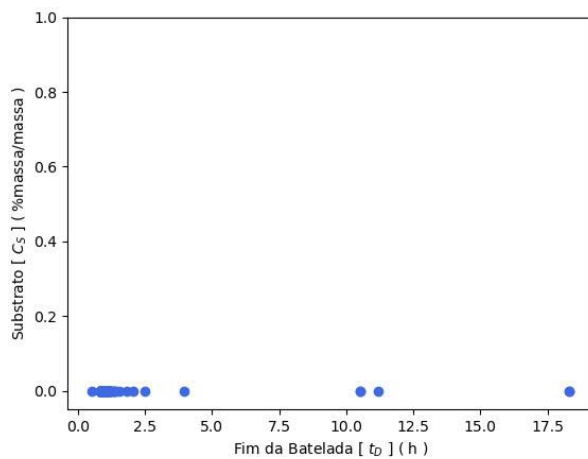
(b) Substrato (C_S) \times V. Entrada ($\nu_{F\downarrow}$) \times F. Alim. (t_B).



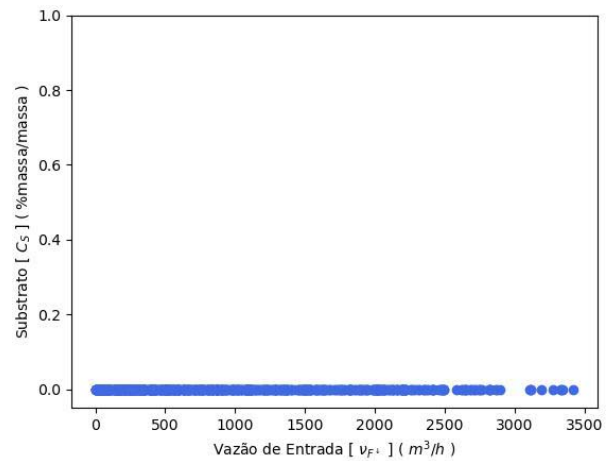
(c) Substrato (C_S) \times V. Entrada ($\nu_{F\downarrow}$) \times F. Batelada (t_D).



(d) Substrato (C_S) \times F. Alim. (t_B).

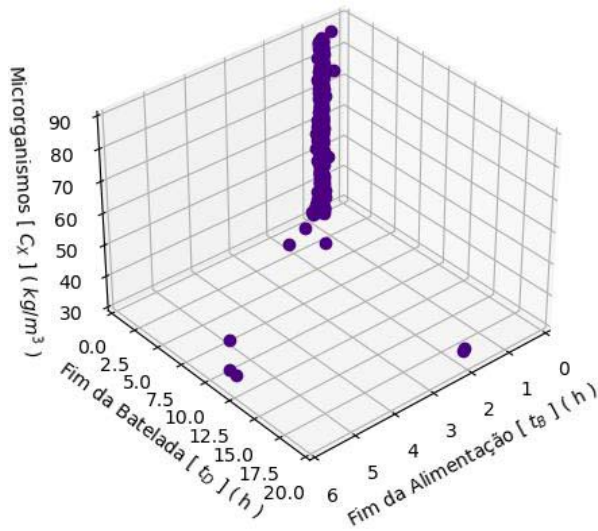


(e) Substrato (C_S) \times F. Batelada (t_D).

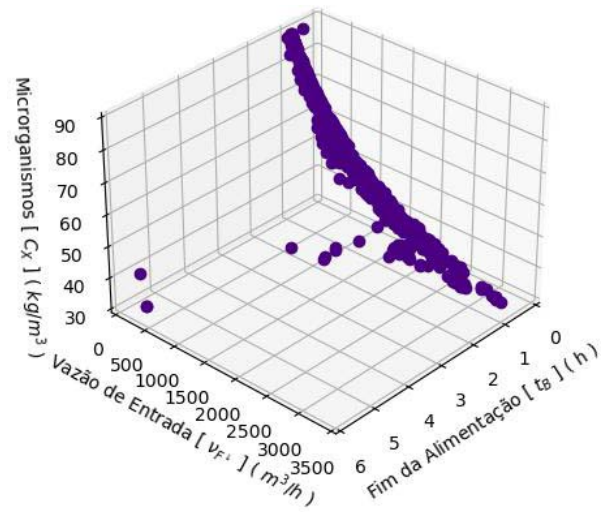


(f) Substrato (C_S) \times V. Entrada ($\nu_{F\downarrow}$).

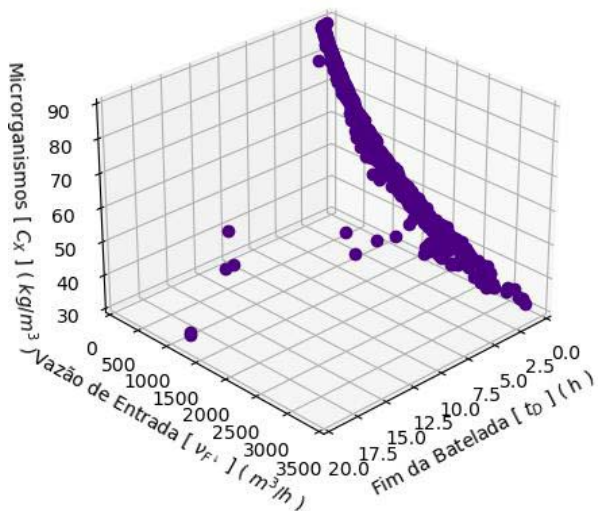
Gráfico 52 - Concentrações de levedura correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo NSGA-II.



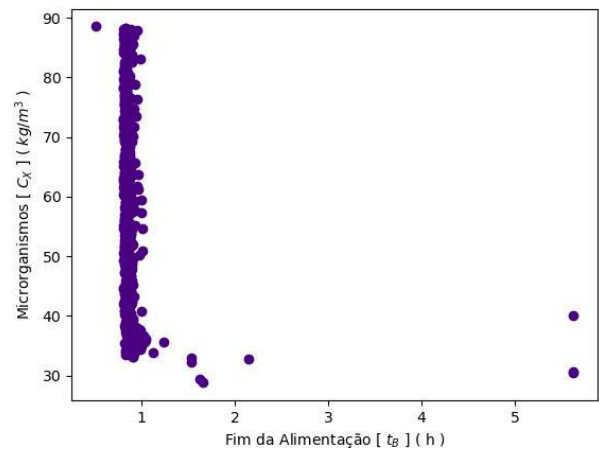
(a) $\text{Microorganismos}(C_X) \times \text{F. Alim.}(t_B) \times \text{F. Batelada}(t_D)$.



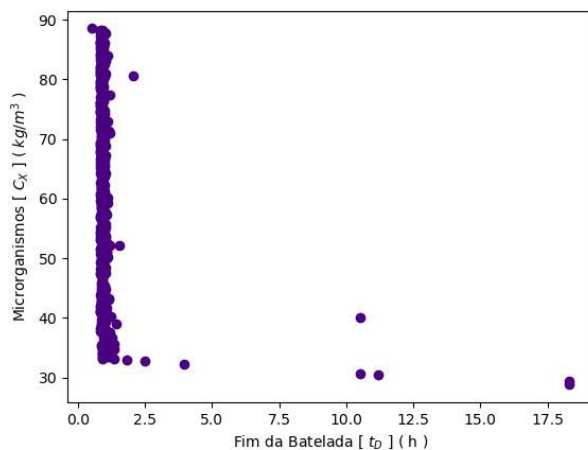
(b) $\text{Microorganismos}(C_X) \times \text{V. Entrada}(v_{F\downarrow}) \times \text{F. Alim.}(t_B)$.



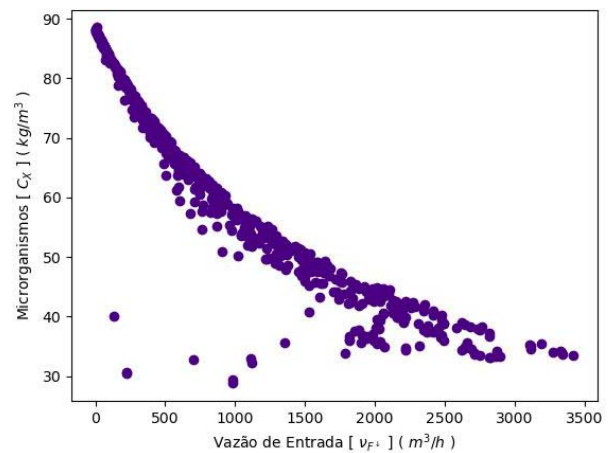
(c) $\text{Microorganismos}(C_X) \times \text{V. Entrada}(v_{F\downarrow}) \times \text{F. Batelada}(t_D)$.



(d) $\text{Microorganismos}(C_X) \times \text{F. Alim.}(t_B)$.

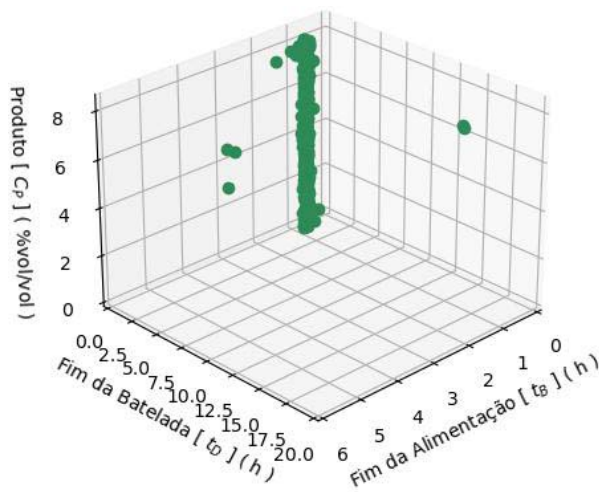


(e) $\text{Microorganismos}(C_X) \times \text{F. Batelada}(t_D)$.

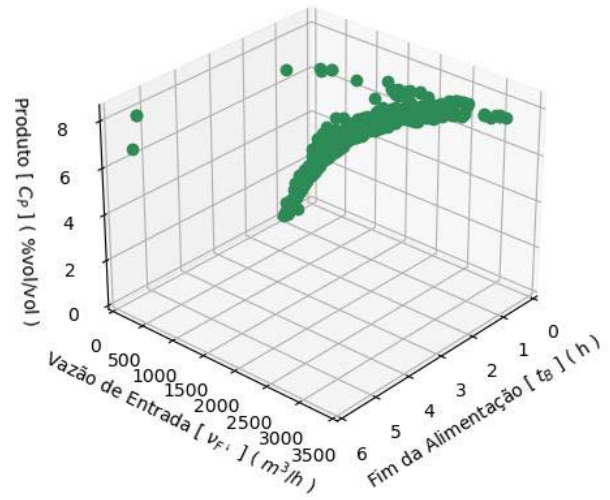


(f) $\text{Microorganismos}(C_X) \times \text{V. Entrada}(v_{F\downarrow})$.

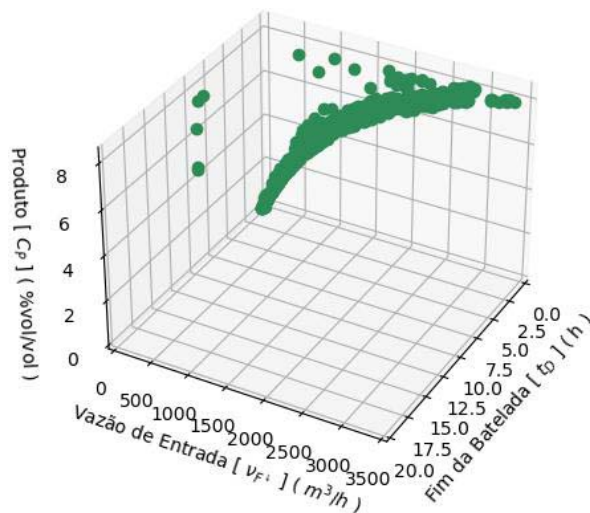
Gráfico 53 - Concentrações de etanol correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo NSGA-II.



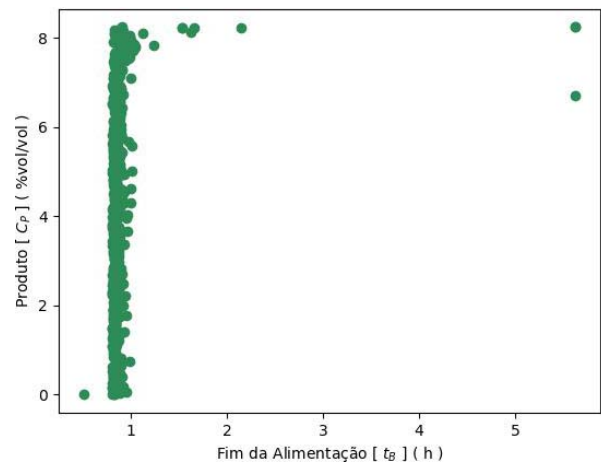
(a) Produto (C_P) \times F. Alim. (t_B) \times F. Batelada (t_D).



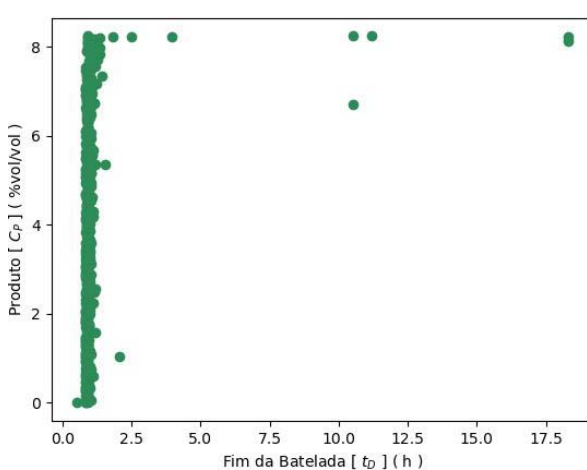
(b) Produto (C_P) \times V. Entrada ($\nu_{F\downarrow}$) \times F. Alim. (t_B).



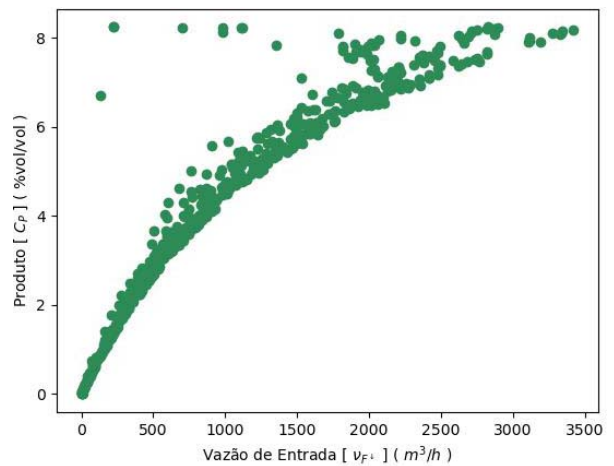
(c) Produto (C_P) \times V. Entrada ($\nu_{F\downarrow}$) \times F. Batelada (t_D).



(d) Produto (C_P) \times F. Alim. (t_B).



(e) Produto (C_P) \times F. Batelada (t_D).



(f) Produto (C_P) \times V. Entrada ($\nu_{F\downarrow}$).

8.4.3.2 Simulação com o AGE-MOEA

A simulação que buscou solucionar o Problema de Otimização B através do método AGE-MOEA demorou 15 horas e 13 minutos e 21 segundos para atingir as 50 gerações. O algoritmo convergiu na 11^a geração, encontrando 600 soluções não dominadas.

As soluções obtidas nesta simulação estão descritas, detalhadas e ilustradas nas subseções a seguir.

8.4.3.2.1 Fronteira de Pareto

O Gráfico 54 apresenta a frente de Pareto do problema de otimização em questão que foi estimada nesta simulação, acompanhada de suas respectivas projeções.

8.4.3.2.2 Conjunto Ótimo de Pareto

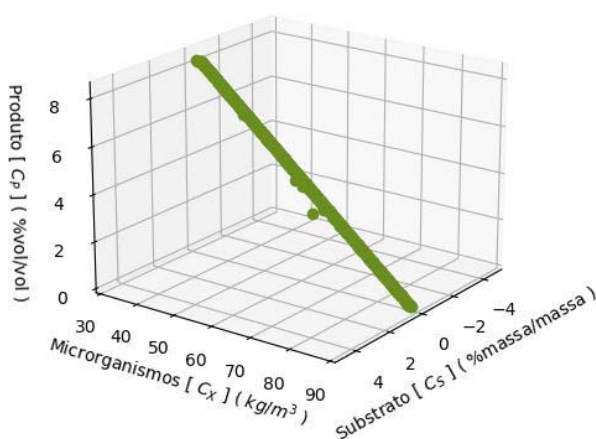
O Gráfico 55 ilustra os otimizadores de Pareto encontrados na presente simulação de otimização, nos quais consiste o conjunto ótimo de Pareto estimado.

8.4.3.2.3 Relação entre os Otimizadores de Pareto e os Objetivos

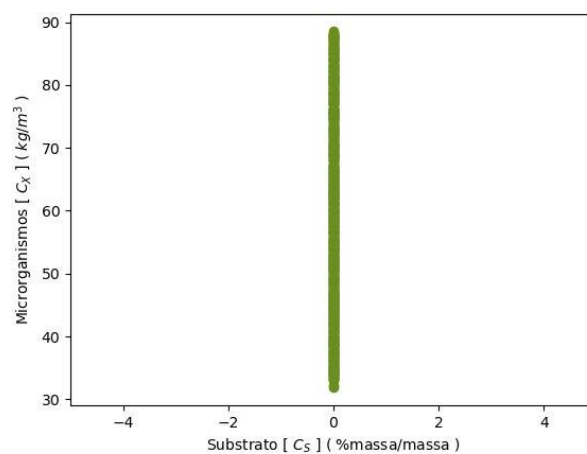
A seguir, estão apresentadas as correspondências dos pontos do conjunto ótimo de Pareto, mostrado no Gráfico 55, com cada objetivo do problema de otimização considerado.

O Gráfico 56 apresenta a concentração de substrato relacionada a cada um dos pontos do conjunto ótimo de Pareto, o Gráfico 57 enuncia a concentração de células de biomassa relacionada a cada um dos otimizadores de Pareto encontrados e o Gráfico 58 ilustra a concentração de produto relacionada a cada um dos elementos do conjunto ótimo de Pareto.

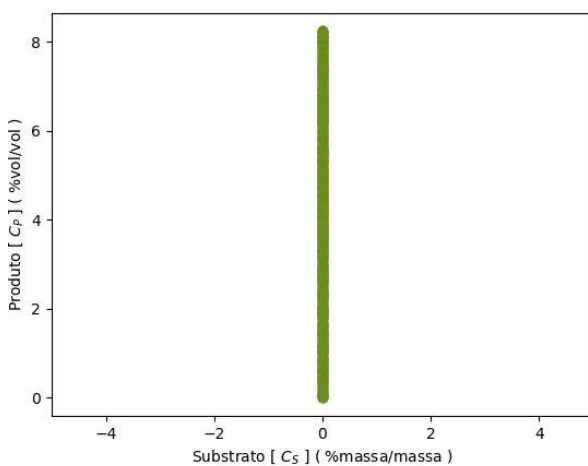
Gráfico 54 - Fronteira de Pareto do Problema B do Estudo de Caso 2 obtida pelo AGE-MOEA e suas projeções.



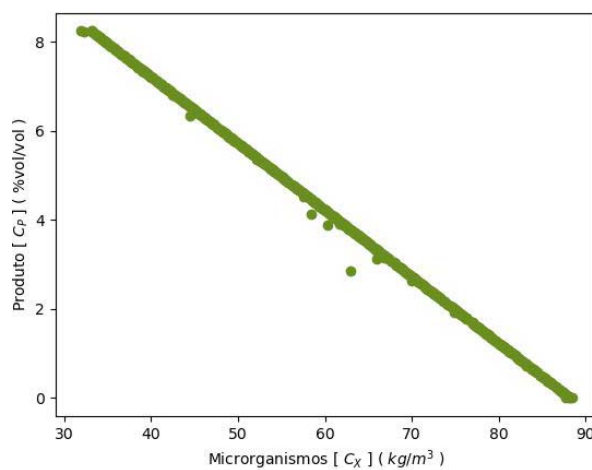
(a) Produto (C_P) \times Microorganismos (C_X) \times Substrato (C_S).



(b) Microorganismos (C_X) \times Substrato (C_S).



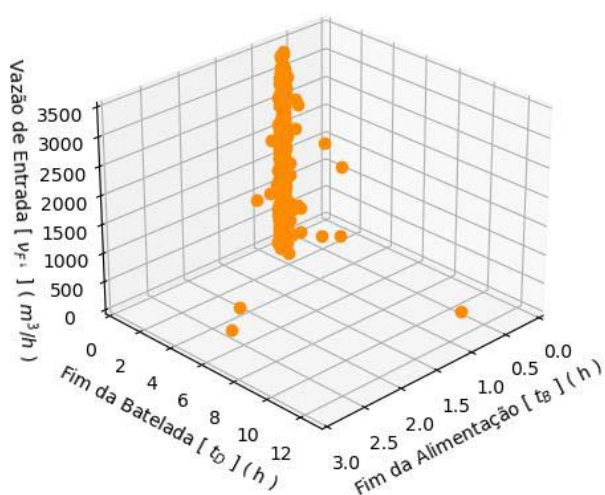
(c) Produto (C_P) \times Substrato (C_S).



