

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Modelo de otimização biogeográfico
hibridizado com algoritmos
meméticos para a conceção ótima de
sistemas mecânicos

Arcílio Carlos Ferreira Peixoto

up202209749@up.pt



Mestrado em Engenharia Mecânica

Orientador: Carlos Conceição António

Julho, 2024

**Modelo de otimização biogeográfico hibridizado com
algoritmos meméticos para a conceção ótima de
sistemas mecânicos**

Arcílio Carlos Ferreira Peixoto

up202209749@up.pt

Mestrado em Engenharia Mecânica

Julho, 2024

Resumo

A ciência da Biogeografia, foi descrita, através de equações matemáticas, em 1967, por Robert MacArthur e Edward Wilson. E em 2008, Dan Simon, apresentou um algoritmo designado por otimização baseada em biogeografia ou BBO, que utilizava alguns dos princípios e definições descritos no livro de MacArthur e Wilson. O método provou ser bastante promissor, tendo apresentado bons resultados no exemplo em que foi estudado. Mas, no final desse artigo, Simon mencionou vários pontos a melhorar. Desde então, vários outros estudos surgiram, tentando e sucedendo na resolução de alguns dos problemas mencionados. Este trabalho propõe-se a estudar o comportamento do método BBO, aplicando-o em alguns exemplos da área da matemática e da engenharia de sistemas mecânicos e a hibridizá-lo, com operadores de outros algoritmos evolucionários, de forma a conseguir ultrapassar algumas das limitações e problemas que surgiram no decorrer desta tese.

Inicialmente, foi testado o método base, com quatro modelos de migração (linear, trapezoidal, quadrático e sinusoidal), dois operadores de mutação (normal e taxonómica) e o conceito de *blending* na migração, que ao contrário do processo normal utiliza informação das duas soluções escolhidas no processo de migração. O programa demonstrou uma enorme facilidade nas gerações iniciais, com a sua excelente pesquisa global, mas apresentou bastantes dificuldades nas gerações intermédias e finais, onde, na maioria dos exemplos, não conseguiu chegar a nenhuma solução ótima. De início pensou-se que o algoritmo poderia estar a ter dificuldades, porque lhe faltava diversidade populacional. Assim, foi implementado um ciclo iterativo de mutação, onde se corria o operador de mutação por vários ciclos, sendo que a nova solução só entrava na população se fosse melhor que a solução atual, e um processo de inserção de novas soluções, que no final de cada geração gerava novas soluções para substituir as piores soluções obtidas. Estas estratégias conseguiram melhorar os resultados obtidos, mas o método continuou longe das melhores soluções.

Em termos de migração, os modelos que mais se destacaram foram os modelos sinusoidal e linear, sendo seguidos, proximamente, pelo modelo quadrático e, finalmente, o modelo trapezoidal que revelou bons resultados, mas apresentou ser o mais lento, quando comparado aos outros modelos, e apresentou uma má sinergia com os outros operadores implementados. Ambas as mutações conseguiram apresentar bons resultados, sendo que, a mutação taxonómica prioriza a precisão, com um custo na redução da velocidade.

De seguida, foram implementados três operadores de crossover com codificação binária (Single-Point, Multi-Point e Uniform), de forma a melhorar a distribuição e partilha de boa informação genética, e duas estratégias de seleção. A primeira estratégia utilizava as melhores soluções anteriores como um dos pais, sendo o outro pai selecionado, na população corrente, através das taxas de migração. A segunda estratégia dividia a população atual em dois polos e selecionava um pai por polo, mediante um processo baseado numa roleta. Esta inovação gerou melhores resultados, principalmente com o uso do *blending* e da mutação normal, mas ainda continuavam aquém do esperado.

Mais tarde, foi adicionado um otimizador local Hooke-Jeeves. Este foi testado com e sem a influência dos operadores de crossover. A inclusão do otimizador local trouxe uma melhoria significativa, tendo superado as outras adições, como o crossover e o ciclo de mutação, em termos de velocidade e precisão, mesmo sem o crossover. Mesmo assim, as melhores soluções foram geradas com a utilização de todas as inovações. De forma geral, o objetivo definido no início desta dissertação, que era a implementação e hibridização do método de otimização BBO, foi bem-sucedido, tendo conseguido chegar e até superar, em alguns casos, os valores ótimos de referência, obtidos na literatura.

Abstract

The science of biogeography was described, through mathematical equations, in 1967, by Robert MacArthur and Edward Wilson. And in 2008, Dan Simon presented an algorithm called biogeography-based optimization, or BBO, which used some of the principles and definitions described in MacArthur and Wilson's book. The method proved to be very promising, having shown good results in the example in which it was studied. But, at the end of that article, Simon mentioned several points for improvement. Since then, several other studies have emerged, trying and succeeding in solving some of the problems mentioned. This work proposes to study the behavior of the BBO method, applying it in some examples of mathematics and mechanical systems engineering, and to hybridize it, with operators of other evolutionary algorithms, in order to overcome some of the limitations and problems that arose during this thesis.