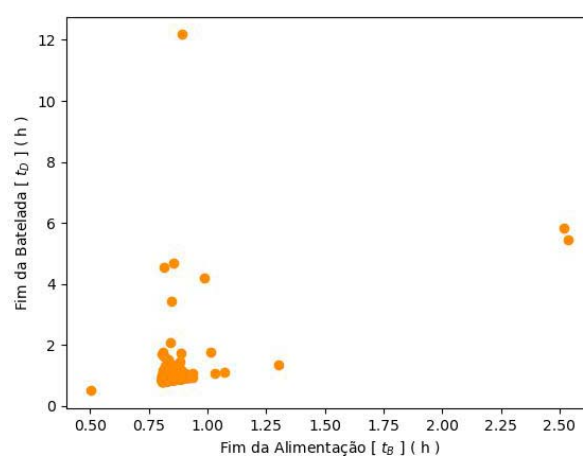
(d) Produto (C_P) \times Microorganismos (C_X).

Fonte: O autor, 2023.

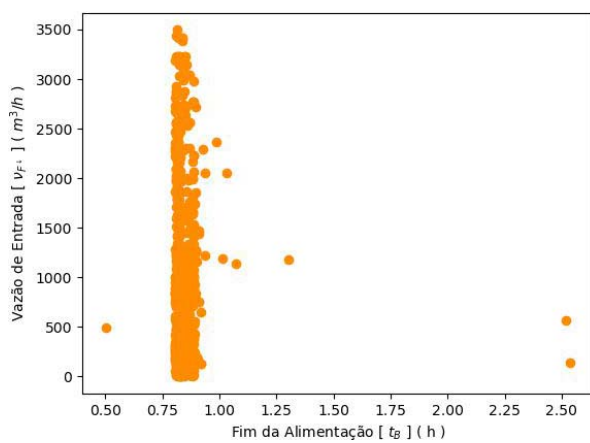
Gráfico 55 - Conjunto ótimo de Pareto do Problema B do Estudo de Caso 2 obtido pelo AGE-MOEA.



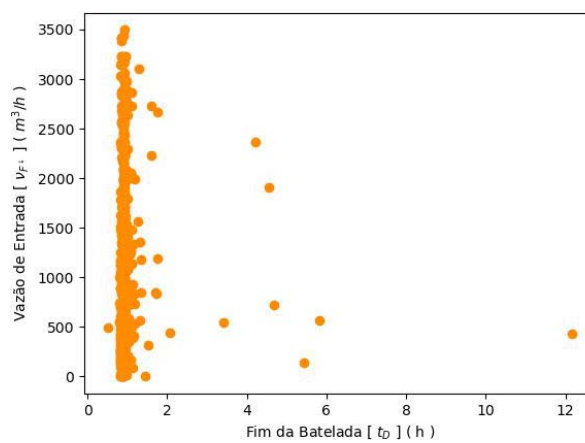
(a) $V. \text{ Entrada } (\nu_{F\downarrow}) \times F. \text{ Alim. } (t_B) \times F. \text{ Batelada } (t_D).$



(b) $F. \text{ Alim. } (t_B) \times F. \text{ Batelada } (t_D).$



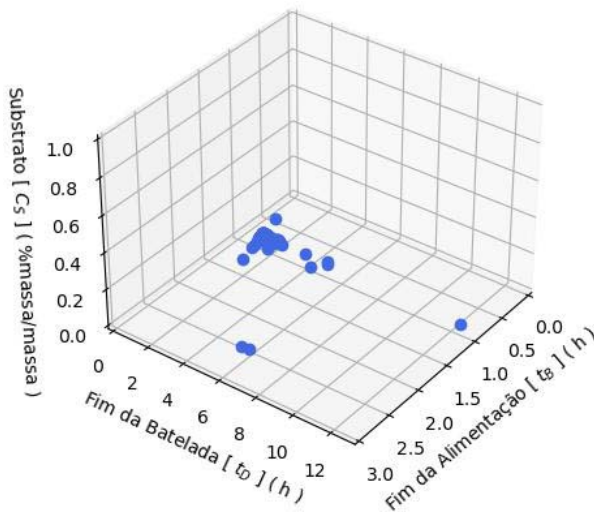
(c) $V. \text{ Entrada } (\nu_{F\downarrow}) \times F. \text{ Alim. } (t_B).$



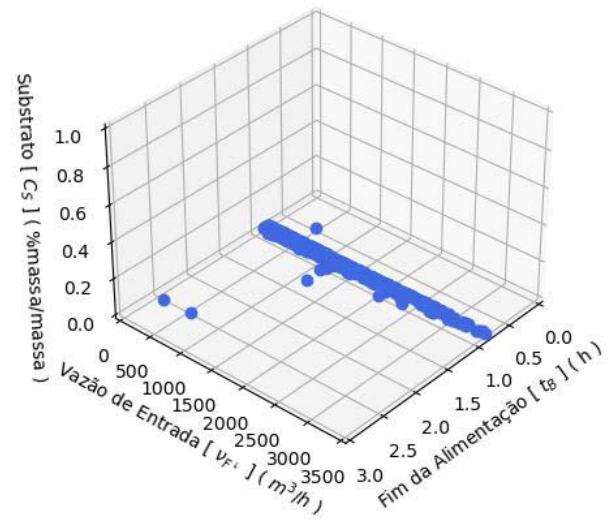
(d) $V. \text{ Entrada } (\nu_{F\downarrow}) \times F. \text{ Batelada } (t_D).$

Fonte: O autor, 2023.

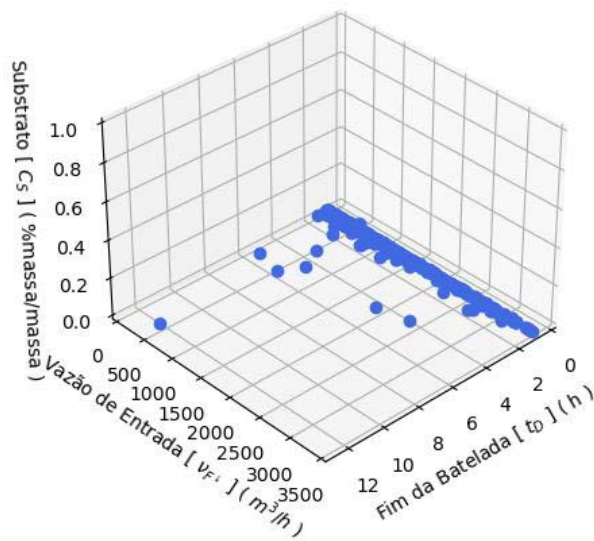
Gráfico 56 - Concentrações de açúcares correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo AGE-MOEA.



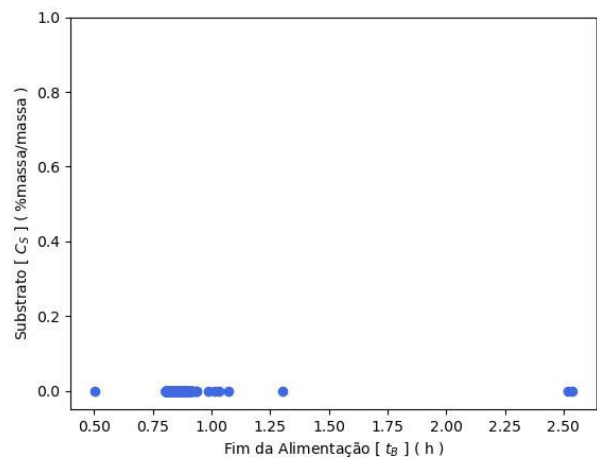
(a) Substrato (C_S) \times F. Alim. (t_B) \times F. Batelada (t_D).



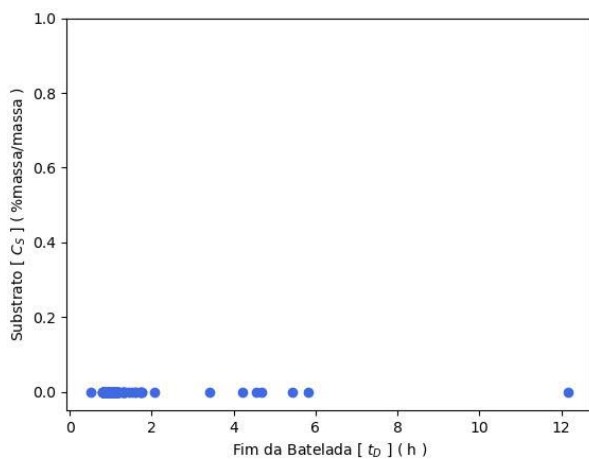
(b) Substrato (C_S) \times V. Entrada ($\nu_{F\downarrow}$) \times F. Alim. (t_B).



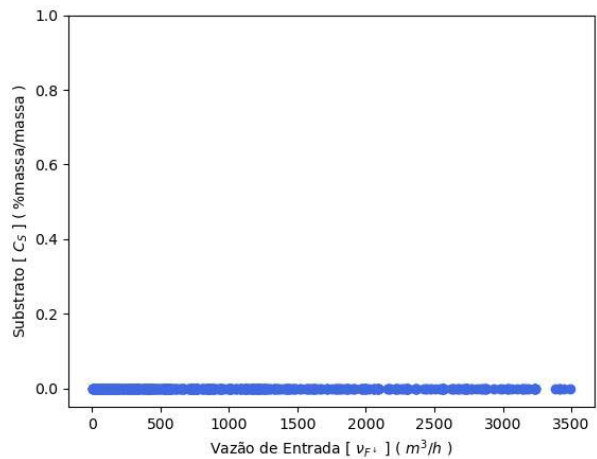
(c) Substrato (C_S) \times V. Entrada ($\nu_{F\downarrow}$) \times F. Batelada (t_D).



(d) Substrato (C_S) \times F. Alim. (t_B).

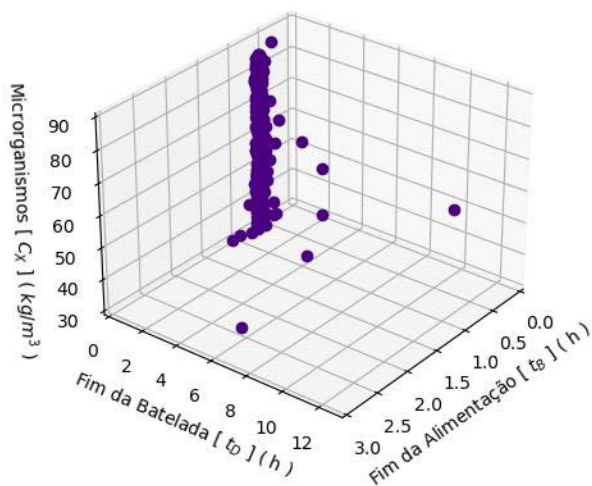


(e) Substrato (C_S) \times F. Batelada (t_D).

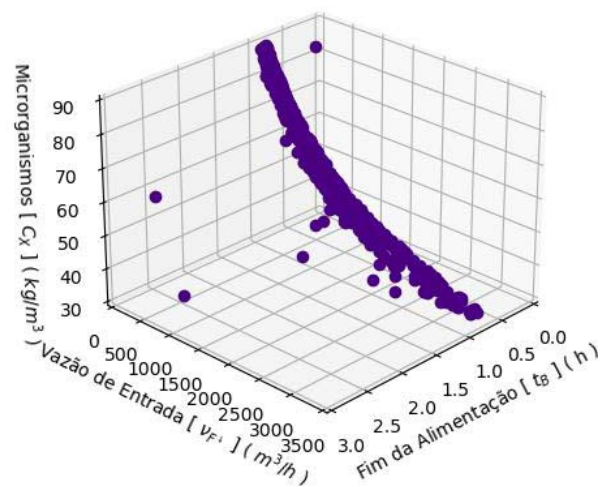


(f) Substrato (C_S) \times V. Entrada ($\nu_{F\downarrow}$).

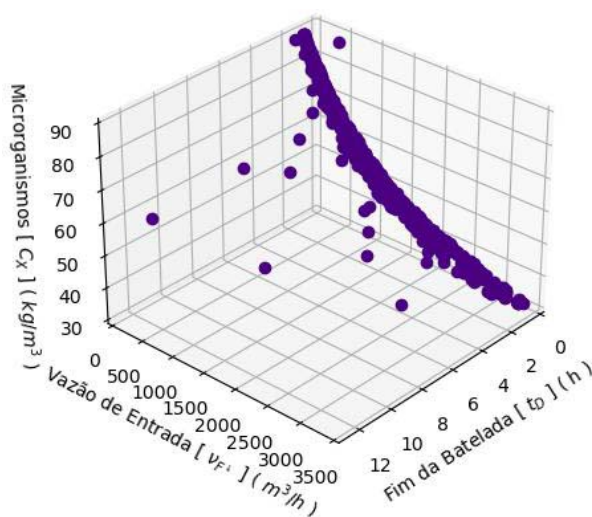
Gráfico 57 - Concentrações de levedura correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo AGE-MOEA.



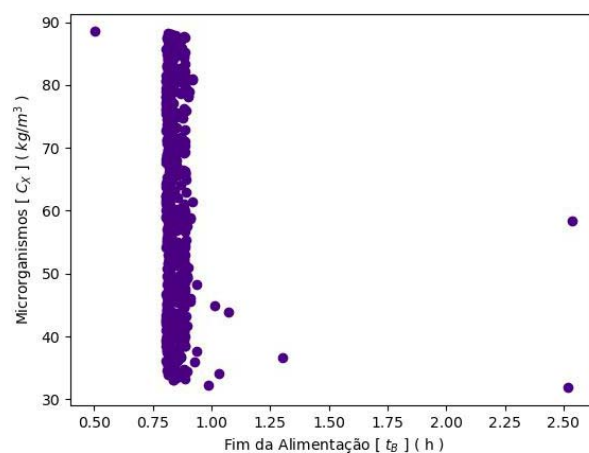
(a) Microorganismos(C_X) \times F. Alim.(t_B) \times F. Batelada(t_D).



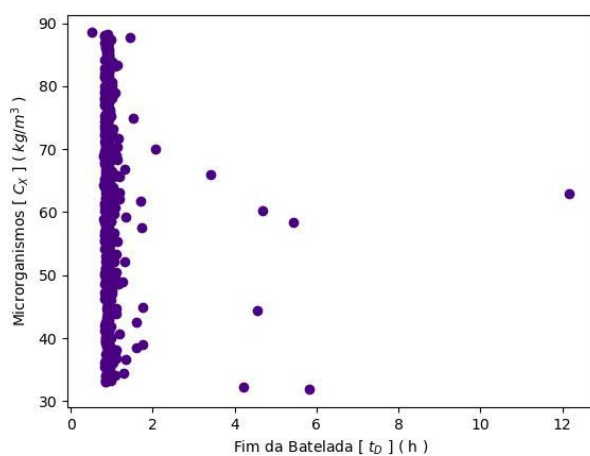
(b) Microorganismos(C_X) \times V. Entrada($\nu_{F\downarrow}$) \times F. Alim.(t_B).



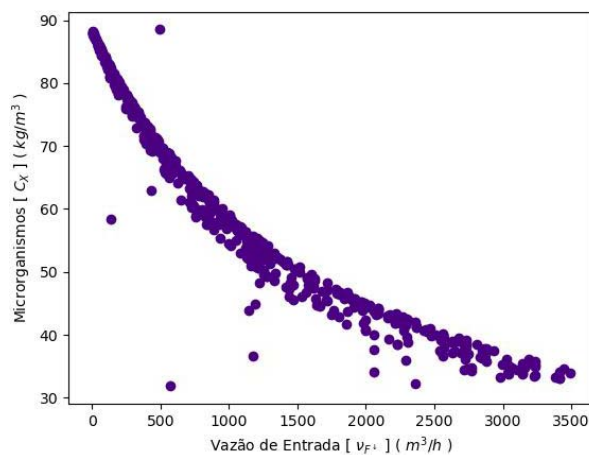
(c) Microorganismos (C_X) \times V. Entrada ($\nu_{F\downarrow}$) \times F. Batelada (t_D).



(d) Microorganismos (C_X) \times F. Alim. (t_B).

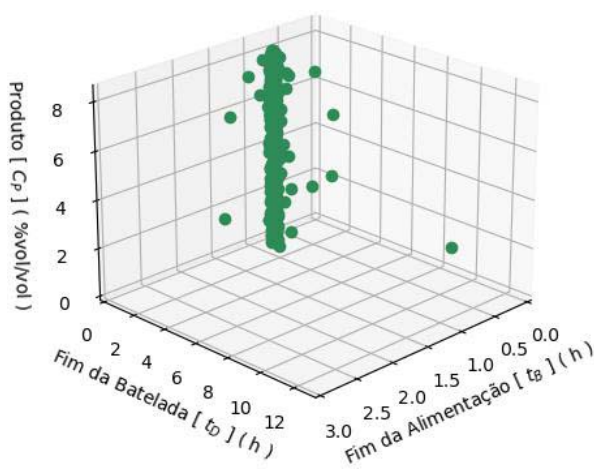


(e) Microorganismos (C_X) \times F. Batelada (t_D).

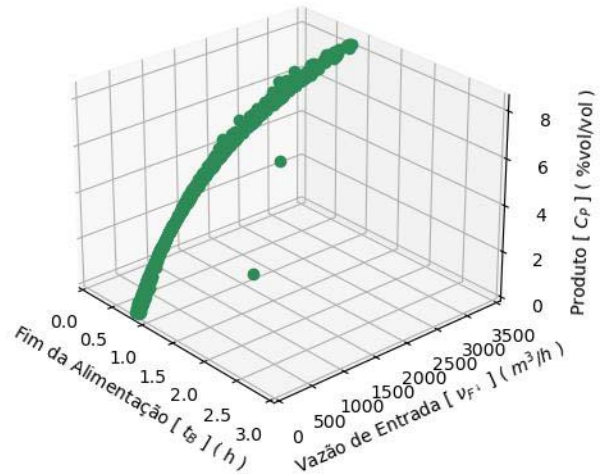


(f) Microorganismos (C_X) \times V. Entrada ($\nu_{F\downarrow}$).

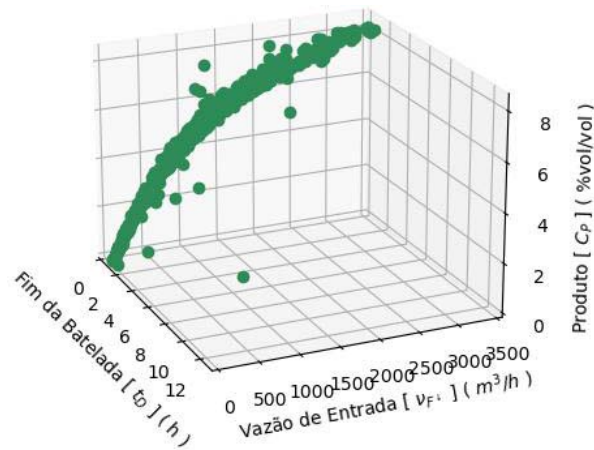
Gráfico 58 - Concentrações de etanol correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo AGE-MOEA.



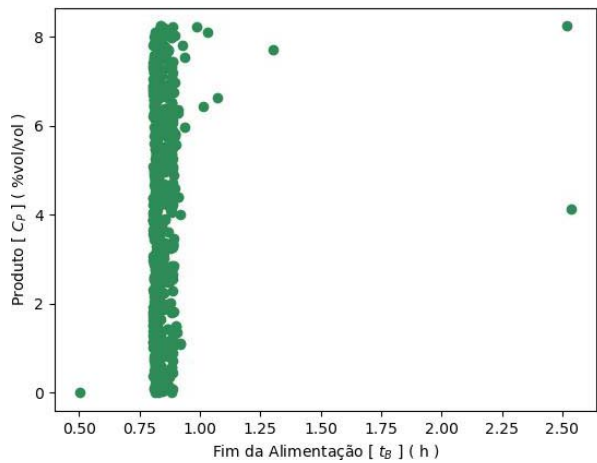
(a) Produto (C_P) \times F. Alim. (t_B) \times F. Batelada (t_D).



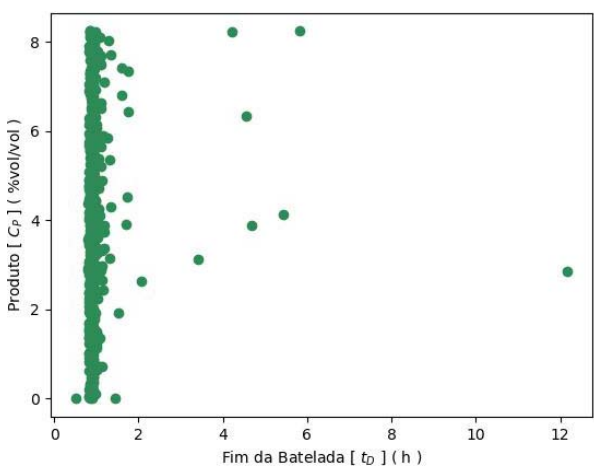
(b) Produto (C_P) \times V. Entrada ($\nu_{F\downarrow}$) \times F. Alim. (t_B).



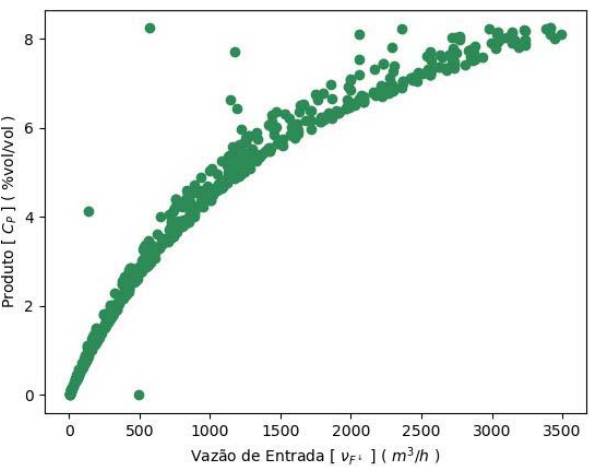
(c) Produto (C_P) \times V. Entrada ($\nu_{F\downarrow}$) \times F. Batelada (t_D).



(d) Produto (C_P) \times F. Alim. (t_B).



(e) Produto (C_P) \times F. Batelada (t_D).



(f) Produto (C_P) \times V. Entrada ($\nu_{F\downarrow}$).

Fonte: O autor, 2023.

8.4.3.3 Simulação com o C-TAEA

A simulação que buscou solucionar o Problema de Otimização B através do método C-TAEA demorou 10 horas e 25 minutos e 5 segundos para atingir as 50 gerações. O algoritmo convergiu na 31^a geração, encontrando 363 soluções não dominadas.

As soluções obtidas nesta simulação estão descritas, detalhadas e ilustradas nas subseções a seguir.

8.4.3.3.1 Fronteira de Pareto

O Gráfico 59 apresenta a fronteira de Pareto do problema de otimização em questão que foi estimada nesta simulação, acompanhada de suas respectivas projeções.

8.4.3.3.2 Conjunto Ótimo de Pareto

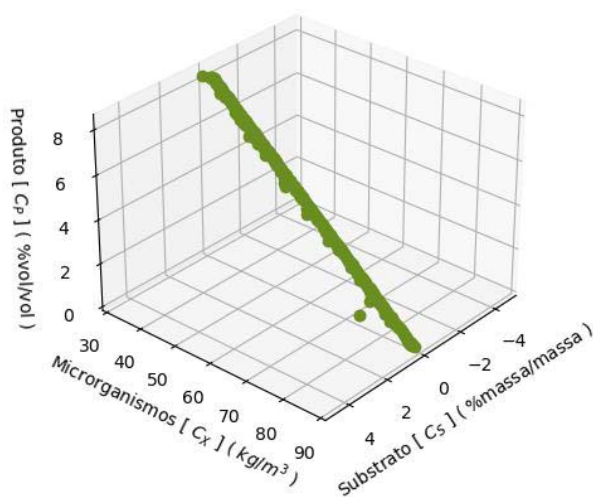
O Gráfico 60 ilustra os otimizadores de Pareto encontrados na presente simulação de otimização, nos quais consiste o conjunto ótimo de Pareto estimado.

8.4.3.3.3 Relação entre os Otimizadores de Pareto e os Objetivos

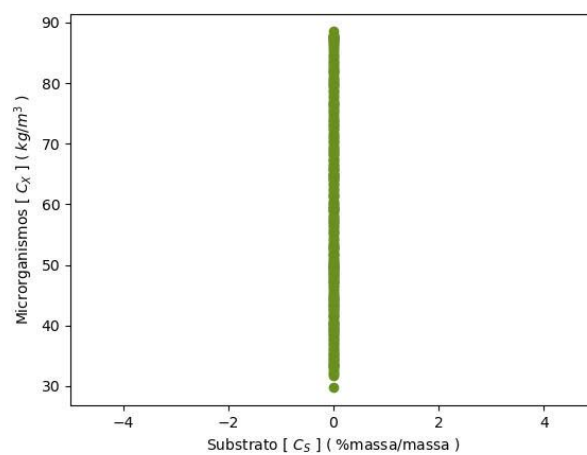
Nesta seção, serão apresentadas as correspondências dos pontos do conjunto ótimo de Pareto mostrado no Gráfico 60 com cada um dos objetivos do problema de otimização considerado.

O Gráfico 61 apresenta a concentração de substrato relacionada a cada um dos otimizadores do conjunto ótimo de Pareto, o Gráfico 62 enuncia a concentração de células de biomassa relacionada a cada um dos otimizadores de Pareto encontrados e o Gráfico 63 ilustra a concentração de produto relacionada a cada um dos elementos do conjunto ótimo.

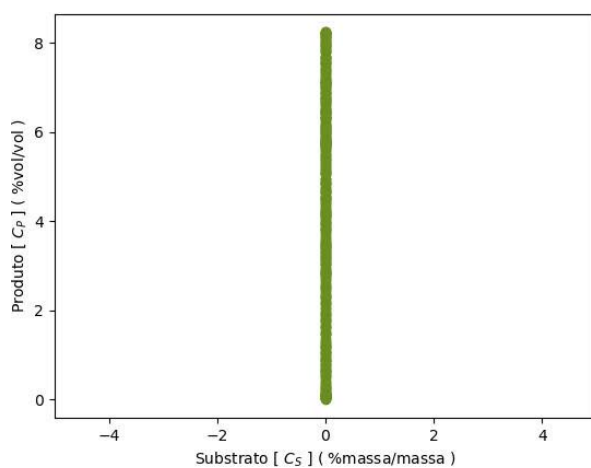
Gráfico 59 - Fronteira de Pareto do Problema B do Estudo de Caso 2 obtida pelo C-TAEA e suas projeções.



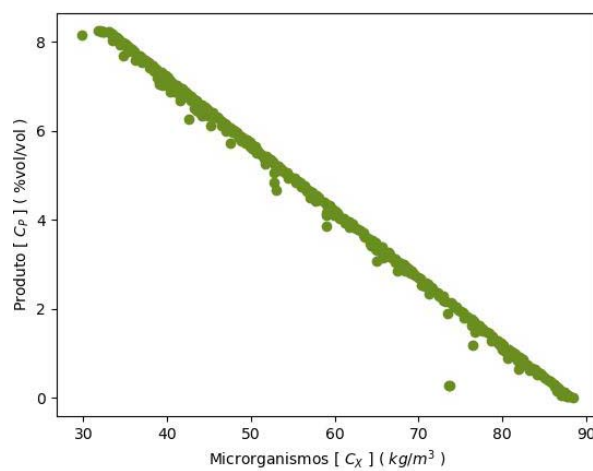
(a) Produto (C_P) \times Microorganismos (C_X) \times Substrato (C_S).



(b) Microorganismos (C_X) \times Substrato (C_S).



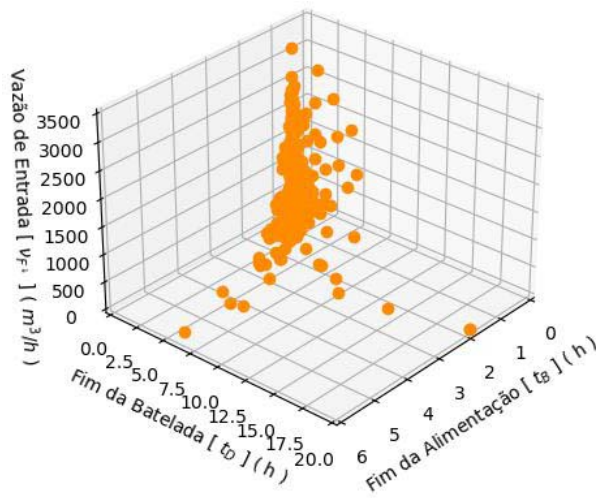
(c) Produto (C_P) \times Substrato (C_S).



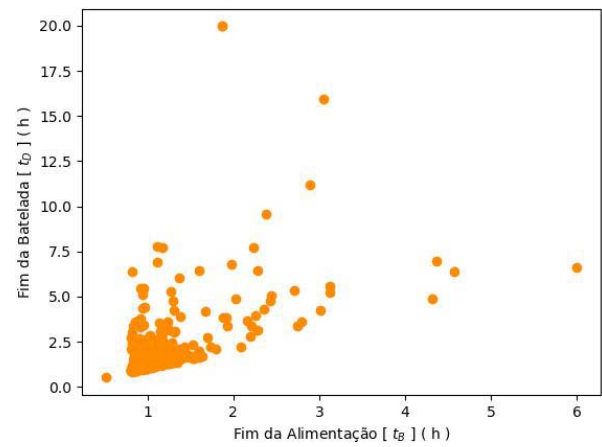
(d) Produto (C_P) \times Microorganismos (C_X).

Fonte: O autor, 2023.

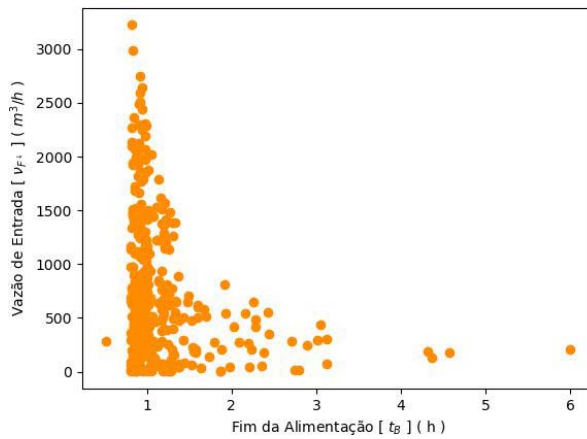
Gráfico 60 - Conjunto ótimo de Pareto do Problema B do Estudo de Caso 2 obtido pelo C-TAEA.



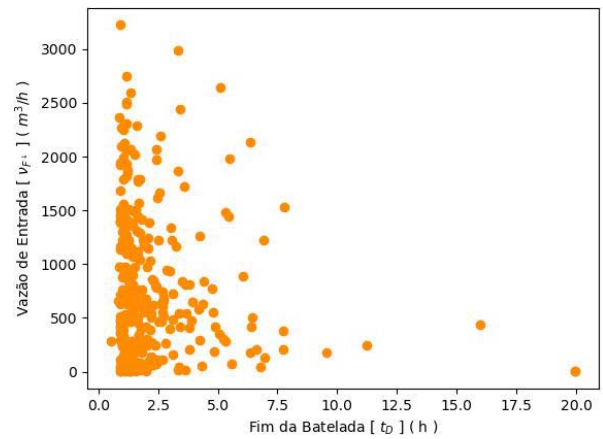
(a) V. Entrada ($\nu_{F\downarrow}$) \times F. Alim. (t_B) \times F. Batelada (t_D).



(b) F. Alim. (t_B) \times F. Batelada (t_D).



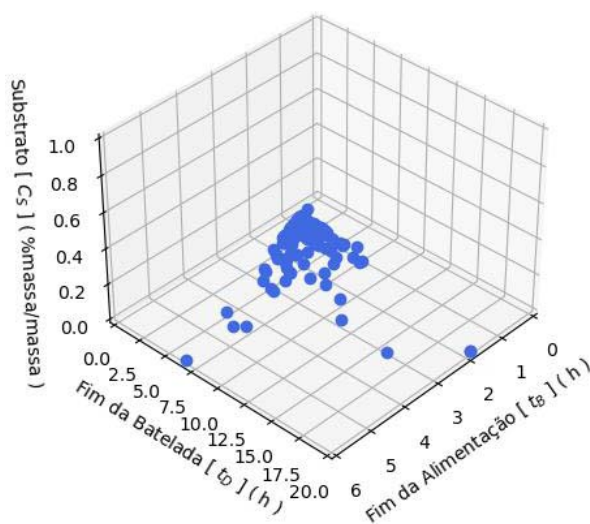
(c) V. Entrada ($\nu_{F\downarrow}$) \times F. Alim. (t_B).



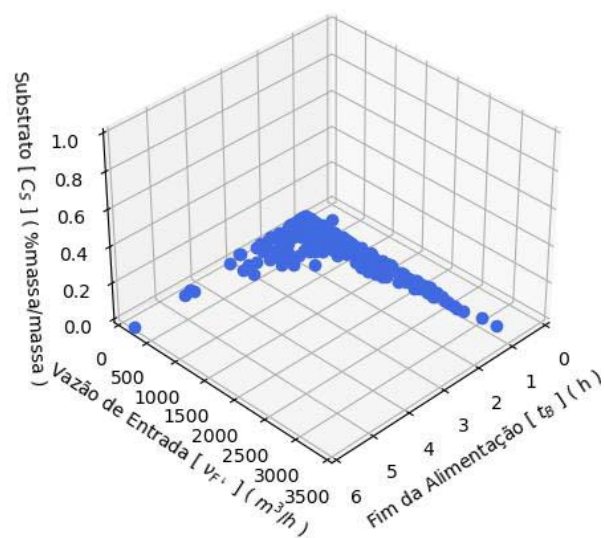
(d) V. Entrada ($\nu_{F\downarrow}$) \times F. Batelada (t_D).

Fonte: O autor, 2023.

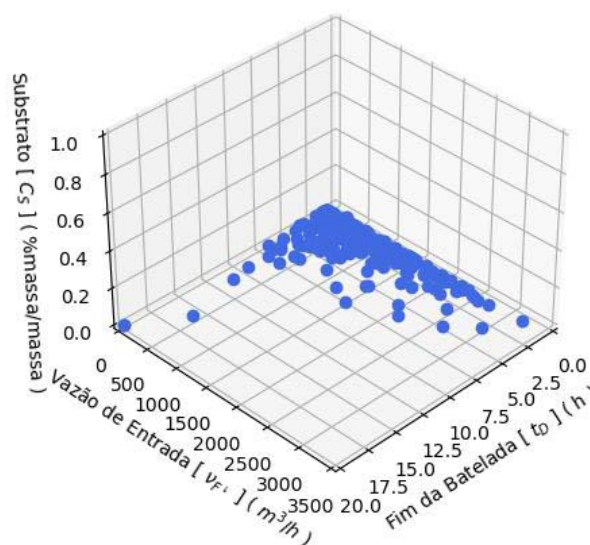
Gráfico 61 - Concentrações de açúcares correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo C-TAEA.



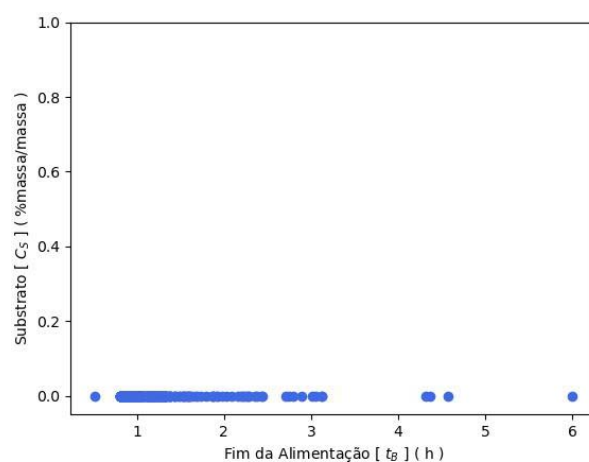
(a) Substrato (C_S) \times F. Alim. (t_B) \times F. Batelada (t_D).



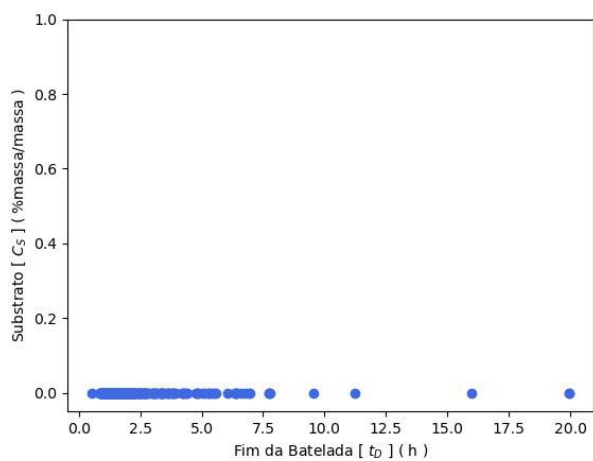
(b) Substrato (C_S) \times V. Entrada ($\nu_{F\downarrow}$) \times F. Alim. (t_B).



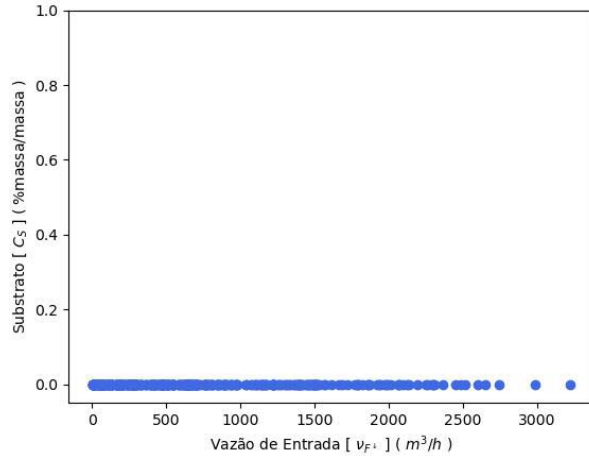
(c) Substrato (C_S) \times V. Entrada ($\nu_{F\downarrow}$) \times F. Batelada (t_D).



(d) Substrato (C_S) \times F. Alim. (t_B).

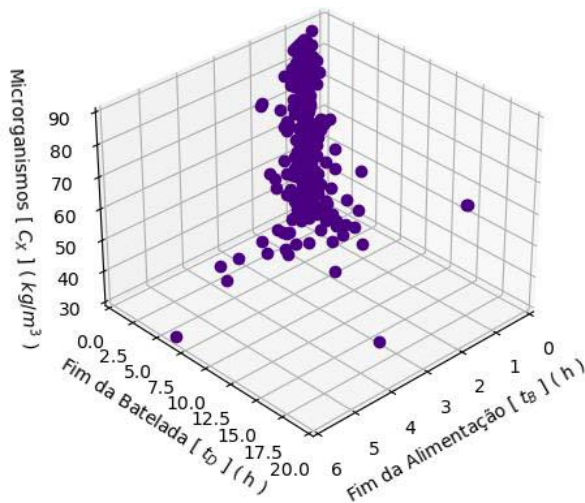


(e) Substrato (C_S) \times F. Batelada (t_D).

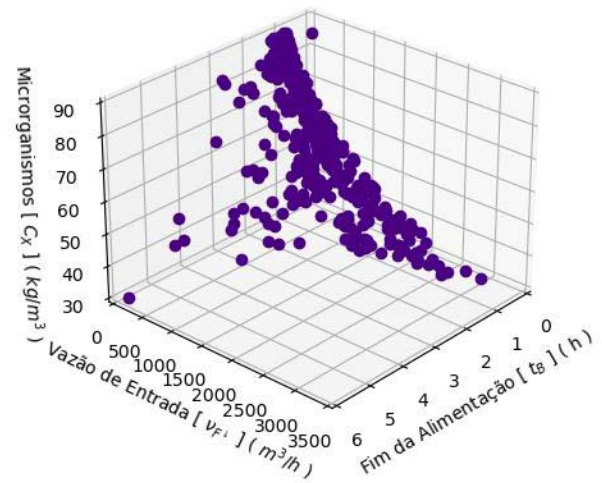


(f) Substrato (C_S) \times V. Entrada ($\nu_{F\downarrow}$).

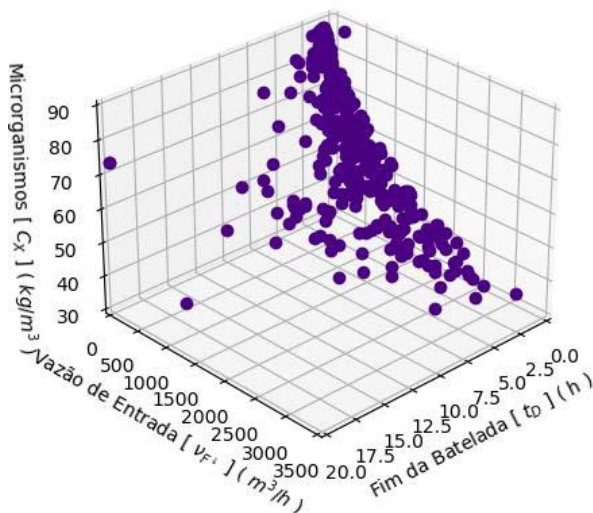
Gráfico 62 - Concentrações de levedura correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo C-TAEA.