Initially, the basic method was tested, with four migration models (linear, trapezoidal, quadratic and sinusoidal), two mutation operators (normal and taxonomic) and the concept of *blending* in migration, which unlike the normal process uses information from the two solutions chosen in the migration process. The programme showed great ease in the early generations, with its excellent overall research, but it presented a great deal of difficulties in the intermediate and final generations, where, in most examples, it was unable to arrive at any optimal solution. At first, it was thought that the algorithm might be having a hard time, because it lacked population diversity. Thus, an iterative mutation cycle was implemented, where the mutation operator was run through several cycles, with the new solution only entering the population if it was better than the current solution, and a process of inserting new solutions, which at the end of each generation generated new solutions to replace the worst solutions obtained. These strategies managed to improve the results obtained, but the method remained far from the best solutions.

In terms of migration, the models that stood out the most were the sinusoidal and linear models, closely followed by the quadratic model and, finally, the trapezoidal model, which showed good results, but proved to be the slowest when compared to the other models, and presented a poor synergy with the other operators implemented. Both mutations were able to show good results, and the taxonomic mutation prioritizes accuracy, at a cost in reducing speed.

Then, three crossover operators with binary coding (Single-Point, Multi-Point and Uniform) were implemented, in order to improve the distribution and sharing of good genetic information, and two selection strategies. The first strategy used the previous best solutions as one parent, with the other parent selected from the current population through migration rates. The second strategy divided the current population into two poles and selected one parent per pole, through a process based on a roulette wheel. This innovation generated better results, mainly with the use of *blending* and the normal mutation, but they still fell short of expectations.

Later, a local Hooke-Jeeves optimizer was added, this was tested with and without the influence of crossover operators. The inclusion of the local optimizer brought a significant improvement, having surpassed the other additions, such as the crossover and mutation cycle, in terms of speed and accuracy, even without the crossover. Even so, the best solutions were generated with the use of all innovations. In general, the objective defined at the beginning of this dissertation, which was the implementation and hybridization of the BBO optimization method, was successful, having managed to reach and even surpass, in some cases, the optimal reference values obtained in the literature.

Agradecimentos

Primeiramente, gostaria de agradecer ao meu orientador, o Prof. Doutor Carlos Conceição António, que me guiou e assistiu por todo o percurso desta dissertação, quer seja na implementação do código, quer na formatação do documento. Agradeço imenso por todo o tempo que perdeu comigo, mesmo quando estava ocupado com outras responsabilidades.

Também gostaria de agradecer, aos meus amigos, colegas de curso e docentes da academia, por estes últimos dois anos de mestrado. Gostaria de realçar dois dos meus colegas: Rodrigo Carvalho e Vicente Silva, por toda a paciência e assistência neste último semestre. Eu bem sei, que muito vos chateei.

Finalmente, e não menos importante, gostaria de agradecer aos meus pais por tudo, que fizeram por mim e por todos os sacrifícios que fizeram, nos meus vinte dois anos de vida, todos sabemos que não foram poucos.

"The time you enjoy wasting is not wasted time."— Bertrand Russell

Conteúdo

Resumo	i
Abstract	iii
Agradecimentos	v
Lista de Figuras	ix
Lista de Tabelas	xi
1 Introdução	1
1.1 Objetivos e Enquadramento	1
1.2 Otimização	2
1.2.1 Definição do problema de otimização	3
1.2.2 Abordagem ao problema de otimização — O Conceito das Três Colunas	3
1.3 Estrutura	4
2 Estado da Arte	7
2.1 Otimização baseada em Biogeografia	7
2.1.1 Hibridizações do BBO	9
2.1.2 Modificações do BBO	10
3 Algoritmos Evolucionários	13
3.1 Algoritmos Genéticos	13
3.1.1 Definição de mérito ou aptidão	14
3.1.2 Aplicação de Restrições	17
3.1.3 Operador Crossover	19
3.1.4 Mutação	24
3.1.5 Outros parâmetros	26
3.2 Algoritmos Meméticos	26
3.2.1 Processos de Aprendizagem	27
3.2.2 Aspectos Essenciais dos Algoritmos Meméticos	28
3.3 Otimização baseada em Biogeografia	31
3.3.1 Modelos Matemáticos	31
3.3.2 Migração	34
3.3.3 Mutação	37
3.3.4 Definições	39
3.3.5 Adaptação do BBO para otimização com restrições	41
3.3.6 Adaptação para otimização multiobjectivo	42