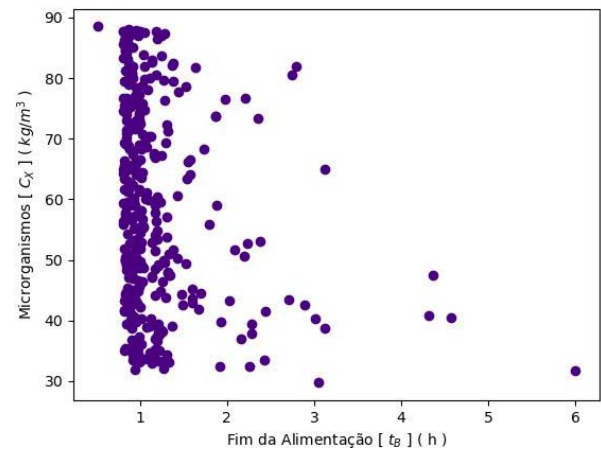
(a) $\text{Microorganismos}(C_X) \times \text{F. Alim.}(t_B) \times \text{F. Batelada}(t_D)$.



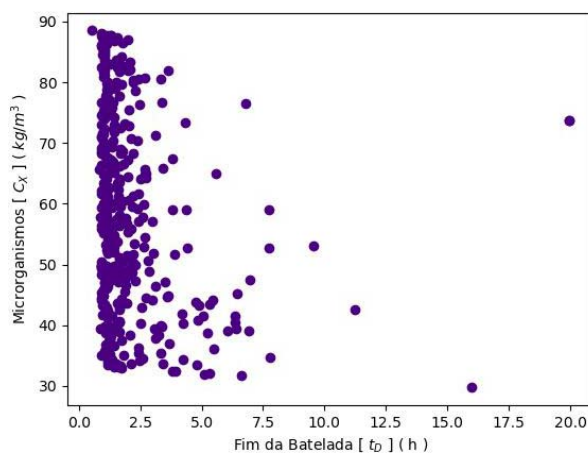
(b) $\text{Microorganismos}(C_X) \times \text{V. Entrada}(v_{F\downarrow}) \times \text{F. Alim.}(t_B)$.



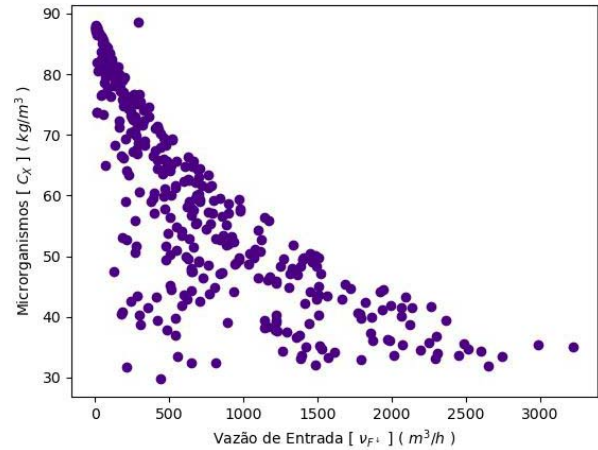
(c) $\text{Microorganismos}(C_X) \times \text{V. Entrada}(v_{F\downarrow}) \times \text{F. Batelada}(t_D)$.



(d) $\text{Microorganismos}(C_X) \times \text{F. Alim.}(t_B)$.

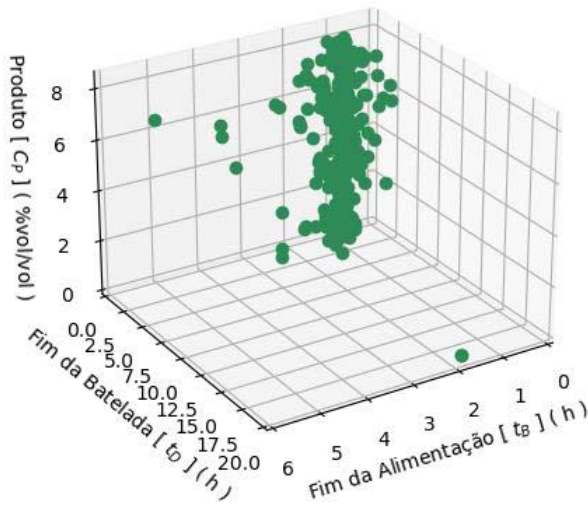


(e) $\text{Microorganismos}(C_X) \times \text{F. Batelada}(t_D)$.

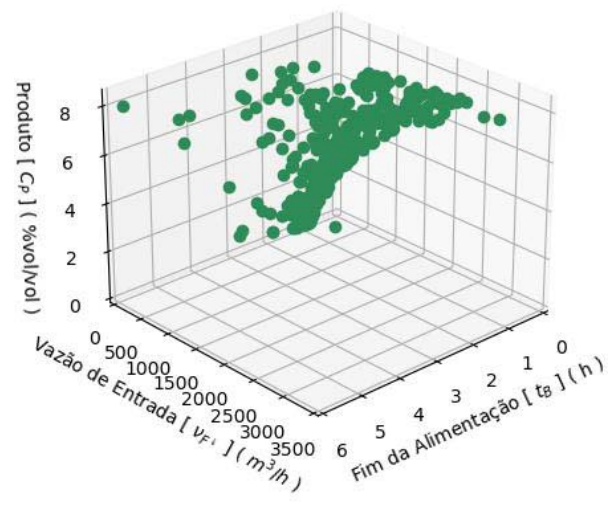


(f) $\text{Microorganismos}(C_X) \times \text{V. Entrada}(v_{F\downarrow})$.

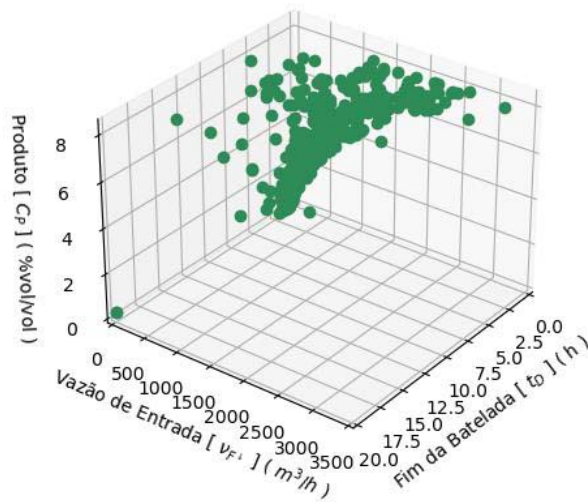
Gráfico 63 - Concentrações de etanol correspondentes aos otimizadores de Pareto do Problema B do Estudo de Caso 2 obtidos pelo C-TAEA.



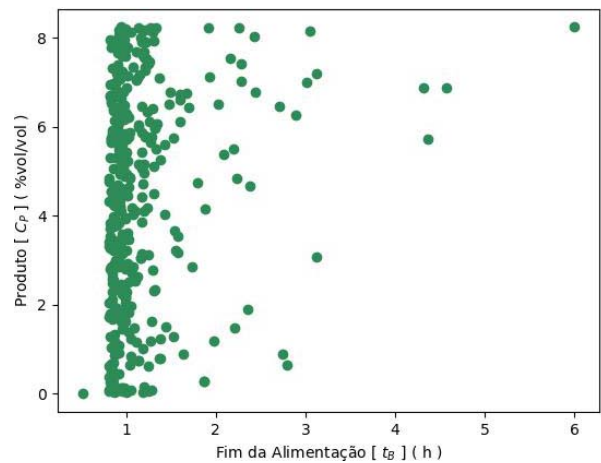
(a) Produto (C_P) \times F. Alim. (t_B) \times F. Batelada (t_D).



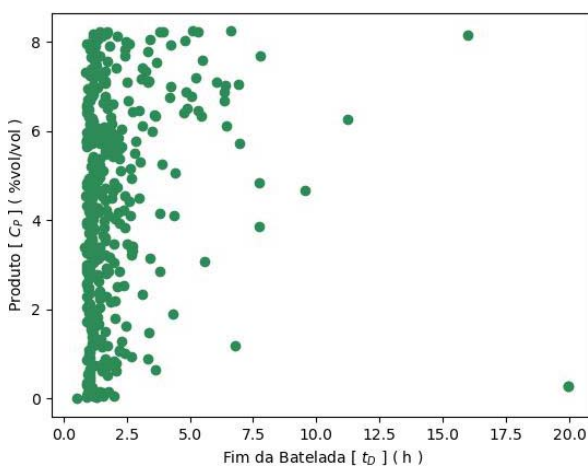
(b) Produto (C_P) \times V. Entrada ($\nu_{F\downarrow}$) \times F. Alim. (t_B).



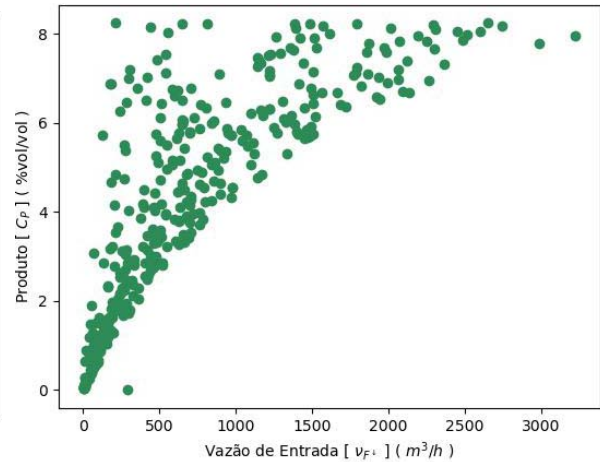
(c) Produto (C_P) \times V. Entrada ($\nu_{F\downarrow}$) \times F. Batelada (t_D).



(d) Produto (C_P) \times F. Alim. (t_B).



(e) Produto (C_P) \times F. Batelada (t_D).



(f) Produto (C_P) \times V. Entrada ($\nu_{F\downarrow}$).

8.4.3.4 Soluções Notáveis

Considerando os ótimos de Pareto obtidos em todas as simulações de otimização realizadas sobre este problema, foram elencadas as soluções que apresentam o melhor valor atingido para cada um dos objetivos, bem como, soluções que equilibram todos os objetivos.

A Tabela 14 expressa cada uma das soluções elencadas, bem como o método usado em sua obtenção.

Tabela 14 - Soluções Notáveis do Problema B do Estudo de Caso 2.

Solução	Variáveis			Objetivos		
	t_B	t_D	$\nu_{F\downarrow}$	$C_S(t_H)$	$C_X(t_H)$	$C_P(t_H)$
AGE-MOEA						
E2P2M1S1	0,50116	0,50125	491,66247	0,00000	88,51741	0,00000
E2P2M1S2	0,83847	0,85309	3413,81714	0,00000	33,15160	8,23981
E2P2M1S3	2,53868	5,43871	137,76785	0,00000	58,44067	4,12138
C-TAEA						
E2P2M2S1	0,50894	0,51154	287,74871	0,00000	88,50984	0,00000
E2P2M2S2	0,93506	5,12579	2647,27755	0,00000	31,99123	8,24062
E2P2M2S2	0,94312	4,38025	630,35777	0,00000	59,03835	4,10643
NSGA-II						
E2P2M3S1	0,50196	0,50299	13,18429	0,00000	88,51612	0,00000
E2P2M3S2	0,90742	0,91676	2823,26096	0,00000	33,13018	8,24038
E2P2M3S3	0,80873	0,85785	906,31473	0,00000	60,75808	4,11099

Fonte: O autor, 2023.

8.5 Análise e Discussão do Estudo de Caso 2

Nesta seção, discorreremos sobre todos os dados e resultados obtidos através das simulações que foram apresentadas nas seções anteriores, realizando análises destas informações e discutindo pontos relevantes que sejam inerentes a elas.

Importante frisar que durante as análises foram elencados alguns cenários que foram discutidos. No entanto, isso não significa que estes são os melhores cenários para serem empregados na prática, uma vez que essa é uma decisão de negócio que, na maioria das vezes, estará vinculada a fatores mercadológicos, tais como, valores de insumos e valor de produto, neste caso do etanol, praticados pelo mercado. Esses fatores, apresentam volatilidade, mudando ao longo do tempo.

Assim, uma solução boa atualmente pode ser ruim no futuro. Por esta razão, somente o tomador de decisão, que estará a par de todos os fatores envolvidos, poderá

realizar a melhor escolha para o cenário de mercado contemporâneo a ele.

8.5.1 Problema de Otimização A

Na presente subseção, estão relacionadas e descritas as análises e discussões feitas em relação às informações e resultados obtidos a partir das simulações realizadas sobre o Problema de Otimização A.

8.5.1.1 Considerações sobre os Métodos de Otimização

Sem o intuito de comparar os métodos de otimização empregados nas simulações realizadas, verifica-se que, com relação ao tempo de convergência dos métodos de otimização nas referidas simulações, verifica-se que o AGE-MOEA e o NSGA-II foram os métodos que levaram menos iterações para convergir, enquanto o C-TAEA foi o método que mais demorou a convergir, apresentando o maior número de iterações para isto.

No que se refere ao tempo de execução levado pelos métodos de otimização para completar as 50 gerações, o C-TAEA foi o método que apresentou a menor duração de execução, o AGE-MOEA teve o segundo menor tempo de execução e o NSGA-II foi o método que mais tempo levou para finalizar sua execução.

No que diz respeito à quantidade de soluções encontradas, tanto o AGE-MOEA quanto o NSGA-II encontraram 600 soluções não dominadas, enquanto o C-TAEA encontrou 298 soluções não dominadas.

Todos os métodos de otimização empregados convergiram para resultados muito próximos uns dos outros, não existindo assim razão para analisar e discutir separadamente os resultados de cada um, uma vez que todos corroboraram entre si.

8.5.1.2 Considerações sobre os Resultados

Considerando os resultados das simulações de otimização realizadas sobre o Problema A deste estudo de caso, verifica-se que o instante de término da alimentação do reator possui maior influência em todos os objetivos, quando comparado ao tempo de batelada.

Essa maior influência se deve provavelmente ao fato do tempo de alimentação regular a quantidade de suco no reator, que por sua vez influi diretamente na concentração de açúcares e na diluição da biomassa no reator.

Todos os otimizadores de Pareto obtidos zeram a quantidade de substrato, o que

significa que houve máxima conversão de açúcares em todas as soluções. Esse fato pode provocar estranheza em um primeiro olhar, pois pode-se pensar que em durações de tempo muito reduzidas, a levedura não teria tempo suficiente para fermentar os açúcares.

Entretanto, provavelmente, isto se deve à quantidade de açúcares ser diretamente proporcional ao tempo de alimentação, de forma que em durações menores da fase de alimentação, menos suco é inserido e conseqüentemente, menos açúcares, sendo o contrário também verdadeiro. Deste modo, a levedura possui tempo suficiente para converter completamente os açúcares em todos os cenários.

A concentração de leveduras ao final da batelada varia fortemente em relação ao tempo de fermentação, de forma que, quanto maior for este tempo, menor a quantidade de células de biomassa vivas no vinho de levedurado, ao final do processo.

Nota ao Leitor 24 *Nesta seção, entende-se por tempo de fermentação ou tempo fermentativo, a soma dos tempos de alimentação e de batelada.*

O inverso acontece com a concentração de etanol, ao final da batelada, pois quanto maior for o tempo de fermentação, maior é a quantidade de álcool presente no vinho de levedurado, ao término do processo.

Ao observar a relação entre os otimizadores de Pareto encontrados e a concentração de biomassa, contrastando-a com a relação entre a concentração de etanol e os pontos do conjunto ótimo de Pareto, constata-se que as concentrações de biomassa e de álcool têm comportamentos espelhados.

Logo, percebe-se que existe um conflito direto entre a concentração de levedura e a concentração de etanol. Este conflito é claramente perceptível quando se observa fronteira de Pareto do problema, onde verifica-se que conforme a quantidade de etanol aumenta, a concentração de levedura diminui e vice-versa.

Essa correlação entre as concentrações de etanol e de levedura se deve principalmente à toxicidade do etanol para as células de biomassa. Os microrganismos convertem os açúcares presentes no mosto em álcool ao longo do tempo. Entretanto, conforme a quantidade de etanol presente no vinho vai aumentando, em função da ação fermentativa das leveduras, estas vão morrendo devido a ação antisséptica do etanol.

Dessa forma, quanto maior a duração da fase alimentação, mais tempo e açúcares estão disponíveis para as leveduras gerarem etanol, que por sua vez, as mata, conforme sua concentração aumenta. A redução da concentração de leveduras na fase de alimentação também se deve a diluição desta em maiores quantidades de suco no reator, conforme o tempo de alimentação aumenta.

Com relação à duração da fase de batelada, quanto maior ela for, mais tempo os microrganismos terão para converter os açúcares presentes no mosto em álcool, mas estes também terão um maior tempo de exposição ao etanol, que estará em concentração crescente, provocando a morte gradual destes.

8.5.1.3 Análise de Cenários

Levando em conta as considerações apresentadas e os resultados obtidos, podemos destacar alguns cenários para serem analisados e discutidos.

Nota ao Leitor 25 *Nas análises de cenários do presente problema, entende-se por solução ou otimizador original, a solução cujos os valores das variáveis são aqueles empregados pelo processo não otimizado cuja simulação foi realizada na Seção 8.4.1.*

8.5.1.3.1 Cenário 1

O primeiro cenário elencado para este problema considerou o otimizador E2P1M1S2, descrito na Tabela 13, que apresenta a maior geração de etanol dentre as demais soluções obtidas.

A Tabela 15 apresenta os valores das variáveis e dos componentes ao fim do processo fermentativo, tanto para a solução original quanto para a solução considerada neste cenário.

Tabela 15 - Valores das variáveis e objetivos referentes à solução original e a considerada no Cenário 1 levantado para o Problema A do Estudo de Caso 2.

Solução	Variáveis		Objetivos		
	t_B	t_D	$C_S(t_H)$	$C_X(t_H)$	$C_P(t_H)$
Original	5,50000	8,50000	0,00000	31,20690	8,23728
E2P1M1S2	5,50276	6,76941	0,00000	31,64833	8,23885

Fonte: O autor, 2023.

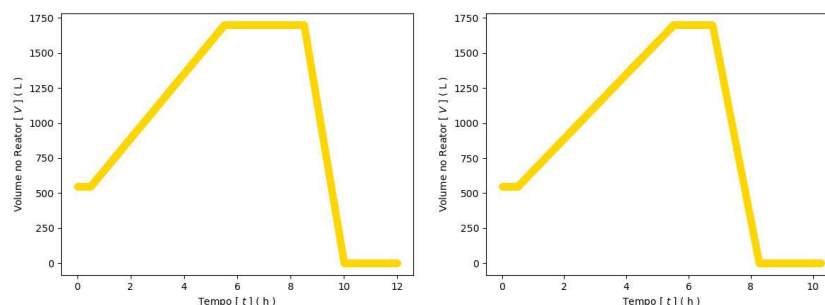
O Gráfico 64 apresenta os comportamentos das concentrações de cada um dos componentes ao longo do tempo no reator de fermentação alcoólica para solução original e para a solução considerada neste cenário.

Ao contrastar a solução original com a considerada neste cenário, constatou-se uma preservação de levedura, na solução considerada, de 0,44143 kg/m³ a mais que na solução original empregada no processo.

Com relação à produção etanólica, a solução considerada neste cenário apresenta um ganho de 0,00157 %vol/vol em relação a solução original, valor que se traduz em 12,39201 g/m³ a mais de álcool.

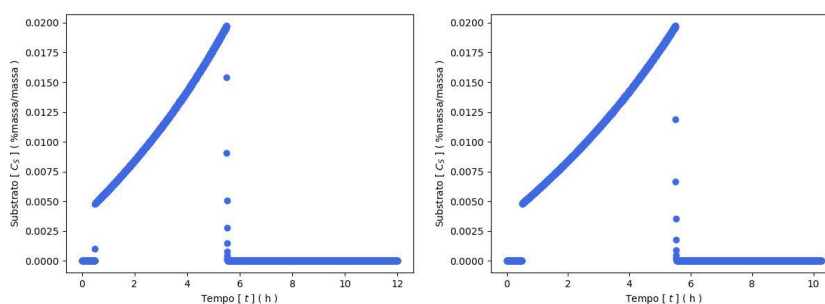
Como a solução considerada apresenta o reator completamente ocupado, tal como a solução original, pode-se expressar as comparações realizadas entre as soluções em valores absolutos.

Gráfico 64 - Concentrações ao longo do tempo na solução original e na solução considerada no Cenário 1 levantado para o Problema A do Estudo de Caso 2.



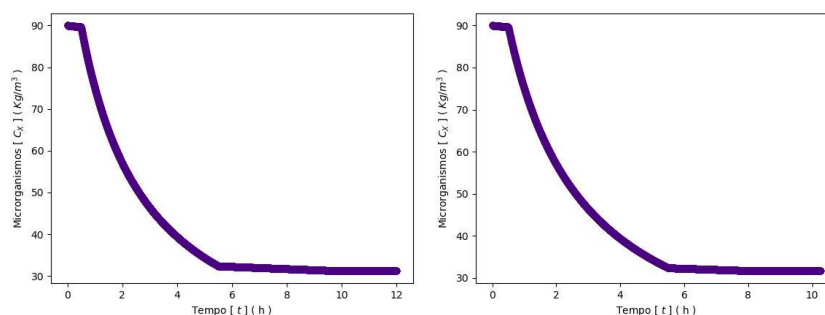
(a) Volume no reator na solução original.

(b) Volume no reator na solução considerada.



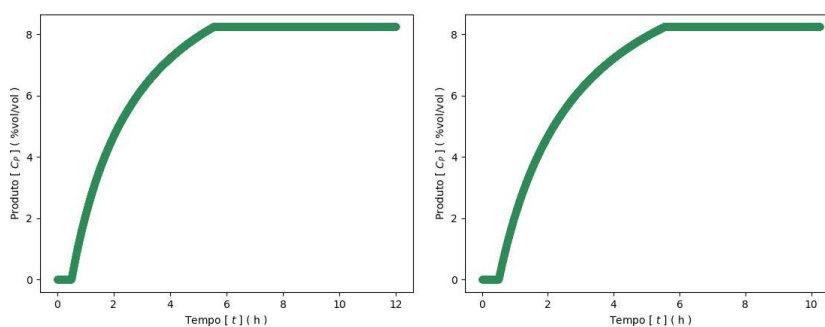
(c) Concentração de substrato na solução original.

(d) Concentração de substrato na solução considerada.



(e) Conc. de microrganismos na solução original.

(f) Conc. de microrganismos na solução considerada.



(g) Concentração de produto na solução original.

(h) Concentração de produto na solução considerada.

Dessa forma, neste cenário, existe um ganho de 21,05402 kg na produção de etanol e de 749,98957 kg na preservação de levedura.

Outro ganho percebido na solução considerada neste cenário com relação à solução original é o de tempo, uma vez que o processo na solução considerada leva 1,73059 h a menos que na solução original.

O ganho na preservação de levedura se deve provavelmente ao menor tempo de exposição ao etanol na solução considerada, uma vez que esta apresenta um tempo menor de fermentação quando comparado ao da solução original.

O tomador de decisão pode avaliar se em escala, os ganhos obtidos neste cenário, justificam a implantação da solução considerada, avaliando se esta trará um ganho relevante para o processo.

8.5.1.3.2 Cenário 2

O primeiro cenário elencado para este problema considerou o otimizador E2P1M1S3, descrito na Tabela 13, que apresenta um equilíbrio entre a geração de etanol e a preservação de levedura.

A Tabela 16 apresenta os valores das variáveis e dos componentes ao fim do processo fermentativo, tanto para a solução original quanto para a solução considerada neste cenário.

Tabela 16 - Valores das variáveis e objetivos referentes à solução original e a considerada no Cenário 2 levantado para o Problema A do Estudo de Caso 2.

Solução	Variáveis		Objetivos		
	t_B	t_D	$C_S(t_H)$	$C_X(t_H)$	$C_P(t_H)$
Original	5,50000	8,50000	0,00000	31,20690	8,23728
E2P1M1S3	1,72389	1,88808	0,00000	60,14030	4,12747

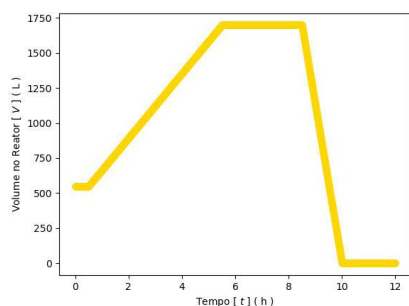
Fonte: O autor, 2023.

O Gráfico 65 apresenta os comportamentos das concentrações de cada um dos componentes ao longo do tempo no reator de fermentação alcoólica para solução original e para a solução considerada neste cenário.

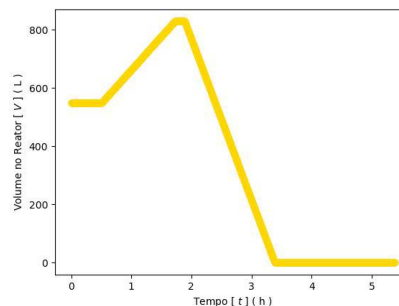
Ao contrastar a solução original com a considerada neste cenário, contata-se uma preservação de levedura, na solução considerada, de 28,9334 kg/m³ a mais que na solução original empregada no processo.

Com relação à produção etanólica, a solução considerada neste cenário apresenta uma perda de 4,10981 %vol/vol em relação a solução original, valor que se traduz em

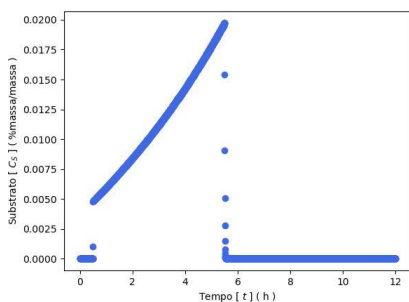
Gráfico 65 - Concentrações ao longo do tempo na solução original e na solução considerada no Cenário 2 levantado para o Problema A do Estudo de Caso 2.



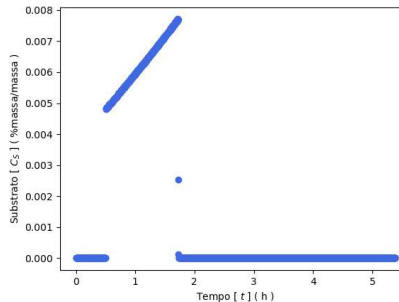
(a) Volume no reator na solução original.



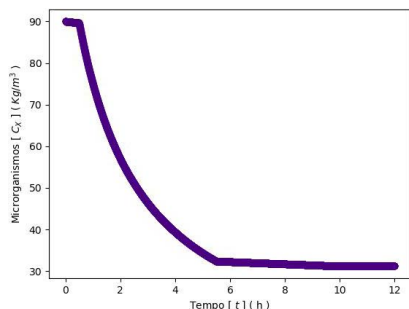
(b) Volume no reator na solução considerada.



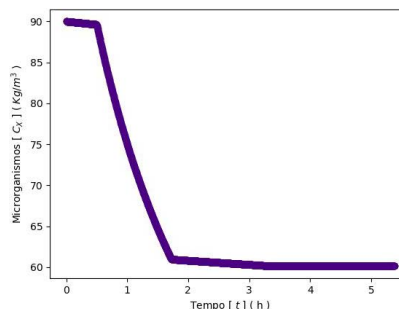
(c) Concentração de substrato na solução original.



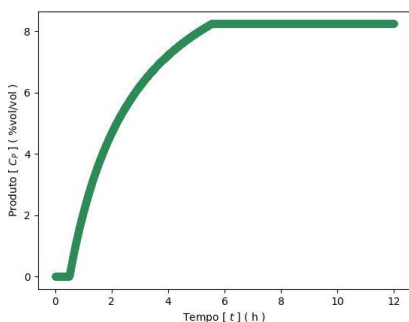
(d) Concentração de substrato na solução considerada.



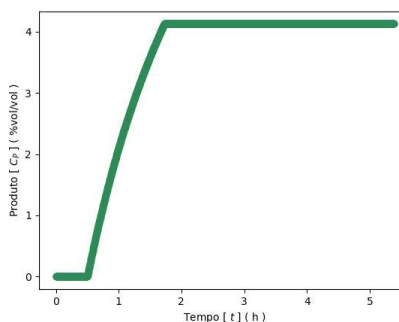
(e) Conc. de microrganismos na solução original.



(f) Conc. de microrganismos na solução considerada.



(g) Concentração de produto na solução original.



(h) Concentração de produto na solução considerada.

32,43873 kg/m³ a menos de álcool.

A solução considerada neste cenário apresenta um processo que leva 6,61192 h a menos que o processo dado pela solução original.

O ganho na preservação de levedura se deve ao baixo tempo de exposição ao etanol na solução considerada, uma vez que esta apresenta um tempo consideravelmente menor de fermentação quando comparado ao da solução original, além de uma menor diluição, uma vez que a quantidade de suco atingida no reator na solução considerada é menor que na original.

No entanto, a solução considerada no presente cenário leva a uma perda considerável na geração de etanol, fazendo-a cair pela metade.

Como a solução considerada apresenta uma quantidade menor de suco no reator que a solução original, não cabe, neste cenário, uma comparação de valores absolutos.

8.5.2 Problema de Otimização B

Na presente subseção, estão relacionadas e descritas as análises e discussões feitas em relação às informações e resultados obtidos a partir das simulações realizadas sobre o Problema de Otimização B.

8.5.2.1 Considerações sobre os Métodos de Otimização

Sem a pretensão de comparar os métodos de otimização usados nas simulações, cabe mencionar que, no que tange ao tempo de convergência, constata-se que o AGE-MOEA foi o método que convergiu mais rapidamente, apresentando o menor número de iterações para convergir, seguido pelo NSGA-II, que teve o segundo menor número de iterações, e por final, o C-TAEA, que apresentou o maior número de iterações para convergir.

Com relação ao tempo de execução que os métodos de otimização levaram para chegar à 50^a geração, o C-TAEA foi o método que teve o menor tempo total de execução, seguido pelo NSGA-II, que apresentou o segundo menor tempo e finalmente o AGE-MOEA, que levou o maior tempo para concluir sua execução.

No tocante ao número de soluções encontradas, tanto o NSGA-II quanto o AGE-MOEA acharam 600 soluções não dominadas, enquanto o C-TAEA encontrou apenas 363 soluções não dominadas.

Assim como no problema anterior, todos os métodos de otimização empregados convergiram para resultados muito próximos uns dos outros. Desta forma, não existe

motivos que justifiquem analisar e discutir separadamente os resultados obtidos por cada um, uma vez que todos corroboraram entre si.

8.5.2.2 Considerações sobre os Resultados

Levando em conta os resultados obtidos nas simulações promovidas sobre o Problema B do presente estudo de caso, percebe-se que a vazão de entrada é a variável que exerce maior influência sobre os objetivos do problema de otimização.

Essa configuração de influência se deve ao controle direto que a vazão de entrada possui sobre a concentração de açúcares e diluição da biomassa no reator, embora o tempo de alimentação também regule esse fatores, pois é ele quem determina quanto tempo a vazão de entrada no reator permanece ativa.

O tempo de batelada, neste caso, apresenta maior relevância em algumas soluções, uma vez que agora, a injeção de açúcares no reator pode não mais ser proporcional ao tempo de alimentação, levando as leveduras a necessitarem de um tempo maior para convertê-los em etanol.

Assim como no problema anterior, aqui também existe conversão máxima de açúcares para todos os otimizadores de Pareto obtidos. Esse resultado se configura neste problema, pois para vazões mais altas, existe um menor tempo de alimentação e vice versa a fim de respeitar a capacidade do reator. Dessa forma, existe tempo hábil para que os açúcares sejam convertidos, não existindo soluções ótimas que apresentam açúcares remanescentes no reator.

Do mesmo modo que no problema anterior, a correlação conflitosa entre a concentração de células de biomassa e a concentração de álcool está presente também neste problema, gerando as mesmas implicações e impactos.

A presença da vazão de entrada como variável de decisão altera consideravelmente o comportamento das concentrações de etanol e de levedura, quando contrastado com o do problema anterior, em que consideramos apenas o tempo de alimentação e o tempo de batelada como variáveis de otimização.

A maior parte das soluções ótimas apresenta tempos de fermentação mais baixos, que variam fortemente de acordo com a vazão de entrada de suco no reator de fermentação em batelada alimentada.

Neste contexto de menores tempos de fermentação, para menores vazões de entrada, as soluções apresentaram maior preservação de células de biomassa vivas e em contraparte, menor geração de álcool.

Neste mesmo contexto, para maiores vazões de entrada, a produção de etanol é maior e por outro lado existe uma maior taxa de morte de células de levedura no vinho dentro do reator de fermentação.

Algumas soluções apresentam maiores tempos de fermentação. Dentre elas, os otimizadores com maiores tempos de batelada ou maiores vazões apresentam menores tempo de alimentação e as soluções que apresentam maiores tempos de alimentação e de batelada possuem baixas vazões.

8.5.2.3 Análise de Cenários

Considerando as ponderações apresentadas e os resultados obtidos, podemos destacar alguns cenários para serem analisados e discutidos.

Nota ao Leitor 26 *Assim como no problema anterior, nas análises de cenários do presente problema, entende-se por solução ou otimizador original, a solução cujos os valores das variáveis são aqueles empregados pelo processo não otimizado cuja simulação foi realizada na Seção 8.4.1.*

8.5.2.3.1 Cenário 1

O primeiro cenário elencado para este problema considerou o otimizador E2P2M3S2, descrito na Tabela 14, que apresenta a maior geração de etanol dentre os demais otimizadores obtidos.

A Tabela 17 apresenta os valores das variáveis e dos componentes ao fim do processo fermentativo, tanto para a solução original quanto para a solução considerada neste cenário.

Tabela 17 - Valores das variáveis e objetivos referentes à solução original e a considerada no Cenário 1 levantado para o Problema B do Estudo de Caso 2.

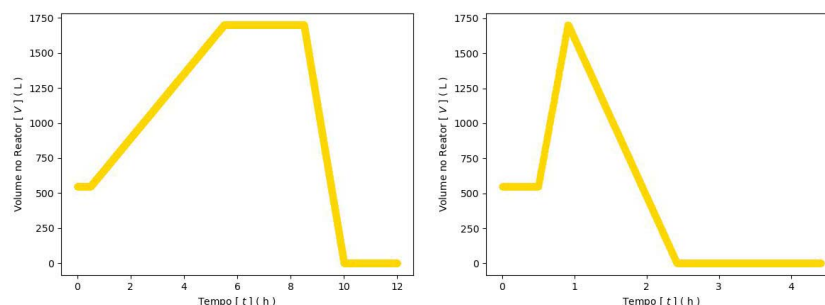
Solução	Variáveis			Objetivos		
	t_B	t_D	$\nu_{F\downarrow}$	$C_S(t_H)$	$C_X(t_H)$	$C_P(t_H)$
Original	5,50000	8,50000	230,00000	0,00000	31,20690	8,23728
E2P2M3S2	0,90742	0,91676	2823,26096	0,00000	33,13018	8,24038

Fonte: O autor, 2023.

O Gráfico 66 apresenta os comportamentos das concentrações de cada um dos componentes ao longo do tempo no reator de fermentação alcoólica para solução original e para a solução considerada neste cenário.

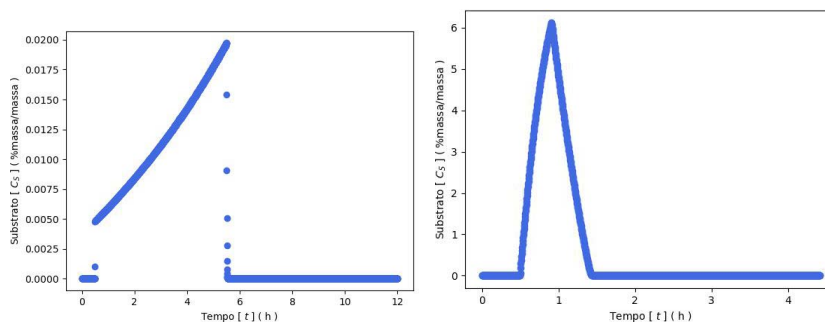
Ao contrastar a solução original com a considerada neste cenário, contata-se uma

Gráfico 66 - Concentrações ao longo do tempo na solução original e na solução considerada no Cenário 1 levantado para o Problema A do Estudo de Caso 2.



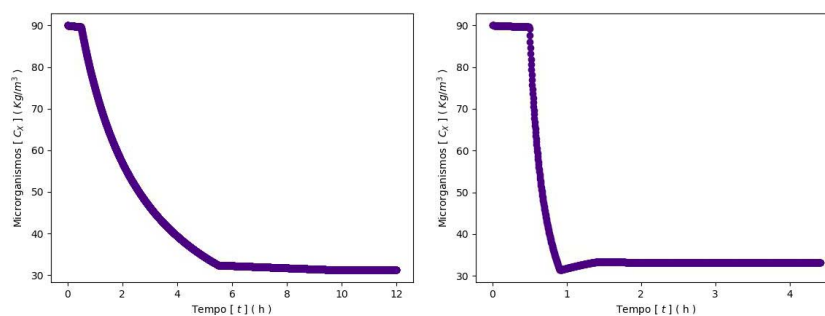
(a) Volume no reator na solução original.

(b) Volume no reator na solução considerada.



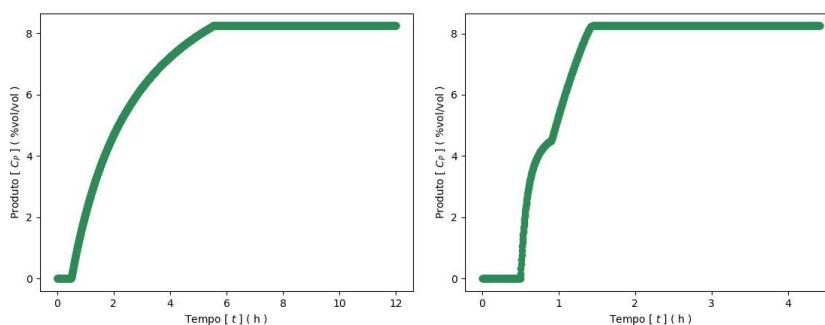
(c) Concentração de substrato na solução original.

(d) Concentração de substrato na solução considerada.



(e) Conc. de microrganismos na solução original.

(f) Conc. de microrganismos na solução considerada.



(g) Concentração de produto na solução original.

(h) Concentração de produto na solução considerada.

preservação de levedura, na solução considerada, de 1,92328 kg/m³ a mais que na solução original empregada no processo.

Com relação à produção etanólica, a solução considerada neste cenário apresenta um ganho de 0,00310 %vol/vol em relação a solução original, valor que se traduz em 24,4683 g/m³ a mais de álcool.

Como a solução considerada apresenta o reator completamente ocupado, tal como a solução original, pode-se expressar as comparações realizadas entre as soluções em valores absolutos.

Dessa forma, neste cenário, existe um ganho de 40,86777 kg na produção de etanol e de 3.267,65 kg na preservação de levedura.

Também existe na solução considerada neste cenário, um ganho expressivo de tempo com relação à solução original, uma vez que o processo na solução considerada leva 7,58321 h a menos que na solução original.

O relevante ganho na preservação de levedura se deve possivelmente ao menor tempo de exposição ao etanol na solução considerada, uma vez que esta apresenta um tempo expressivamente menor de fermentação quando comparado ao da solução original.

Entretanto, a solução considerada exige uma vazão 2.593,26096 m³/h maior que a solução original, que é expressivamente maior, resultando em uma maior quantidade de substrato em um menor tempo no sistema.

O tomador de decisão pode avaliar se a implantação da solução considerada neste cenário traz um ganho relevante para o processo, avaliando a viabilidade de instalação do equipamento necessário para permitir a vazão requerida.

8.5.2.3.2 Cenário 2

O primeiro cenário elencado para este problema considerou o otimizador E2P2M3S3, descrito na Tabela 14, que apresenta um equilíbrio entre a geração de etanol e a preservação de levedura.

A Tabela 18 apresenta os valores das variáveis e dos componentes ao fim do processo fermentativo, tanto para a solução original quanto para a solução considerada neste cenário.

O Gráfico 67 apresenta os comportamentos das concentrações de cada um dos componentes ao longo do tempo no reator de fermentação alcoólica para solução original e para a solução considerada neste cenário.

Ao contrapor a solução original com a considerada neste cenário, verifica-se uma preservação de levedura, na solução considerada, de 29,55118 kg/m³ a mais que na solução original empregada no processo.

Com relação à produção etanólica, a solução considerada neste cenário apresenta

Gráfico 67 - Concentrações ao longo do tempo na solução original e na solução considerada no Cenário 2 levantado para o Problema B do Estudo de Caso 2.