4	Metodologia	45
4.1	Implementação do BBO	45
4.2	Implementação da Hibridização	47
4.2.1	Aplicação de operadores de crossover	48
4.2.2	Aplicação do otimizador local (Hook-Jeeves)	48
5	Resultados e Discussão	53
5.1	Exemplo de Teste: Função Quadrática	53
5.1.1	Resultados	53
5.2	Exemplo de Teste Polinomial	54
5.2.1	Resultados	54
5.3	Dimensionamento de uma Viga Soldada	56
5.3.1	Resultados	58
5.4	Dimensionamento de um Reforço Soldado Gusset	70
5.4.1	Resultados	72
5.5	Otimização de um sistema de molas e pesos	84
5.5.1	Resultados	86
6	Conclusões e Desenvolvimentos Futuros	101
A	Gráficos da média do mérito da elite	105

Lista de Figuras

3.2	Pseudo algoritmo de aplicação do vetor máscara [6].	21
3.3	Mutação de troca genética do tipo " <i>flip-bit</i> ".	25
3.4	Representação gráfica do modelo de migração linear ([5]).	31
3.5	Representação gráfica do modelo de migração quadrático ([59]).	36
3.6	Representação gráfica do modelo de migração sinusoidal ([59]).	36
3.7	Representação gráfica do modelo de migração trapezoidal ([10]).	37
4.1	Diagrama de fluxo do BBO implementado.	46
4.2	Diagrama de fluxo completo do método hibridizado.	50
5.1	Geometria do problema da viga soldada.	56
5.2	Evolução da média do mérito da elite em 5000 gerações, sem <i>blending</i>	59
5.3	Evolução da média do mérito da elite em 5000 gerações, sem <i>blending</i> e com mutação taxonómica.	60
5.4	Evolução da média do mérito da elite em 5000 gerações, com <i>blending</i>	60
5.5	Evolução da média do mérito da elite em 5000 gerações, com <i>blending</i> e mutação taxonómica.	61
5.6	Evolução da média do mérito da elite em 5000 gerações com a utilização do otimizador local, sem <i>blending</i>	66
5.7	Evolução da média do mérito da elite em 5000 gerações com a utilização do otimizador local, com mutação taxonómica.	66
5.8	Evolução da média do mérito da elite em 5000 gerações com a utilização do otimizador local, com <i>blending</i>	67
5.9	Evolução da média do mérito da elite em 5000 gerações com a utilização do otimizador local, com <i>blending</i> e mutação taxonómica.	67
5.10	Dimensionamento de um reforço soldado Gusset.	71
5.11	Evolução da média do mérito da elite ao longo das gerações, para o algoritmo sem <i>blending</i> e com mutação normal.	74
5.12	Evolução da média do mérito da elite ao longo das gerações, para o algoritmo com mutação taxonómica e sem <i>blending</i>	74
5.13	Evolução da média do mérito da elite ao longo das gerações, para o algoritmo com <i>blending</i> e com mutação normal.	75
5.14	Evolução da média do mérito da elite ao longo das gerações, para o algoritmo com mutação taxonómica e <i>blending</i>	75
5.15	Evolução da média do mérito da elite ao longo das gerações, para o algoritmo com otimizador local, sem <i>blending</i> e com mutação normal.	79
5.16	Evolução da média do mérito da elite ao longo das gerações, para o algoritmo com otimizador local e mutação taxonómica e sem <i>blending</i>	79

5.17	Evolução da média do mérito da elite ao longo das gerações, para o algoritmo com otimizador local, mutação normal e <i>blending</i>	80
5.18	Evolução da média do mérito da elite ao longo das gerações, para o algoritmo com otimizador local, mutação taxonómica e <i>blending</i>	80
5.19	Exemplo da evolução da média do mérito da elite, para um teste com codificador binário e mutação taxonómica.	84
5.20	Sistema de molas na posição inicial e na posição deformada.	85
5.21	Evolução da média do mérito da elite ao longo das gerações, para o exemplo das molas e pesos, usando a mutação normal sem <i>blending</i>	89
5.22	Evolução da média do mérito da elite ao longo das gerações, para o exemplo das molas e pesos, usando a mutação taxonómica sem <i>blending</i>	89
5.23	Evolução da média do mérito da elite ao longo das gerações, para o exemplo das molas e pesos, usando <i>blending</i> e a mutação normal.	90
5.24	Evolução da média do mérito da elite ao longo das gerações, para o exemplo das molas e pesos, usando <i>blending</i> e a mutação taxonómica.	90
5.25	Evolução da média do mérito da elite pelas gerações, para o exemplo das molas e pesos, usando o otimizador local, com a mutação normal e sem <i>blending</i>	97
5.26	Evolução da média do mérito da elite pelas gerações, para o exemplo das molas e pesos, usando o otimizador local e <i>blending</i> com a mutação normal.	97
5.27	Evolução da média do mérito da elite pelas gerações, para o exemplo das molas e pesos, usando o otimizador local, mutação taxonómica e <i>blending</i>	98