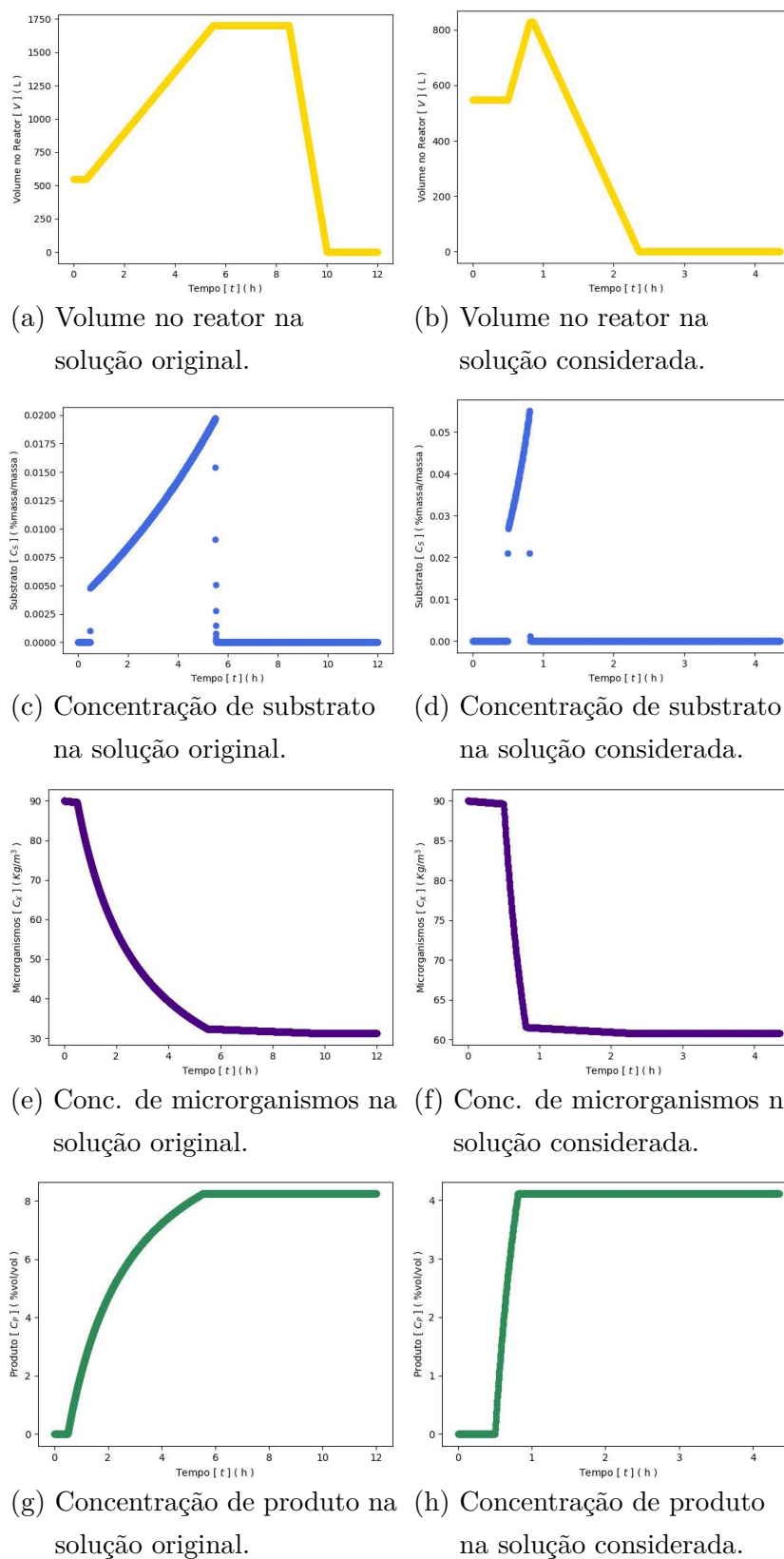


Tabela 18 - Valores das variáveis e objetivos referentes à solução original e a considerada no Cenário 2 levantado para o Problema B do Estudo de Caso 2.

Solução	Variáveis			Objetivos		
	t_B	t_D	$\nu_{F\downarrow}$	$C_S(t_H)$	$C_X(t_H)$	$C_P(t_H)$
Original	5,50000	8,50000	230,00000	0,00000	31,20690	8,23728
E2P2M3S3	0,80873	0,85785	906,31473	0,00000	60,75808	4,11099

Fonte: O autor, 2023.

uma perda de 4,12629 %vol/vol em relação a solução original, valor que se traduz em 32,56880 kg/m³ a menos de álcool.

O ganho de tempo da solução considerada neste cenário em relação à solução original é bastante expressivo, sendo na solução considerada cerca de 7,64215 h a menos que na solução original.

O ganho na preservação de levedura se deve ao baixo tempo de exposição ao etanol na solução considerada, uma vez que esta apresenta um tempo consideravelmente menor de fermentação quando comparado ao da solução original, além de uma menor diluição, uma vez que a quantidade de suco atingida no reator na solução considerada é menor que na original.

Entretanto, essa mesma configuração gera uma perda considerável na geração de etanol, fazendo-a cair pela metade, além da solução considerada neste cenário requerer uma vazão 676,31473 m³/h maior que a solução original, promovendo uma maior quantidade de substrato em um menor tempo de operação.

Como a solução considerada apresenta uma quantidade menor de suco no reator que a solução original, não cabe, neste cenário, uma comparação de valores absolutos.

Nota ao Leitor 27 Tanto para o Problema A quanto para o Problema B do Estudo de Caso 2, não foram elencado cenários que consideraram soluções que levam a maior preservação de levedura, pois em tais soluções, não existe nenhuma geração de etanol, fato que faz com que não exista qualquer interesse na adoção destas soluções, uma vez que o objetivo do processo é produzir álcool.

CONCLUSÃO

Mesmo que já tenhas feito uma longa caminhada, sempre haverá um caminho a percorrer.

Santo Agostinho

Um dos atributos que se espera de um processo é a eficiência, isto é, almeja-se que este processo cumpra seus objetivos de forma correta, com o mínimo de custos e máximo desempenho. Caso contrário, o processo pode apresentar desperdício de recursos e baixo rendimento, que em dadas proporções, podem inviabilizar o emprego do mesmo.

Para evitar a proposição de processos ineficientes ou aumentar a eficiência de processos existentes, emprega-se procedimentos de otimização a fim de encontrar as configurações que levam o processo a apresentar maior eficiência.

A importância dos processos modelados por equações diferenciais ordinárias desperta o interesse em seu estudo, solução e otimização, para os mais variados casos, nas mais variadas áreas, onde estes estão inseridos.

Neste sentido, a comunidade acadêmica volta-se predominantemente à solução e otimização de casos particulares deste tipo de problema, empregando em cada caso, abordagens e ferramentas distintas no processo.

Diante disso, percebe-se que carece de interesse, o desenvolvimento e proposição de metodologias mais gerais e unificadas capazes de promover a otimização deste tipo de problema sem se ater a especificidades destes.

Neste contexto, estão inseridos os problemas de fermentação, área cuja relevância motiva o desenvolvimento de um grande número de pesquisas e trabalhos, que buscam simular, otimizar, comparar e revisar os processos inerentes a esta área.

Um trabalho deste gênero é o realizado por Ani, Gujarathi e Vakili-Nezhaad (2021), que realizou estudos de caso envolvendo a otimização de um processo de fermentação em batelada alimentada de L-lisina.

Entretanto, existem produções científicas trabalham com parâmetros não otimizados, o que introduz a incerteza de que os resultados fornecidos pelos processos considerados possam não ser ótimos.

Este é o caso do trabalho realizado por Fonseca, Costa e Cruz (2017), que considera os dados e especificações de um processo fermentativo em batelada alimentada de uma planta de fermentação industrial para definir os parâmetros de suas simulações deste tipo de processo.

Síntese dos Resultados

Sintetizando, de forma breve, os resultados desta tese, temos que este trabalho desenvolveu e propôs um algoritmo para otimização de sistemas modelados por equações diferenciais ordinárias denominado CODEMSOA. Além de construir uma implementação computacional para este algoritmo na forma de biblioteca de código.

Foram realizados dois estudos de caso envolvendo a otimização de processos fermentativos. O primeiro considerou um processo de fermentação de L-lisina, enquanto o segundo considerou um processo de fermentação alcoólica.

No primeiro estudo de caso foram considerados dois problemas de otimização sobre os quais foram realizados seis simulações, três sobre cada um dos problemas, cada uma empregando um método de otimização diferente, a saber, o NSGA-II, AGE-MOEA e C-TAEA. Nestas simulações, foi usado o RK-2 para solução de EDOs e *Splines* Cúbicas para interpolação.

Nas referidas simulações, todos os métodos de otimização convergiram para resultados muito próximos uns dos outros, encontrando o número máximo de soluções não dominadas possíveis e apresentando tempos de execução e convergência bastante similares entre si.

Baseado nos resultados obtidos para o problema A do estudo de caso 1, constatou-se que valores maiores que 5 L para o volume inicial ocupado no reator não apresentam vantagem para nenhum dos objetivos.

Em todas as soluções, o instante de término da operação e o instante de término da alimentação do reator ficaram muito próximos do tempo máximo de operação a fim de que haja o maior tempo possível de injeção de suco no reator e de geração de L-lisina.

A variável que apresentou maior influência na relação entre os objetivos do problema foi o instante de início da alimentação do reator, pois quanto antes este se der, maior a quantidade de açúcares no reator, o que leva a um aumento na produtividade de L-lisina e conseqüentemente a uma queda no seu rendimento.

Por outro lado, quanto mais tarde for o instante de início da alimentação do reator, menor a quantidade de açúcares inserida neste, elevando assim o rendimento e conseqüentemente baixando a produtividade, uma vez que uma quantidade menor de L-lisina é produzida em um mesmo intervalo de tempo.

Deste modo, o conflito entre objetivos é tal que, quanto maior for a produtividade obtida, menor o rendimento alcançado e vice-versa, estabelecendo dessa forma o comportamento da curva de Pareto.

Considerando os resultados obtidos para o problema B do estudo de caso 1, verificou-se, assim como no problema anterior, que valores maiores que 5 L para o volume inicial ocupado no reator não apresentam vantagem para nenhum dos objetivos.

Em todas as soluções, o instante de início da alimentação está sempre muito próximo do início do processo, com intuito de garantir a maior quantidade de substrato no reator, no menor tempo possível a fim de fornecer a maior conversão de açúcares em L-lisina, o mais rápido possível.

O instante de término da alimentação do reator e o instante de término do processo são as variáveis de maior influência no comportamento da fronteira de Pareto, estando fortemente correlacionadas.

Assim, quanto maior o tempo de operação, maior o tempo de alimentação e conseqüentemente mais açúcares no reator, o que eleva a produtividade de L-lisina. Por outro lado, quanto menor o tempo de operação, menor a quantidade de açúcares, reduzindo a produtividade.

Os resultados obtidos nesta produção para ambos os problemas de otimização do estudo de caso 1 foram muito semelhantes aos obtidos pelo trabalho original do qual estes foram retirados.

No segundo estudo de caso foram propostos dois problemas de otimização, que assim como no primeiro estudo de caso, foram objetos de seis simulações, três sobre cada um dos problemas, cada uma empregando um método de otimização diferente, a saber, o NSGA-II, AGE-MOEA e C-TAEA. Nestas simulações, foi usado o RK-2 para solução de EDOs e *Splines* Cúbicas para interpolação.

Além destas, foi realizada uma primeira simulação através do método RK-2 para reproduzir computacionalmente o processo de fermentação que acontece na planta considerada, a fim de conhecer as concentrações dos componentes envolvidos no processo fermentativo sem nenhum tipo de otimização.

Nas simulações de otimização, todos os métodos empregados convergiram para resultados muito próximos uns dos outros. O AGE-MOEA e o NSGA-II encontraram o máximo de soluções não dominadas, enquanto o C-TAEA encontrou um número menor de soluções, apresentando tempos de convergência e execução variados.

Pelos resultados das simulações sobre o problema A do estudo de caso 2, verificou-se que o instante de término da alimentação é a variável que possui maior influência sobre os objetivos, pois regula a concentração de açúcares e a diluição da levedura no reator de fermentação.

Todos os otimizadores encontrados nestas simulações conduziram as concentrações de açúcares para zero, em virtude da quantidade de açúcares no reator ser diretamente proporcional ao tempo de alimentação, permitindo que as leveduras sempre consigam metabolizá-la por completo.

A partir da análise da fronteira de Pareto obtida, constatou-se o conflito entre os objetivos concentração de etanol e concentração de levedura, pois os microrganismos convertem os açúcares presentes no suco em álcool, mas, conforme a quantidade de etanol presente no suco vai aumentando, em função da ação fermentativa das leveduras, estas

vão morrendo devido a ação antisséptica do etanol.

Dessa forma, concluiu-se que quanto maior a quantidade de etanol no reator, ao final do processo, menor a concentração de células de biomassa presente no reator, sendo o inverso também verdadeiro.

Através dos resultados obtidos nas simulações realizadas sobre o problema B do estudo de caso 2, percebeu-se que a vazão de entrada apresenta a influência mais relevante, em virtude desta controlar diretamente a concentração de açúcares e diluição de biomassa no reator.

Também para este problema, todas as soluções zeraram as concentrações de açúcares, ao final do processo, uma vez que sempre existe tempo hábil e levedura suficiente para converter completamente os açúcares no mosto.

As soluções que possuem menores tempos de fermentação e menores vazões de entrada, apresentaram maior preservação de células de biomassa vivas e em contraparte, menor geração de álcool. Por outro lado, para maiores vazões de entrada, a produção de etanol é maior, existindo uma maior taxa de morte de células de levedura no vinho dentro do reator de fermentação.

Dentre soluções apresentam maiores tempos de fermentação, as com maiores tempos de batelada ou maiores vazões apresentam menores tempos de alimentação e as com maiores tempos de alimentação e de batelada possuem baixas vazões.

Assim como no problema anterior, quanto maior a quantidade de etanol no reator, ao final do processo, menor a concentração de células de biomassa presente no reator e vice-versa, sendo este o comportamento expresso pela frente de Pareto.

Para todos os problemas de ambos os estudos de caso, foram elencados cenários associados aos pontos do conjunto ótimo de Pareto de cada um, que levam aos melhores valores para cada um dos objetivos e que levam a um equilíbrio entre eles.

Contribuições

Esta tese buscou contribuir com a proposição de um método para otimização de processos modelados por equações diferenciais ordinárias, que consistiu em um algoritmo desenvolvido para este fim.

A fim de permitir o uso prático do algoritmo proposto em estudos e trabalhos, esta tese contribuiu também com o desenvolvimento de uma implementação computacional deste em uma única plataforma de programação, que pode ser usada pela comunidade na otimização de variados processos modelados por equações diferenciais ordinárias.

No mesmo sentido, esta produção acadêmica buscou contribuir com a comunidade científica através da proposição de problemas de otimização cujas definições foram fundadas nas possíveis vantagens que os resultados destes trariam para a área de fermentação

alcooleira, podendo estes problemas serem também empregados a outras configurações de plantas de fermentação do mesmo tipo da abordada aqui.

Produções

Como principais produções desta tese, pode-se elencar as que estão enunciadas e descritas:

O algoritmo para otimização de sistemas modelados por equações diferenciais ordinárias, que permite o uso de variados métodos de otimização, interpolação e de solução de equações diferenciais ordinárias.

A biblioteca de *software* desenvolvida integralmente em Python que implementa o algoritmo proposto através de uma abordagem multiparadigma, envolvendo a programação orientada a objetos e a programação funcional.

Os problemas de otimização propostos, que consistem em problemas de controle ótimo multiobjetivo cujas variáveis, restrições e objetivos envolvem os parâmetros inerentes ao modelo de reator de fermentação em batelada alimentada considerado no segundo estudo de caso deste trabalho.

Os resultados obtidos nos processos de otimização, suas análises e conclusões, cujos dados constituem valores de especificações e parâmetros otimizados, além de informações relevantes acerca destes.

Perspectivas Futuras

No curso do tempo, surgem novos métodos, novas tecnologias, técnicas são atualizadas e novos modelos são propostos. Do mesmo modo, no decorrer do tempo, o presente trabalho está sujeito a sofrer aprimoramentos, alterações, continuações e ramificações, amadurecendo e adaptando-se à realidade que se apresentar.

Ademais, existem inúmeras vertentes e novos caminhos que podem ser explorados a partir da presente tese. Neste sentido, ficam como sugestões para trabalhos futuros a este, as seguintes proposições:

Realização de uma análise que busque identificar novos problemas em outras áreas do conhecimento, em que seja possível o uso do algoritmo proposto e da ferramenta computacional desenvolvida para gerar novos estudos e trabalhos.

Acrescer novos métodos de otimização à biblioteca de *software* desenvolvida com intuito de aumentar a variabilidade de métodos suportados pela ferramenta, ampliando as opções dos eventuais usuários da mesma.

Usar a metodologia desenvolvida para otimizar outros modelos de reatores de fer-

mentação, que envolvam novos componentes ou especificações dos que foram apresentados nos estudos de caso realizados neste trabalho.

Otimizar reatores de fermentação contínua, fazendo uso do método de solução apresentado neste tese, pois, ainda que usados industrialmente em menor escala, possuem relevante interesse por parte da academia e da indústria.

Aplicar os problemas de otimização propostos em novos estudos de caso que envolvam o mesmo modelo considerado, mas apresentem diferentes configurações em suas especificações e configurações, permitindo avaliar e comparar os diferentes estudos de caso abordados.

Elaborar novos problemas de otimização sobre os processos fermentativos trabalhados, considerando diferentes conjuntos de variáveis e de objetivos de otimização, a fim de ampliar as perspectivas e análises sobre os processos em questão, no que tange à otimização.

Empregar outros métodos de otimização para resolução dos problemas propostos neste trabalho e promover um estudo comparativo entre os métodos utilizados, com objetivo de aferir a acurácia, vantagens e desvantagens de cada um com relação à sua aplicação neste tipo de problema.

Além das propostas de trabalhos futuros apresentadas acima, qualquer trabalho originado desta tese que apresente contribuição relevante para o algoritmo proposto, para o sistema desenvolvido, para modelo fermentativo considerado ou para outros correlatos a este, pode ser considerado uma extensão da presente tese.

REFERÊNCIAS

- ABRAHAM, A.; JAIN, L.; GOLDBERG, R. *Evolutionary multi-objective optimization: theoretical advances and applications*. 1. ed. New York, NY: Springer, 2005.
- AGUIRRE, L. A. *Introdução a identificação de sistemas: técnicas lineares e não lineares a sistemas reais*. 1. ed. Belo Horizonte, MG: Editora UFMG, 2004.
- AIBA, S.; HUMPHREY, A. E.; MILLIS, N. F. *Biochemical engineering*. 2. ed. New York, NY: Academic Press Inc., 1973.
- AIBA, S.; SHODA, M.; NAGATANI, M. Kinetics of product inhibition in alcohol fermentation. *Biotechnology and Bioengineering*, v. 10, n. 1, p. 845–864, 1968.
- AL-SIYABI, B.; GUJARATHI, A. M.; SIVAKUMAR, N. Harmonic multi-objective differential evolution approach for multi-objective optimization of fed-batch bioreactor. *Materials and Manufacturing Processes*, v. 10, n. 1, p. 1152–1161, 2017.
- AN, G. H.; SONG, K. B.; SINSKEY, A. J. Redirection of carbon flux to lysine in a recombinant of corynebacterium lactofermentum atcc 21799 by limited supply of pantothenate. *Journal of Bioscience and Bioengineering*, v. 88, n. 2, p. 168–172, 1999.
- ANASTASSIADIS, S. L-lysine fermentation. *Resent Patents on Biotechnology*, v. 1, n. 1, p. 11–24, 2007.
- ANDERSSON, J. *A survey of multiobjective optimization in engineering design*. Linköping, SW, 2001.
- ANDRETTA, M. *Fundamentos de otimização irrestrita*. 1. ed. São Carlos, SP: Departamento de Matemática, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2010.
- ANDRETTA, M. *Fundamentos de otimização restrita*. 1. ed. São Carlos, SP: Departamento de Matemática, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2010.
- ANDRETTA, M. *Introdução à otimização*. 1. ed. São Carlos, SP: Departamento de Matemática, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2010.
- ANDRETTA, M. *Problemas de programação linear*. 1. ed. São Carlos, SP: Departamento de Matemática, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2016.
- ANDRIETTA, S. R. *Modelagem, simulação e controle de fermentação alcoólica contínua em escala industrial*. Tese (Doutorado) — Universidade Estadual de Campinas, Campinas, SP, 1994.
- ANI, Z. A.; GUJARATHI, A. M.; VAKILI-NEZHAAD, G. Hybridized multi-objective optimization approach (hmode) for lysine fed-batch fermentation process. *Korean Journal of Chemical Engineering*, v. 38, n. 1, p. 8–21, 2021.