Lista de Tabelas

5.1	Resultados para o primeiro exemplo.	54
5.2	Resultados do estudo dos modelos de migração para o segundo exemplo. . .	55
5.3	Resultados obtidos através de codificação binária de soluções para o segundo exemplo.	55
5.4	Melhores soluções para o teste dos modelos de migração e mutação, sem o crossover e o otimizador local, para o exemplo da viga soldada.	59
5.5	Resultados obtidos para os operadores de crossover utilizando a primeira estratégia (secção 4.2.1) de seleção e codificação, para o exemplo da viga soldada.	63
5.6	Resultados obtidos para os operadores de crossover utilizando a segunda estratégia (secção 4.2.1) de seleção e codificação, para o exemplo da viga soldada.	64
5.7	Resultados do teste com otimizador local, sem crossover, para o exemplo da viga soldada.	65
5.8	Resultados do teste com otimizador local, com crossover utilizando a primeira estratégia (secção 4.2.1), para o exemplo da viga soldada.	68
5.9	Resultados do teste com otimizador local, com crossover utilizando a segunda estratégia (secção 4.2.1), para o exemplo da viga soldada.	69
5.10	Melhores soluções, para o exemplo da viga Gusset.	73
5.11	Melhores soluções, para o exemplo da viga Gusset, quando é utilizado o crossover, seguindo a primeira estratégia (secção 4.2.1).	76
5.12	Melhores soluções, para o exemplo da viga Gusset, quando é utilizado o crossover, seguindo a segunda estratégia (secção 4.2.1).	77
5.13	Melhores soluções, para o exemplo da viga Gusset, quando é utilizado o otimizador local.	78
5.14	Melhores soluções, para o exemplo da viga Gusset, para o algoritmo com otimizador local e crossover, seguindo a primeira estratégia (secção 4.2.1). . .	81
5.15	Melhores soluções, para o exemplo da viga Gusset, para o algoritmo com operador de otimização local e crossover, seguindo a segunda estratégia (secção 4.2.1).	82
5.16	Comparação da velocidade entre as duas estratégias, para o exemplo da viga de Gusset.	83
5.17	Melhores soluções usando o modelo linear, para o exemplo das molas e pesos.	87
5.18	Melhores soluções usando o modelo trapezoidal, para o exemplo das molas e pesos.	87
5.19	Melhores soluções usando o modelo quadrático, para o exemplo das molas e pesos.	88
5.20	Melhores soluções usando o modelo sinusoidal, para o exemplo das molas e pesos.	88

5.21	Melhores soluções usando o modelo linear com operadores de crossover, com a primeira estratégia (secção 4.2.1), para o exemplo das molas e pesos. . . .	91
5.22	Melhores soluções usando o modelo trapezoidal com operadores de crossover, com a primeira estratégia (secção 4.2.1), para o exemplo das molas e pesos. .	91
5.23	Melhores soluções usando o modelo quadrático com operadores de crossover, com a primeira estratégia (secção 4.2.1), para o exemplo das molas e pesos. .	92
5.24	Melhores soluções usando o modelo sinusoidal com operadores de crossover, com a primeira estratégia (secção 4.2.1), para o exemplo das molas e pesos. .	92
5.25	Melhores soluções usando o modelo linear com operadores de crossover, com a segunda estratégia (secção 4.2.1), para o exemplo das molas e pesos. . . .	93
5.26	Melhores soluções usando o modelo trapezoidal com operadores de crossover, com a segunda estratégia (secção 4.2.1), para o exemplo das molas e pesos. .	94
5.27	Melhores soluções usando o modelo quadrático com operadores de crossover, com a segunda estratégia (secção 4.2.1), para o exemplo das molas e pesos. .	94
5.28	Melhores soluções usando o modelo sinusoidal com operadores de crossover, com a segunda estratégia (secção 4.2.1), para o exemplo das molas e pesos. .	95
5.29	Melhores soluções usando os modelos linear e trapezoidal com otimizador local, para o exemplo das molas e pesos.	96
5.30	Melhores soluções usando os modelos quadrático e sinusoidal com otimizador local, para o exemplo das molas e pesos.	96
5.31	Melhores soluções usando o otimizador local com operadores de crossover, com a primeira estratégia (secção 4.2.1), para o exemplo das molas e pesos. .	99
5.32	Melhores soluções usando o otimizador local com operadores de crossover, com a segunda estratégia (secção 4.2.1), para o exemplo das molas e pesos. .	100

Lista de Algoritmos

1	Pseudocódigo de um operador de procura local [6].	29
2	Pseudocódigo para a migração [59].	35
3	Algoritmo para a mutação normal do BBO.	38
4	Pseudocódigo para a mutação diferencial, da mutação taxonómica [61].	39
5	Pseudocódigo para a mutação aleatória, da mutação taxonómica [61].	39
6	Pseudocódigo do BBO base.	47
7	Pseudocódigo do método de pesquisa local de Hooke-Jeeves	49
8	Pseudocódigo do BBO hibridizado com pesquisa local.	51

Capítulo 1

Introdução

Esta tese foi desenvolvida no decorrer da unidade curricular Dissertação, da Faculdade de Engenharia da Universidade do Porto, do Mestrado em Engenharia Mecânica — Especialização em Estruturas Aeronáuticas e de Veículos.

Este capítulo apresentará os objetivos propostos e executados no decorrer desta dissertação. Também será exposta uma pequena secção referente ao enquadramento deste trabalho e a área dos algoritmos evolucionários, havendo um maior foco nos algoritmos mais explorados neste trabalho, como o algoritmo genético e o algoritmo memético. Realça-se também a motivação que levou à criação e desenvolvimento desta dissertação, sendo também apresentada uma pequena introdução à área de otimização. Finalmente, irá ser apresentada a estrutura desta dissertação, que contará com a descrição de cada capítulo e secção.

1.1 Objetivos e Enquadramento

Os problemas em engenharia têm-se tornado cada vez mais complexos, sendo a sua resolução analítica praticamente impossível de decifrar. Assim, métodos de solução numéricos, acabaram por receber um maior foco. A otimização, enquadrando-se nesta área, tornou-se uma das disciplinas mais fulcrais para o quotidiano de um engenheiro moderno. Até porque não existe uma área da engenharia onde a otimização não esteja envolvida de alguma forma. Por isso, esta área tem sofrido um enorme desenvolvimento nos tempos recentes. Um dos eventos mais marcantes foi a introdução do algoritmo genético por John Holland [1], o primeiro algoritmo evolucionário com operadores estruturados, um evento que mudou o paradigma da otimização. Desde então a comunidade científica conseguiu melhorar o algoritmo genético, surgindo uma grande diversidade de propostas de operadores genéticos baseados em esquemas e heurísticas diferentes, para a resolução de problemas de otimização. Sendo o algoritmo genético um algoritmo que retira conhecimentos de outras áreas, como a biologia e a genética, e aplica-os, matematicamente, sob a forma de um algoritmo, nos vários campos da engenharia. Também existem outros métodos que se inspiraram em áreas diversas, como é o caso dos algoritmos meméticos, que se baseiam na biologia

(evolução genética) [2] e na sociologia (evolução cultural) [3]-[4], e do método que irá ser estudado nesta dissertação, otimização baseada em biogeografia (BBO).

Biogeografia é o estudo da distribuição das espécies e ecossistemas no espaço geográfico através da idade geológica. As pesquisas biogeográficas combinam informações e ideias de vários campos, desde as limitações fisiológicas e ecológicas sobre a dispersão dos organismos, aos fenômenos geológicos e climatológicos que operam na escala global. em períodos evolutivos.

A otimização baseada em biogeografia (BBO) é um algoritmo evolucionário, introduzido em 2008 por Dan Simon [5], que otimiza uma função ao melhorar as soluções candidatas iterativamente e estocasticamente de acordo com algum medidor de qualidade, ou função de mérito. Tipicamente, este método é utilizado para otimizar problemas multivariados e, como não faz uso do gradiente da função, também pode ser usado em funções descontínuas. O BBO otimiza um problema ao manter uma população de soluções candidatas, criando soluções através do processo de migração.

O principal objetivo deste trabalho é o desenvolvimento e implementação do BBO, seguindo a literatura já disponível, e, mais tarde, hibridizar o método com outros conceitos de otimização, como o otimizador local e outras propostas usadas por outros algoritmos evolucionários. O BBO é aplicado a vários exemplos da área da otimização e da engenharia, para conseguir estudar o comportamento do algoritmo e o efeito das modificações introduzidas, no processo de hibridização.

1.2 Otimização

António [6] define otimização como a criação racional de um projeto, que será o melhor, dentro de todos os projetos, de acordo com um ou mais objetivos predefinidos. Consultando a Wikipédia [7], otimização é definida como a seleção de um melhor elemento, com relação a algum critério, de algum conjunto de alternativas disponíveis. A otimização é um ramo da Matemática, usado numa variada gama de aplicações industriais, desde engenharia estrutural a engenharia química. Por isso, esta área foi sujeita a várias investigações, teóricas e práticas, ao longo dos anos. Mas, a otimização é um pouco mais que isso, otimização é algo que está presente no nosso cotidiano, otimização é algo extremamente humano. Algo que se vê, na humanidade, por toda a sua história. Reduzir a otimização a uma área matemática, é algo irônico, a otimização está tão “otimizada” na mente do Homem, que agora consegue ser reduzida apenas a uma área quando, na verdade, é muito mais que isso. A otimização é conseguir construir um mundo com o pouco que se tem e ainda sobrar para oferecer aos outros. E, assim, talvez seja possível perceber o porquê desta área sempre ter um grande destaque, não só na área da engenharia, mas nas várias disciplinas da vida humana.