- ARENALES, M. N. *et al. Pesquisa operacional*. 1. ed. Rio de Janeiro, RJ: Elsevier, 2007.
- BACK, T.; FOGEL, D. B.; MICHALEWICZ, T. *Evolutionary computation: basic algorithms and operators*. 1. ed. Nova York, NY: Taylor & Francis Group, 2000.
- BAILEY, J. E.; OLLIS, D. F. *Biochemical engineering fundamentals*. 2. ed. New York, NY: McGraw Hill, 1986.
- BAZARAA, M. S.; JARVIS, J. J. *Linear programming and network flows*. 1. ed. Nova York, NY: John Wiley and Sons, 1977.
- BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. *Nonlinear programming: theory and algorithms*. 3. ed. New York, NY: Wiley, 2006.
- BINDER, M. *Liderando a produção sustentável de lisina*. 2020. Disponível em: <https://sindiavipar.com.br/artigo-liderando-a-producao-sustentavel-de-lisina/>. Acesso em: 10 ago. 2023.
- BONOMI, A.; SCHMIDELL, W. *Modelagem matemática e simulação de processos fermentativos*. 2. ed. São Paulo, SP: Edgard Blucher, 2001.
- BORZANI, W. Fermentação alcoólica contínua. *Anais da Semana da Fermentação Alcoólica de Piracicaba*, v. 1, n. 1, p. 263–276, 1960.
- BOTT, M. Offering surprises: Tca cycle regulation in corynebacterium glutamicum. *Trends in Microbiology*, v. 15, n. 9, p. 417–425, 2007.
- BRASIL, R. M. L. R. F.; SILVA, M. A. *Otimização de projetos de engenharia*. 1. ed. São Paulo, SP: Blucher, 2019.
- CARVALHO, A. P. L. F. *et al. Computação bioinspirada*. 1. ed. Porto Alegre, RS: Sociedade Brasileira de Computação (SBC), 2004.
- CARVALHO, J. C. M.; SATO, S. *Fermentação descontínua*. 1. ed. São Paulo, SP: Edgard Blucher, 2001. v. 2.
- CASTRO, L. N. *Fundamentals of natural computing: basic concepts, algorithms and applications*. 1. ed. Boca Raton, FL: Chapman & Hall/CRC, 2006.
- CASTRO, N. M. *Etanol hidratado é negociado por valores abaixo do custo de produção*. 2021. Disponível em: <https://www.novacana.com/n/etanol/mercado/precos/etanol-hidratado-negociado-valores-abaixo-custo-producao-230420>. Acesso em: 18 jun. 2022.
- CHEMICAL ENGINEERING. *Technology profile: L-lysine hcl production from glucose*. 2016. Disponível em: <https://www.chemengonline.com/l-lysine-hcl-production-glucose>. Acesso em: 10 ago. 2023.
- CHEN, Z. *et al.* Deregulation of feedback inhibition of phosphoenolpyruvate carboxylase for improved lysine production in corynebacterium glutamicum. *Applied and Environmental Microbiology*, v. 80, n. 4, p. 1388–1393, 2014.
- CHICONE, C. *Ordinary differential equations with applications*. 1. ed. New York, NY: Springer, 2006.

- CNPEN. *Fermentação: Contínua ou em batelada?* 2017. Disponível em: <https://cnpem.br/fermentacao-continua-ou-em-batelada>. Acesso em: 10 fev. 2023.
- COELLO, C. A. C. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, v. 1, n. 3, p. 269–308, 1999.
- COELLO, C. A. C. Evolutionary multi-objective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine*, v. 1, n. 1, p. 28–36, 2006.
- COELLO, C. A. C.; VELDHUIZEN, D. A. V.; LAMONT, G. B. *Evolutionary algorithms for solving multi-objective problems: genetic algorithms and evolutionary computation*. 1. ed. Nova York, NY: Kluwer Academic, 2002.
- COLLETTE, Y.; SIARRY, P. *Multiobjective optimization: principles and case studies*. 1. ed. Heidelberg, BE: Springer, 2003.
- CORNE, D.; KNOWLES, J.; OATES, M. *The Pareto envelope-based selection algorithm for multiobjective optimization*. 1. ed. Boston, MA: Springer, 2000.
- CORREIA, A.; MATIAS, J.; SERODIO, C. Métodos de penalidade exacta para resolução de problemas de otimização não linear. *Core*, v. 28, n. 1, p. 17–30, 2008.
- DARKWAH, K.; KNUTSON, B. L.; SEAY, J. R. Multi-objective versus single-objective optimization of batch bioethanol production based on a time-dependent fermentation model. *Clean Technologies and Environmental Policy*, v. 20, n. 1, p. 1271–1285, 2018.
- DARMAYANTI, R. F. *et al.* Extractive biobutanol fermentation using immobilized cells with organic acid esters as novel co-substrates in large extractant volume. *Bioresource Technology Reports*, v. 23, n. 1, p. 1–11, 2023.
- DARWIN, C. R. *A origem das espécies*. 1. ed. Porto, PT: Lello & Irmão Editores, 1959.
- DAS, S.; SUGANTHAN, P. N. Differential evolution: a survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, v. 15, n. 1, p. 4–31, 2011.
- DEB, K. *Multi-objective optimization using evolutionary algorithms*. 1. ed. Nova York, NY: John Wiley & Sons, 2001.
- DEB, K. *et al.* *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii*. Kanpur, KP, 2000.
- DEB, K.; JAIN, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, v. 18, n. 4, p. 577–601, 2014.
- DEB, K.; JAIN, H. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, v. 18, n. 4, p. 602–622, 2014.
- DEB, K. *et al.* A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182–197, 2002.

- DELBEM, A. C. B.; CARVALHO, A. C. P. L. F.; BRETAS, N. G. Main chain representation for evolutionary algorithm applied to distribution system reconfiguration. *IEEE Transactions on Power Systems*, v. 20, n. 1, p. 425–436, 2005.
- DIEHL, M. *et al. Recent advances in optimization and its applications in engineering*. 1. ed. Berlin, BE: Springer Berlin Heidelberg, 2014.
- DORAN, P. M. *Bioprocess engineering principles*. 1. ed. London, LD: Academic Press, 1995.
- EDGEWORTH, F. Y. *Mathematical physics*. 1. ed. London, UK: Charles Kegan Paul & Co., 1881.
- EDUARDO, J. H. P. Fermentação contínua. *Stab: Açúcar, Álcool e Subprodutos*, v. 9, n. 4/5, p. 42–45, 1991.
- EGGELING, L.; BOTT, M. A giant market and a powerful metabolism: L-lysine provided by corynebacterium glutamicum. *Applied Microbiology and Biotechnology*, v. 99, n. 1, p. 3387–3394, 2015.
- EIBEN, A. E.; KEMENADE, C. H. V.; KOK, J. N. *Orgy in the computer: multi-parent reproduction in genetic algorithms*. 1. ed. Nova York, NY: Springer, 1995.
- EIBEN, A. E.; SCHIPPERS, C. A. *Multi-parent's niche: n-ary crossovers on nk-landscapes*. 1. ed. Nova York, NY: Springer, 1996.
- EIBEN, A. E.; SMITH, J. E. *Introduction to evolutionary computing: natural computing series*. 1. ed. Berlin, BE: Springer, 2003.
- EVANS, G. W. An overview of techniques for solving multiobjective mathematical programs. *Management Science*, v. 30, n. 11, p. 1268–1387, 1984.
- FACCIOTTI, M. C. R. *Fermentação contínua*. 1. ed. São Paulo, SP: Edgard Blucher, 2001.
- FERRAZ, B. A. *Métodos computacionais de otimização*. 1. ed. Ponta Grossa, PR: Atena, 2021.
- FERREIRA, P. A. V. *Otimização multiobjetivo: teoria e aplicações*. 1. ed. Campinas, SP: Universidade Estadual de Campinas, 1999.
- FLETCHER, R. *Practical methods of optimization*. 2. ed. New York, NY: Wiley, 2000.
- FOGEL, D. B. An introduction to simulated evolution. *IEEE Transaction on Neural Networks*, v. 5, n. 1, p. 3–14, 1994.
- FOGEL, L. Autonomus automata. *Industrial Research*, v. 4, n. 1, p. 14–19, 1962.
- FONSECA, C.; FLEMING, P. *Genetic algorithms for multiobjective optimization: formulation, discussion and generalization*. 1. ed. San Mateo, CA: Morgan Kauffman Publishers, 1993.

- FONSECA, G. C.; COSTA, C. B. B.; CRUZ, A. J. G. Comparing a dynamic fed-batch and a continuous steady-state simulation of ethanol fermentation in a distillery to a stoichiometric conversion simulation. *Brazilian Journal of Chemical Engineering*, v. 34, n. 4, p. 1121–1131, 2017.
- FULCHER, J. *Computational intelligence: a compendium*. 1. ed. New York, NY: Springer-Verlag, 2008.
- GABRIEL, P. H. R.; DELBEM, A. C. B. *Fundamentos de algoritmos evolutivos*. 1. ed. São Carlos, SP: ICMC/USP, 2008.
- GABRIEL, P. H. R.; DELBEM, A. C. B. *Fundamentos de algoritmos evolutivos*. 1. ed. São Carlos, SP: Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2008.
- GASPAR-CUNHA, A.; TAKAHASHI, R.; ANTUNES, C. H. *Manual de computação evolutiva e meta heurística*. 1. ed. Coimbra, PT: Coimbra University Press, 2012.
- GHOSE, T. K.; TYAGI, R. D. Rapid ethanol fermentation of cellulose hydrolysate ii: product and substrate inhibition and optimization of fermentor design. *Biotechnology and Bioengineering*, v. 21, n. 8, p. 1401–1420, 1979.
- GLOVER, F.; KOCHENBERGER, G. A. *Handbook of metaheuristics*. 1. ed. Boston, MA: Kluwer, 2003.
- GODOY, A.; HENRIQUE, V. A.; LOPES, M. L. Continuous and batch fermentation processes: advantages and disadvantages of these processes in the brazilian ethanol production. *International Sugar Journal*, v. 110, n. 121, 2008.
- GOLDBERG, D. E. *Genetic algorithms in search, optimization and machine learning*. 1. ed. Boston, MA: Addison-Wesley Longman Publishing, 1989.
- GOLDBERG, D. E. *The design of innovation: lessons from and for competent genetic algorithms*. 1. ed. Boston, MA: Kluwer, 2002.
- GUIMARAES, I.; SOUZA, M. J. F.; JUNIOR, G. P. Modelo de simulação computacional para validação dos resultados de alocação dinâmica de caminhos com atendimento de metas de qualidade e de produção em minas a céu aberto. *Proceedings do XIV Simpósio de Engenharia de Produção (SIMPEP)*, v. 1, n. 1, p. 263–282, 2007.
- GUJARATHI, A. M.; PATEL, S. P.; SIYABI, B. A. Insight into evolutionary optimization approach of batch and fed-batch fermenters for lactic acid production. *Digital Chemical Engineering*, v. 8, n. 1, p. 1–13, 2023.
- HAESER, G. *Condições de otimalidade de primeira e segunda ordem em otimização não linear*. 1. ed. São Paulo, SP: Departamento de Matemática Aplicada, Instituto de Matemática e Estatística, Universidade de São Paulo, 2015.
- HAN, K.; LEVENSPIEL, O. Extend monod kinetics for substrate, product and cell inhibition. *Biotechnology and Bioengineering*, v. 32, n. 1, p. 430–437, 1988.
- HAND, R. *L-lysine and L-arginine amino acid ratios in food diet*. 2015. Disponível em: https://www.traditionaloven.com/tutorials/l-lysine_amino_acid.html. Acesso em: 10 ago. 2023.

- HARINGA, C. *et al.* Computational fluid dynamics simulation of an industrial p. chrysogenum fermentation with a coupled 9-pool metabolic model: towards rational scale-down and design optimization. *Chemical Engineering Science*, v. 175, n. 1, p. 12–24, 2018.
- HENDLER, B. *Construção de um software de simulação e modelagem para processo de fermentação alcoólica em batelada alimentada, considerando diauxia para a glicose e a frutose*. Dissertação (Mestrado) — Faculdade de Engenharia Química, Universidade Estadual de Campinas, Campinas, SP, 2011.
- HIRA, A.; OLIVEIRA, L. G. No substitute for oil? how brazil developed its ethanol industry. *Energy Policy*, v. 37, n. 1, p. 2450–2456, 2009.
- HOLLAND, J. H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. 1. ed. Ann Arbor, MI: University of Michigan Press, 1975.
- HOLLANDER, J. A. Potential methabolic limitations in lysine production by corynebacterium glutamicum as revealed by metabolic network analysis. *Applied Microbiology Biotechnology*, v. 42, n. 1, p. 508–515, 1994.
- HOPCROFT, J. E.; MOTWANI, R.; ULLMAN, J. D. *Introduction to automata theory, languages and computation*. 3. ed. Noida, DAR: Pearson, 2008.
- HORN, J.; NAFPLIOTIS, N.; GOLDBERG, D. *A niched Pareto genetic algorithm for multiobjective optimization*. 1. ed. Piscataway, NJ: IEEE Service Center, 1994.
- HORTA, L. A. N. *Perspectivas de un programa de biocombustibles em américa central*. 1. ed. Cidade do México, CDMX: Proyecto Cepal/GTZ uso Sustentable de Hidrocarburos, Comisión Económica para América Latina y el Caribe, 2004.
- HWANG, C. L. *et al.* *Multiple objective decision making methods and applications*. 1. ed. Berlin, BE: Springer, 1979.
- ITABORAHY FILHO, M. A. *Análise de algoritmos genéticos e evolução diferencial para otimização de funções não-lineares multimodais*. Monografia (Graduação) — Departamento de Elétrica, Universidade Tecnológica Federal do Paraná. Ponta Grossa, PR, 2018.
- IZMAILOV, A.; SOLODOV, M. *Otimização: condições de otimalidade, elementos de análise convexa e de dualidade*. 3. ed. Rio de Janeiro, RJ: Instituto de Matemática Pura e Aplicada, 2014. v. 1.
- JIN, Y. F. *et al.* Immunological activity of bagasse polysaccharides. *Acta Pharmacologica Sinica*, v. 2, n. 4, p. 269–275, 1981.
- JONG, K. A. *Evolutionary computation: a unified approach*. 1. ed. Cambridge, MA: MIT Press, 2006.
- JUNIOR, L. A. L.; LETTI, G. V. M.; SOCCOL, C. R. Development of an l-lysine enriched bran for animal nutrition via submerged fermentation by corynebacterium glutamicum using agroindustrial substrates. *Brazilian Archives of Biology and Technology*, v. 59, n. 1, p. 1–8, 2016.

- KHARE, V. *Performace scaling of multi-objective evolutionary algorithms*. Dissertação (Mestrado) — University of Birmingham, Birmingham, UK, 2002.
- KIEFER, P.; HEINZLE, E.; WITTMANN, C. Influence of glucose, fructose and sucrose as carbon sources on kinetics and stoichiometry of lysine production by corynebacterium glutamicum. *Journal of Industrial Microbiology & Biotechnology*, v. 28, n. 1, p. 338–343, 2002.
- KIRCHER, M.; PFEFFERLE, W. The fermentative production of l-lysine as an animal feed additive. *Chemosphere*, v. 43, n. 1, p. 27–31, 2001.
- KOZA, J. R. *Hierarchical genetic algorithms operating on population of computer programs*. 1. ed. Detroid, MI: Morgan Kaufmann, 1989.
- KOZA, J. R. *Genetic programming: on the programming of computers by means of natural selection*. 1. ed. Cambridge, MA: MIT Press, 1992.
- KOZA, J. R. *Genetic programming II: automatic discovery of reusable programs*. 1. ed. Cambridge, MA: MIT Press, 1994.
- KOZA, J. R. *et al. Genetic programming III: darwinian invention and problem solving*. 1. ed. San Mateo, CA: Morgan Kaufmann, 1999.
- KOZA, J. R. *et al. Genetic programming IV: routine human-competitive machine intelligence*. 1. ed. Dordrecht, SH: Kluwer Academic Publishers, 2003.
- KRISHNAKUMAR, K. Micro-genetice algorithms for stationary and non-stationary function optimization. *SPIE: Intelligent Control and Adaptative Systems*, v. 1196, n. 1, p. 289–296, 1989.
- LARRAENAGA, P.; LOZANO, J. *Estimation of distribution algorithms: a new tool for evolutionary computation*. 1. ed. Nova York, NY: Kluwer Academic Publishers, 2001.
- LAUMANN, M. *et al. Combining convergence and diversity in evolutionary multiobjective optimization. The Third China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems*, v. 10, n. 3, p. 263–282, 2002.
- LEE, J. H.; PAGAN, R. J.; ROGERS, P. L. Continuous simultaneous saccharification and fermentation of starch using zymomonas mobilis. *Biotechnology and Bioengineering*, v. 25, n. 3, p. 659–669, 1983.
- LETT, L. A. J. *Produção de L-lisina por processos fermentativos e desenvolvimento de produto para nutrição animal*. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba, PR, 2014.
- LEVENSPIEL, O. The monod equation: a revisit and a generalization to product inhibition situations. *Biotechnology and Bioengineering*, v. 22, n. 8, p. 1671–1687, 1980.
- LI, K. *et al. Achieving balance between proximity and diversity in multi-objective evolutionary algorithm*. v. 182, n. 1, p. 220–242, 2012.
- LIMA, L. R.; MARCONDES, A. A. *Álcool carburante: uma estratégia brasileira*. 1. ed. Curitiba, PR: Editora UFPR, 2002.

- LIMA, U. A.; BASSO, L. C.; AMORIM, H. V. *Biotecnologia industrial: processos fermentativos e enzimáticos*. 1. ed. São Paulo, SP: Edgard Blucher, 2001.
- LINDEN, R. *Algoritmos genéticos*. 2. ed. Rio de Janeiro, RJ: Brasport, 2008.
- LIU, J. *et al.* Industrial production of l-lysine in *Corynebacterium glutamicum*: progress and prospects. *Microbiological Research*, v. 262, n. 1, p. 1–13, 2022.
- LOBATO, F. S. *Otimização multi-objetivo para o projeto de sistemas de engenharia*. Tese (Doutorado) — Universidade Federal de Uberlândia, Uberlândia, MG, 2008.
- LUKE, S. *Essentials of metaheuristics*. 2. ed. Virginia, VA: Department of Computer Science, George Mason University, 2016.
- LUONG, J. H. T. Kinetics of ethanol inhibition in alcohol fermentation. *Biotechnology and Bioengineering*, v. 27, n. 1, p. 280–285, 1985.
- MAIORELLA, B. L.; BLANCH, H. W.; WILKE, C. R. *Alcohol production and recovery*. 1. ed. Heidelberg, AL: Springer Verlag, 1981. v. 20.
- MARTINEZ, S. Z. *et al.* Using a family of curves to approximate the pareto front of a multi-objective optimization problem. *International Conference on Parallel Problem Solving from Nature*, v. 1, n. 1, p. 682–691, 2014.
- MCLURE, M. *Pareto, economics and society: the mechanical analogy*. 1. ed. London, LD: Routledge, 2001.
- MELO, V. V. *et al.* Improving global numerical optimization using a search-space reduction algorithm. *Genetic and Evolutionary Computation Conference, ACM*, v. 1, n. 1, p. 1195–1202, 2007.
- MHLENBEIN, H. The equation for response to selection and its use for prediction. *Evolutionary Computation*, v. 5, n. 3, p. 303–346, 1998.
- MHLENBEIN, H.; PAA, G. From recombination of genes to the estimation of distributions: binary parameters. *Parallel Problem Solving from Nature (PPSN)*, v. 1411, n. 4, p. 178–187, 1996.
- MICHALEWICZ, Z. *Genetic algorithms + data structures = evolution programs*. 1. ed. Berlin, BE: Springer, 1996.
- MICHALEWICZ, Z.; JANIKOW, C. Z. Handling constraints in genetic algorithms. *Proceedings of the Fourth International Conference on Genetic Algorithms*, v. 1, n. 1, p. 151–157, 1991.
- MIETTINEN, K. *On the methodology of multiobjective optimization with applications*. 1. ed. Jyväskylä, CF: Universitat Jyväskylä, 1994.
- MIETTINEN, K. *Nonlinear multiobjective optimization*. 1. ed. Boston, MA: Kluwer Academic Publishers, 1999.
- MORALES, A. F. K. Pattern recognition via vasconcelo’s genetic algorithm. *Computer Science, Springer Berlin / Heidelberg*, v. 3287, n. 1, p. 328–335, 2004.

- MORALES, A. F. K.; GARCIA, J. G. *Penalty functions methods for constrained optimization with genetic algorithms: a statistical analysis*. 1. ed. Heidelberg, HDB: Springer-Verlag, 2001.
- NELSON, D. L.; COX, M. M. *Princípios de bioquímica*. 5. ed. São Paulo, SP: Sarvier, 2011.
- NOVA CANA. *Processos de fabricação do etanol*. 2020. Disponível em: <https://www.novacana.com/etanol/fabricacao>. Acesso em: 10 fev. 2023.
- NOVA CANA. *Aplicações e usos do etanol*. 2021. Disponível em: <https://www.novacana.com/noticias/aplicacoes>. Acesso em: 16 mai. 2023.
- OHNO, H.; NAKANISHI, E. Optimal control of a semibatch fermentation. *Biotechnology and bioengineering*, v. 18, n. 1, p. 847–864, 1976.
- PACHECO, T. F. *Fermentação alcoólica com leveduras de características floculantes em reator tipo torre com escoamento ascendente*. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Uberlândia, MG, 2010.
- PAIVA, F. A. P. de. *Otimização por enxames de partículas: usando uma adaptação de serendipidade*. 1. ed. Natal, RN: IFRN, 2018.
- PANICHELLA, A. An adaptive evolutionary algorithm based on non-euclidean geometry for many-objective optimization. *Proceedings of the Genetic and Evolutionary Computation Conference*, v. 19, n. 3, p. 595–603, 2019.
- PARETO, V. *Cours d'économie politique*. 1. ed. Lausanne, CH: Rouge, 1896.
- PATERNINA, L. M. C. *Modelagem e simulação do processo de fermentação extrativa a vácuo com uma câmara de flash e separação do CO₂ utilizando uma coluna de absorção*. Dissertação (Mestrado) — Faculdade de Engenharia Química, Universidade Estadual de Campinas, Campinas, SP, 2011.
- PAUL, T. K.; IBA, H. Linear and combinatorial optimization by estimation of distribution algorithms. *Proceedings of the 9th MPS Symposium on Evolutionary Computation*, 2003.
- PELIKAN, M.; GOLDBERG, D. E.; LOBO, F. *A survey of optimization by building and using probabilistic models*. Urbana, IL, 1999.
- PEREIRA, A. I. *et al.* *Optimization: learning algorithms and applications*. 1. ed. New York, NY: Springer, 2021.
- PETROLI NETO, S. *Computação evolutiva: desvendando os algoritmos genéticos*. 1. ed. Itatiba, SP: Universidade São Francisco, 2011.
- PFEIFER, A. A.; LOBATO, F. S. Controle ótimo de sistemas algébrico-diferenciais chaveados usando o algoritmo de busca gravitacional. *Trends in Computational and Applied Mathematics*, v. 14, n. 1, p. 81–94, 2013.
- PORTO, L. M. *Modelagem de processo industrial de fermentação alcoólica contínua com reatores de mistura ligados em série*. Tese (Doutorado) — Universidade Estadual de Campinas, Campinas, SP, 2005.