1.2.1 Definição do problema de otimização

No caso mais simples, um problema de otimização consiste em maximizar ou minimizar uma função real escolhendo sistematicamente os valores de entrada de um conjunto permitido e calculando o valor da função. De maneira mais geral, a otimização inclui encontrar os melhores valores disponíveis de alguma função objetivo dado um domínio definido. O procedimento para a formulação de um problema de otimização pode ser definido em cinco passos.

1. Uma descrição geral do problema ou projeto;
2. Recolha de dados;
3. Definição das variáveis de projeto;
4. Definição do objetivo (critério de otimização);
5. Identificação de possíveis restrições;

O quarto passo é, normalmente, associado à criação de uma função objetivo, ou seja, uma expressão que reflete o problema em análise e que irá ser minimizado ou maximizado. Esta função pode ser o peso ou o custo de uma estrutura, por exemplo. Um bom modelo de projeto deverá ser um modelo que represente a realidade na forma mais simples e significativa.

De forma geral, um estudo de otimização é realizado após uma análise sofisticada. Os seus conceitos são introduzidos para converter um recurso de análise num recurso de projeto. Como foi dito antes, os problemas de otimização são problemas de minimização ou maximização, sujeitos a um conjunto de restrições. A formulação matemática de um problema de otimização pode ser definida como,

$$\min_{x \in \mathbb{R}^n} \{f(\mathbf{x}) : \mathbf{h}(\mathbf{x}) = 0, \mathbf{g}(\mathbf{x}) \leq 0\} \quad (1.1)$$

onde \mathbf{x} é o vetor de variáveis, $f(\mathbf{x})$ a função objetivo, \mathbb{R}^n o conjunto de valores reais, $\mathbf{h}(\mathbf{x})$ o vetor de restrições de igualdade e $\mathbf{g}(\mathbf{x})$ o vetor de restrições de desigualdade.

1.2.2 Abordagem ao problema de otimização — O Conceito das Três Colunas

Um problema de otimização pode ser tratado de acordo com o Conceito das Três Colunas (*Three-Columns-Concept*) integrado no processo do projeto de otimização. Este conceito define três entidades:

1. Modelo de Análise do Sistema
2. Algoritmo de otimização
3. Modelo de otimização

A primeira coluna: Modelo de Análise do Sistema

A análise do modelo do sistema avalia cada solução de projeto. Existem duas relações, direta e indireta, entre os valores das variáveis do problema e a configuração dos parâmetros do sistema. As mudanças produzidas no sistema modificarão os valores das variáveis de estado (tensões, deslocamentos, deformações) sendo atualizadas para a solução atual.

A segunda coluna: Algoritmo de otimização

Os algoritmos de otimização são baseados em processos ou heurísticas que começam a partir de uma solução inicial, ou de uma população de soluções, e produzem uma sequência de soluções melhoradas. Sendo este processo terminado após atingir algum critério de paragem, previamente definido.

Métodos de programação matemática, normalmente baseados em processos iterativos ou heur que utilizam gradientes para determinar a direção de pesquisa e tamanho do passo, são utilizados para a resolução de problemas de otimização com restrições. Para problemas específicos vários autores utilizam técnicas baseadas em critérios de otimalidade, que utilizam as condições de otimalidade de Karush-Kuhn-Tucker (KKT). Finalmente, um novo método chamado de Métodos de Pesquisa Evolucionária, surgiu no ramo da otimização de estruturas. Estes são baseados na lei da sobrevivência de espécies e na teoria de Darwin e não necessitam do uso de gradientes.

A terceira coluna: Modelo de otimização

O modelo de otimização forma uma ponte entre o modelo do sistema ou fenómeno físico a ser analisado e o algoritmo de otimização, sendo um pilar fundamental dentro do processo de otimização. As variáveis de estado são retiradas dos parâmetros da análise do modelo físico. O modelo de otimização descreve a relação matemática entre as variáveis de estado e as variáveis de projeto. O modelo de otimização também contém algumas definições e transformações de variáveis, denominadas de parametrização. Por exemplo, no caso dos algoritmos genéticos, que definem as suas variáveis em listas de *bits* e cuja informação codificada é transformada nas variáveis do fenótipo. O módulo de análise de sensibilidade calcula o gradiente das funções objetivo e de restrição, de acordo com o modelo ou outras variáveis de interesse.

1.3 Estrutura

Esta dissertação está dividida em 6 capítulos. O primeiro, a Introdução, apresenta uma breve introdução ao mundo da otimização, os objetivos deste projeto, o enquadramento, motivação e a estrutura da dissertação. O segundo, irá apresentar uma revisão do estado da arte sobre a otimização baseada em biogeografia, expondo alguns dos desenvolvimentos mais recentes com este método. O terceiro, apresenta as definições necessárias para

o entendimento de algoritmos evolucionários, assim, serão expostos algumas definições relativamente aos algoritmos genéticos e meméticos e um pouco da formulação matemática originada na criação destes métodos de otimização. O quarto capítulo, apresenta a metodologia formulada e implementada nesta dissertação, em conjunto com algumas definições para os operadores utilizados no decorrer deste trabalho. No quinto capítulo, apresentam-se os casos de estudo, resultados obtidos e uma discussão sobre o comportamento do algoritmo e dos novos operadores utilizados. Finalmente, o sexto apresenta uma breve conclusão sobre o trabalho realizado e alguns aspetos a melhorar e limitações, que surgiram no decorrer da dissertação. Existe também um apêndice que contará com alguns gráficos, que não foram apresentados na secção dos resultados, relativamente, aos testes com os operadores de crossover.

Capítulo 2

Estado da Arte

Este capítulo irá apresentar uma breve revisão do estado da arte, do método de otimização utilizado nesta dissertação. Inicialmente, será apresentada uma breve contextualização histórica, relativamente à biogeografia e ao desenvolvimento da criação deste método. Também serão apresentadas algumas definições, de forma, a poder contextualizar alguns dos conceitos que serão utilizados nas secções futuras. Serão, também, apresentadas secções relativamente, às modificações e hibridizações do BBO, de forma a tentar descrever o estado atual da otimização baseada em biogeografia.

2.1 Otimização baseada em Biogeografia

A ciência da Biogeografia, criada em meados do século XIX, era considerada descritiva e histórica, mas em 1967, Robert MacArthur e Edward Wilson, publicaram *The Theory of Island Biogeography* [8], onde estes tentaram descrever este fenómeno em equações matemáticas, tornando-se assim uma área de pesquisa com um grande foco. Estes modelos matemáticos tentam descrever como as espécies migram de uma ilha para a outra, como surgem novas espécies e como estas se tornam extintas. Neste caso, o termo ilha não deve ser tomado no seu sentido literal, sendo a definição de ilha, para este cenário, uma localização geograficamente isolada de outros habitats. Por isso, o termo habitat será usado para melhor descrever estas localizações.

Localizações com características aptas para a retenção de vida são marcadas por terem um alto índice de aptidão habitacional (HSI — *habitat suitability index*). Fatores, como a presença de água potável, que se relacionam com o HSI, são designados por variáveis dos índices de aptidão (SIV — *suitability index variables*). Em termos matemáticos, os SIV são os termos independentes e os HSI podem ser considerados os termos dependentes. Assim, habitats com HSI alto terão um número de espécies mais elevado e habitats com HSI baixo terão menos habitantes. Por isso, os habitantes de habitats com HSI elevado terão uma tendência maior a emigrar, não porque estes queiram sair do seu habitat, mas sim devido à acumulação de efeitos aleatórios em populações com muitos habitantes. Na vida real, a emigração de uma espécie, nem sempre resulta em extinção, mas, neste caso, a emigração

irá resultar na extinção da espécie nesse habitat. Habitats com um HSI baixo, terão uma taxa de imigração elevados, devido à sua baixa população e ambiente menos competitivo. Pelo contrário, habitats com HSI elevado irão ter uma taxa de emigração elevada e uma taxa de imigração baixa.

Boas soluções estão associadas a habitats com HSI mais elevado, enquanto habitats com baixo HSI estão associados a más soluções. Soluções com HSI alto são mais resistentes à mudança e, assim, soluções com HSI elevados tendem a partilhar características com as soluções com baixo HSI. Diversidade de indivíduos está relacionada ao HSI, por isso, quando um indivíduo chega a um habitat com HSI baixo, este tende a aumentar. A esta abordagem foi designada o termo otimização Baseada em Biogeografia (BBO).

O Algoritmo proposto por Dan Simon [5] pode ser chamado de BBO parcialmente baseado em imigração, visto que, o habitat imigrante é selecionado antes do habitat emigrante, sendo esta migração realizada, independentemente, para cada uma das variáveis do problema. Garg e Deep [9] usaram este algoritmo e compararam-no com outros métodos baseados em populações. Nesse teste, foi possível determinar que os métodos SGA (*Steady State Genetic Algorithm*) e o BBO tiveram os melhores resultados em 7 dos 14 parâmetros de referência. De acordo, com a melhor solução absoluta o SGA foi considerado o melhor, sendo o segundo lugar do BBO. E, em termos de tempo demorado, o BBO foi o quinto mais rápido. Os autores referem que, como não houve nenhum afinamento dos parâmetros do algoritmo e como não foram realizados nenhuns testes estatísticos, não era possível chegar a uma conclusão definitiva sobre qual algoritmo é considerado o “melhor”. Neste artigo, também é referido que o método coloca demasiado ênfase no cálculo das probabilidades usadas na taxa de mutação, tornando o operador desnecessariamente complicado e computacionalmente pesado.