- PURSHOUSE, R. C.; FLEMING, P. J. On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation*, v. 11, n. 6, p. 770–784, 2007.
- RAMPAZZO, P. C. B. *Planejamento hidrelétrico: otimização multiobjetivo e abordagens evolutivas*. Tese (Doutorado) — Universidade Estadual de Campinas, Campinas, SP, 2012.
- RAO, S. S. *Engineering optimization: theory and practice*. 4. ed. New York, NY: Wiley, 2009.
- RECHENBERG, I. *Cybernetic solution path of an experimental problem*. Franborough, UK, 1965.
- REZENDE, S. O. *Sistemas inteligentes: fundamentos e aplicações*. 1. ed. Barueri, SP: Malone, 2003.
- RFA. *Annual ethanol production: Us and world ethanol production*. 2022. Disponível em: <https://ethanolrfa.org/markets-and-statistics/annual-ethanol-production>. Acesso em: 10 fev. 2023.
- RIBEIRO, A. A.; KARAS, E. W. *Um curso de otimização*. 1. ed. Curitiba, PR: Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, 2010.
- ROBERTS, C. E. *Ordinary differential equations: applications, models, and computing*. 1. ed. Boca Raton, FL: Chapman and Hall/CRC, 2010.
- ROCHA, N. C.; SARAMAGO, S. F. P. *Estudo de algumas estratégias da evolução diferencial*. 3. ed. Uberlândia, MG: Congresso de Matemática Aplicada e Computacional (SBMAC), 2011.
- ROSARIO, R. R. L. *Algoritmos evolutivos adaptativos para problemas de programação de pessoal*. Tese (Doutorado) — Universidade Federal de Santa Catarina, Florianópolis, SC, 2011.
- SABIONI, C. L. *Desenvolvimento de métodos para solução de problemas de otimização multiobjetivo com incertezas*. Tese (Doutorado) — Universidade Federal de Minas Gerais, Belo Horizonte, MG, 2017.
- SANTHARAM, L. *et al.* Modeling of exo-inulinase biosynthesis by *kluveromyces marxianus* in fed-batch mode: correlating production kinetics and metabolic heat fluxes. *Applied Microbiology and Biotechnology*, v. 101, n. 1, p. 1877–1887, 2017.
- SANTOS, W. R. L. *Algoritmos evolutivos para otimização binária*. Monografia (Graduação) — Departamento de Elétrica, Universidade Tecnológica Federal do Paraná. Ponta Grossa, PR, 2019.
- SARKAR, D.; MODAK, J. M. Pareto-optimal solutions for multi-objective optimization of fed-batch bioreactors using nondominated sorting genetic algorithm. *Chemical Engineering Science*, v. 60, n. 1, p. 481–492, 2005.

- SCHAFFER, J. Multiple objective optimization with vector evaluated genetic algorithms. *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms, Lawrence Erlbaum*, v. 1, n. 1, p. 93–100, 1985.
- SCHMIDELL, W.; FACCIOTTI, M. C. R. *Biorreatores e processos fermentativos*. 1. ed. São Paulo, SP: Edgard Blucher, 2001.
- SCHWEFEL, H. P. *Evolutions strategie and numerishe optimierung*. Tese (Doutorado) — Technical University of Berlin, Berlin, BE, 1975.
- SEVELY, Y. *et al.* Fermentation model. *8th IFAC Triennial Congress*, v. 1, n. 1, 1980.
- SHAH, A. H.; HAMEED, A.; KHAN, G. M. Fermentative production of l-lysine: bacterial fermentation. *International Journal of Medical Sciences*, v. 2, n. 3, p. 152–157, 2002.
- SHARMA, S. *et al.* Development of yeast aerobic granules for long-term continuous bioethanol production from rice straw hydrolysate. *Fuel*, v. 351, n. 1, p. 1–9, 2023.
- SILVA, R. C. P. *Um estudo sobre a auto adaptação de parâmetros na evolução diferencial*. Monografia (Graduação) — Universidade Federal de Ouro Preto. Ouro Preto, MG, 2010.
- SOUZA, C. S. *Avaliação da produção de etanol em temperaturas elevadas por linhagem de Saccharomyces Cerevisiae*. Tese (Doutorado) — Universidade de São Paulo, São Paulo, SP, 2009.
- SOUZA, M.; EHRHARDT, M. A. D. *Otimização e análise convexa: aspectos teóricos e aplicações*. 1. ed. Campinas, SP: Departamento de Matemática Aplicada, Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, 2011.
- SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, v. 2, n. 3, p. 221–248, 1995.
- SRIVASTAVA, S. *Top 10 data science programming languages for 2020*. 2021. Disponível em: <https://www.analyticsinsight.net/top-10-data-science-programming-languages-for-2020>. Acesso em: 18 mai. 2023.
- STEUER, R. E. *Multiple criteria optimization: theory, computation and application*. 1. ed. Nova York, NY: Wiley, 1986.
- STORN, R.; PRICE, K. *Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces*. Berkeley, CA, 1995.
- STORN, R.; PRICE, K. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization. Kluwer Academic Publishers*, v. 11, n. 1, p. 341–359, 1997.
- STORN, R.; PRICE, K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. *Journal of Global Optimization*, v. 11, n. 1, p. 341–359, 1997.

- STREMEL, D. P. *Desenvolvimento de modelos estruturados alternativos para o processo de produção de etanol*. Tese (Doutorado) — Universidade Estadual de Campinas, Campinas, SP, 2001.
- TA, D.-T. *et al.* High production of poly(3-hydroxybutyrate) in *Escherichia coli* using crude glycerol. *Bioresource Technology*, v. 384, n. 1, p. 1–8, 2023.
- TAKAHASHI, R. H. C. *Otimização escalar e vetorial*. 1. ed. Belo Horizonte, MG: Departamento de Matemática, Universidade Federal de Minas Gerais, 2007.
- TANABE, R.; OYAMA, A. Benchmarking moeas for multi and many objective optimization using an unbounded external archive. *Proceedings of the Genetic and Evolutionary Computation Conference*, v. 1, n. 1, p. 633–640, 2017.
- THAKUR, H. *et al.* Anaerobic co-digestion of food waste, bio-flocculated sewage sludge, and cow dung in cstr using e(c2)tx synthetic consortia. *Environmental Technology & Innovation*, v. 32, n. 1, p. 1–19, 2023.
- THANOSHAN, M. V. *What exactly is a programming paradigm?* 2021. Disponível em: <https://www.freecodecamp.org/news/what-exactly-is-a-programming-paradigm/>. Acesso em: 18 mai. 2023.
- THATIPALAMA, R.; ROHANI, S.; HILL, G. A. Effects of high product and substrate inhibitions on the kinetics and biomass and product yields during ethanol batch fermentation. *Biotechnology and Bioengineering*, v. 40, n. 1, p. 289–297, 1992.
- TIAN, Y. *et al.* An indicatorbased multiobjective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Transactions on Evolutionary Computation*, v. 22, n. 4, p. 609–622, 2018.
- TIAN, Y. *et al.* Guiding evolutionary multi-objective optimization with generic front modeling. *IEEE Transactions on Cybernetics*, v. 1, n. 1, p. 1–14, 2018.
- TORIDE, Y. *Lysine and other amino acids for feed: production and contribution to protein utilization in animal feeding*. 2016. Disponível em: <https://www.fao.org/3/y5019e/y5019e0a.htm>. Acesso em: 10 ago. 2023.
- TOSETTO, G. M. *Influencia da materia-prima no comportamento cinético de levedura na produção de etanol*. Dissertação (Mestrado) — Faculdade de Engenharia Química, Universidade Estadual de Campinas, Campinas, SP, 2002.
- UNICA. *Indústria brasileira de cana-de-açúcar: uma trajetória de evolução*. 2012. Disponível em: <http://old.unica.com.br/linhadotempo/index.html>. Acesso em: 10 fev. 2023.
- VAN, J. F. I. *et al.* Optimal control of the penicillin g. fed-batch fermentation: an analysis of the model of heijnen *et al.* *Optimal Control Applications and Methods*, v. 15, n. 1, p. 13–34, 1994.
- VIEGAS, M. C. *Otimização de sistema de fermentação alcoólica contínua utilizando reatores tipo torre e leveduras com características floculantes*. Tese (Doutorado) — Universidade Estadual de Campinas, Campinas, SP, 2003.

- ÁVILA, S. L. *Otimização paramétrica com computação evolutiva*. 1. ed. Florianópolis, SC: Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, 2020.
- VILELA, P. R. C. *Modelagem, simulação e otimização dinâmica aplicada a um processo de fermentação alcoólica em batelada alimentada*. Dissertação (Mestrado) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, SP, 2015.
- VILLA, C. R. de *et al.* Comparative study of inoculation strategies of *torulaspora delbrueckii* and *saccharomyces cerevisiae* on the performance of alcoholic and malolactic fermentations in an optimized synthetic grape must. *International Journal of Food Microbiology*, v. 404, n. 1, p. 1–11, 2023.
- VILLEN, R. A. *Biotecnologia: histórico e tendências*. 1. ed. São Caetano do Sul, SP: Centro Universitário do Instituto Mauá de Tecnologia, Escola de Engenharia Mauá, 2009.
- WANG, J.; TEPFENHART, W. *Formal methods in computer science*. 1. ed. Boca Raton, FL: CRC Press, 2019.
- WECK, O. L. Multiobjective optimization: history and promise. *The Third China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems*, v. 1, n. 1, p. 403–408, 2004.
- YABARRENA, J. M. S. C. *Modelagem do processo de fermentação etanoica com interferência de bactérias heterofermentativas e homofermentativas*. Dissertação (Mestrado) — Escola de Engenharia de São Carlos, Universidade de São Carlos, São Carlos, SP, 2012.
- YANG, S. *et al.* A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 17, n. 5, p. 721–736, 2013.
- YUAN, Y. *et al.* A new dominance relationbased evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 20, n. 1, p. 16–37, 2016.
- ZARPELON, F.; ANDRIETTA, S. R. Fermentação contínua para produção de álcool. *Stab: Açúcar, Alcool e Subprodutos*, v. 1, n. 1, p. 23–28, 1992.
- ZITZLER, E.; DEB, K.; THIELE, L. Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation*, v. 8, n. 2, p. 173–195, 2000.
- ZITZLER, E.; THIELE, L. *An evolutionary algorithm for multiobjective optimization: the strength pareto approach*. Zurich, CH, 1998.
- ZUBEN, F. J. V. Computação evolutiva. *Anais da I Jornada de Estudos em Computação de Piracicaba e Região (JECOMP)*, v. 1, n. 1, p. 25–45, 2000.

APÊNDICE A – Métodos Disponíveis na Biblioteca de *Software* desenvolvida

As pedras que encontrares em seu caminho,
serão aquelas com as quais construirás o teu
castelo.

Nemo Nox

Neste anexo são apresentados os métodos de otimização, interpolação e para equações diferenciais ordinárias, que estão disponíveis para uso na biblioteca de código desenvolvida. Cada um deste será enunciado a seguir.

A.1 Métodos Numéricos para Equações Diferenciais Ordinárias

A lista abaixo enumera os métodos numéricos para equações diferenciais ordinárias disponíveis.

- Método de Runge-Kutta de 2^a Ordem (RK-2);
- Método de Runge-Kutta de 4^a Ordem (RK-4);
- Método de Runge-Kutta de 8^a Ordem (RK-8);
- Método Preditor-Corretor Não-Rígido de Adams (ADAMS);
- Backward Differentiation Formula (BDF).

A.2 Métodos de Interpolação

A lista abaixo enumera os métodos de interpolação disponíveis.

- Interpolação Linear;
- Interpolação de Krogh (KROGH);
- Interpolação de Hermite Cúbica por Partes (PCHIP);
- Interpolação de Akima (AKIMA);
- Interpolação baricêntrica;
- *Spline* Linear;

- *Spline* Quadrática;
- *Spline* Cúbica.

A.3 Métodos de Otimização

Nesta seção, serão enunciados os métodos de otimização mono-objetivo e multiobjetivo disponíveis na biblioteca construída.

A.3.1 Métodos de Otimização Mono-Objetivo

Abaixo estão mostrados os métodos de otimização mono-objetivo disponíveis para uso.

- Método de Nelder-Mead (NELDER-MEAD);
- Busca de Padrões (PS);
- Algoritmo Genético (AG);
- Evolução Diferencial (ED);
- Estratégia Evolutiva (ES);
- Covariance Matrix Adaptation Evolution Strategy (CMA-ES);
- Stochastic Ranking Evolutionary Strategy (SRES);
- Improved Stochastic Ranking Evolutionary Strategy (ISRES).

A.3.2 Métodos de Otimização Multiobjetivo

Abaixo estão mostrados os métodos de otimização multiobjetivo disponíveis para uso.

- Non-dominated Sorting Genetic Algorithm II (NSGA-II);
- Reference Point Based Non-dominated Sorting Genetic Algorithm II (R-NSGA-II);
- Non-dominated Sorting Genetic Algorithm III (NSGA-III);
- Unified Non-dominated Sorting Genetic Algorithm III (U-NSGA-III);

- Reference Point Based Non-dominated Sorting Genetic Algorithm III (R-NSGA-III);
- Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D);
- Adaptive Geometry Estimation Based Multiobjective Evolutionary Algorithm (AGE-MOEA);
- Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization (C-TAEA);
- Multiobjective Selection Based on Dominated Hypervolume (SMS-EMOA);
- Reference Vector Guided Evolutionary Algorithm (RVEA);

APÊNDICE B – Linguagem e Bibliotecas Usadas pela Ferramenta Computacional Desenvolvida

Tudo o que um sonho precisa para ser realizado é de alguém que acredite que possa torná-lo real.

Roberto Shinyashiki

Neste anexo, serão listadas as bibliotecas de *software* usadas no sistema que implementa o CODEMSOA, que consiste em uma biblioteca de código, bem como as informações acerca da linguagem de programação empregada.

B.1 Linguagem de Programação

A biblioteca de código foi desenvolvida na versão 3.10.11 da linguagem de programação Python. Dessa forma, não há como garantir que a mesma funcione corretamente em versões anteriores, uma vez que não foram realizados testes de compatibilidade para atestar quais as versões compatíveis.

B.2 Bibliotecas Utilizadas

A biblioteca de código desenvolvida neste trabalho faz uso de uma série de outras bibliotecas, que devem estar previamente instaladas, a fim de que esta opere corretamente. O Quadro 2 apresenta as bibliotecas utilizadas, suas respectivas versões e uma breve descrição de cada uma.

Neste contexto, também não é possível garantir o funcionamento do *software* desenvolvido com versões diferentes das bibliotecas apresentadas, pois não foram executados testes para aferir a compatibilidade deste com outras versões destas bibliotecas.

Quadro 2 - Bibliotecas de códigos usadas.

Biblioteca	Versão	Descrição
matplotlib	3.6.2	Biblioteca que fornece recursos para plotagem de gráficos e visualizações de dados.
numpy	1.23.5	Biblioteca que fornece tipos e rotinas para se trabalhar com arranjos multidimensionais e com grande número de posições.
scipy	1.9.3	Biblioteca que fornece métodos para otimização, integração, interpolação, problemas de valor próprio, equações algébricas, equações diferenciais e estatísticas.
autograd	1.5	Biblioteca que fornece rotinas para diferenciação algorítmica (DA) e suporte a otimização.
pymoo	0.6.0.1	Biblioteca que fornece métodos de otimização e recursos para visualização de dados.
cma	3.2.2	Biblioteca que fornece uma implementação para o método de otimização CMAES.
numba	0.56.4	Biblioteca que traduz funções do Python para código de máquina otimizado, em tempo de execução, reduzindo o tempo de conclusão dos algoritmos.

Fonte: O autor, 2023.

APÊNDICE C – Exemplo de Chamada ao CODEMSOA na Biblioteca de Código Desenvolvida

A felicidade pode ser encontrada inclusive nos momentos mais escuros; só é preciso se lembrar de acender a luz.

Rowling

Neste anexo, será apresentado um exemplo simples de chamada ao CODEMSOA, a nível de código, na implementação dada pela biblioteca de *software* proposta por esta tese.

A Figura 50 apresenta uma chamada ao CODEMSOA usando a abordagem funcional fornecida pela biblioteca de *software* desenvolvida. Esta se dá pela invocação da operação estática `optimize` da classe `StandardFunctionalOptimizer`.

Figura 50 - Exemplo de chamada ao CODEMSOA na biblioteca de código desenvolvida.

```

1 solucao = StandardFunctionalOptimizer.optimize(
2     algorithmName           = 'NSGA-II',
3     objectives              = [ obj_1, obj_2 ],
4     variablesNumber         = 2,
5     inequalityConstraints    = [ ict_1, ict_2, ict_3 ],
6     equalityConstraints      = [ ect_1, ect_2, ect_3 ],
7     variablesLowerLimits    = [ var_1_low_lim, var_2_low_lim ],
8     variablesUpperLimits    = [ var_1_upr_lim, var_2_upr_lim ],
9     calls                   = [ odeProblemSolver ],
10    algorithmParameters     = { "pop_size": 1000,
11                               "n_offsprings": 10,
12                               "eliminate_duplicates": True
13                               },
14    algorithmConfigurations = { "termination": ('n_gen', 5) },
15    objectivesArguments     = ( obj_arg_1, obj_arg_2 ),
16    callsArguments          = ( call_arg_1, call_arg_2 )
);

```

Fonte: O autor, 2023.

No presente exemplo, empregou-se o CODEMSOA para resolver um problema de otimização de dois objetivos (`obj_1`, `obj_2`), que recebem os argumentos `obj_arg_1`, `obj_arg_2`, três restrições de desigualdade (`ict_1`, `ict_2`, `ict_3`), três restrições de igualdade (`ect_1`, `ect_2`, `ect_3`), duas variáveis, cujos limites inferiores e superiores são `var_1_low_lim`, `var_2_low_lim`, `var_1_upr_lim`, `var_2_upr_lim`, respectivamente.

O algoritmo de otimização escolhido foi o NSGA-II com uma população de 1000 indivíduos, com 10 descendentes por geração, não permitindo repetições, tendo por condição de parada, alcançar a 5ª geração.

O problema de equações diferenciais ordinárias é introduzido ao problema de otimização através da implementação de solução `odeProblemSolver`, que recebe os argumentos `call_arg_1`, `call_arg_2`.

Cabe ressaltar que este é apenas um exemplo ilustrativo bastante simplista para que o leitor tenha uma impressão inicial do uso operacional da biblioteca de código proposta.

Existem outras partes requeridas para esse exemplo, diversas configurações disponíveis e outros elementos da biblioteca que não foram ilustrados aqui.

APÊNDICE D – Configurações do *Hardware* onde as Simulações foram Realizadas

O caminho mais certo para vencer é tentar mais uma vez.

Thomas Edison

Neste anexo, estão descritas, no Quadro 3, as configurações de hardware dos nós do Cluster do Laboratório de Computação de Alto Desempenho (LCAD) do Instituto Politécnico (IPRJ) da Universidade do Estado do Rio de Janeiro (UERJ), onde foram realizadas as simulações computacionais de otimização realizadas neste trabalho, que executaram o CODEMSOA.

Quadro 3 - Configurações dos computadores que executaram as simulações.

Informação	Valor
PROCESSADOR	
Modelo	Intel Xeon E5-2620 v3 2.40 GHz
CPUs	24
Cache L1	32 KB
Cache L1i	32 KB
Cache L2	256 KB
Cache L3	15360 KB
RAM	
Modelo	DIMM DDR3
Capacidade	50 GB

Fonte: O autor, 2023.