No artigo inicial escrito por Dan Simon ([5]), este apresenta, no final, uma lista com várias direções para avançar o desenvolvimento deste algoritmo. A lista dada é a seguinte:

1. Investigar as propriedades de A' , apresentada na secção 3.3.1, mais especificamente os seus valores próprios, que poderão afetar a estabilidade, convergência e equilíbrio do algoritmo.
2. Aplicar o método a cenários com mérito dinâmico.
3. Explorar a ideia de espécies partilharem apenas entre outras soluções semelhantes, ou seja, a partilha de informação entre soluções cujos valores do mérito estejam próximos, tal como, o "*niching*" dos Algoritmos Genéticos.
4. Estender o algoritmo para o uso de variáveis contínuas, não só discretas.
5. Encontrar mecanismos de forma a melhorar a velocidade do BBO.
6. Encontrar um mecanismo que garanta que todas as soluções geradas são viáveis.

Várias hibridizações e modificações, nos operadores de migração e mutação, cujas definições estão expostas no capítulo 3.3, do BBO foram propostas, por diferentes investigadores, de forma a eliminar as desvantagens listadas previamente.

2.1.1 Hibridizações do BBO

A hibridização de métodos baseados em populações, tem sido algo comum nos tempos presentes. O BBO pode ser combinado com um operador de procura local, de forma a criar um algoritmo memético, ou poderá ser combinado com outros tipos de algoritmos. O BBO foi hibridizado com vários outros algoritmos evolucionários, tal como a otimização por enxame de partículas (PSO), algoritmos com raciocínio baseado em casos (CBR) e Algoritmos Simplex.

Algoritmos como o BBO, foram descritos com uma procura global incrível, mas não têm boa eficiência em procuras locais. Pelo contrário, algoritmos de pesquisa local têm uma pesquisa global medíocre e uma pesquisa local notável. Deste modo, a hibridização entre estes dois algoritmos é algo bastante promissor. Ma [10] apresentou uma versão linear do BBO (LBBO), onde o autor apresentou um BBO com pesquisa local e uma reinicialização periódica do algoritmo. Foram consideradas 45 funções de referência, e os resultados provaram-se positivos, quando comparados a algoritmos evolucionários recentes.

Albrase et al.[11] propuseram um BBO híbrido com programação linear para a coordenação de “*directional over-current relays*”, que é um problema de otimização altamente restrungido. Os resultados experimentais demonstraram uma superioridade sobre outros métodos de otimização. Um outro algoritmo híbrido chamado Metroplis BBO [12], que combina o BBO com recozimento simulado (SA), que contraria a tendência do BBO de encalhar em mínimos locais, causando convergência permanente. Este método foi testado em 36 funções de referência, onde este algoritmo demonstrou a sua convergência superior, em relação aos outros algoritmos evolucionários. Fen et al. [13], também tentaram melhorar a procura local do BBO, utilizando a informação de várias soluções para melhorar a exploração. Este algoritmo foi testado em 27 funções de referência e obteve resultados comparáveis a outras versões do BBO e a algoritmos de evolução diferencial (DE). Os vários testes realizados, em diferentes funções de referência, demonstram a eficiência deste método.

Na hibridização com outros algoritmos baseados em populações, o método DE é o mais popular e vários autores utilizaram esta hibridização, chamada BBO/DE, para resolver inúmeros problemas de otimização. Gong et al. [14], foram os primeiros a utilizar este método, para estudar 23 funções de referência. Neste artigo, os autores descobriram que o método mostrava resultados semelhantes ou melhores quando comparado a algoritmos DE de última geração. Esta abordagem também foi utilizada para problemas de EELD (emissões económicas de despacho de carga) e provou-se bastante útil para a resolução de casos práticos ([15]-[16]). Boussard et al. [17] utilizaram este método para a resolução de um problema relacionado a distribuição de energia numa rede de sensores sem fio, os

resultados obtidos demonstraram que este método supera a utilização do BBO e DE. Estes autores, também propuseram uma modificação deste algoritmo, introduzindo um sistema de atualização de duas fases, de forma a preservar soluções mais aptas para as gerações futuras.

A seguinte lista demonstra alguns outros algoritmos baseados em população, que foram hibridizados com o BBO:

- Raciocínio baseado em casos (CBR - *Case Based Reasoning*) [18]
- Otimização por enxame de partículas (PSO) [19]
- Algoritmos Simplex [20]
- Algoritmos de colmeia artificiais (*Artificial Bee Colony*) [21]
- Pesquisa harmónica (*Harmony Search*) [22]

Jayaraman e Ravi [23], apresentaram um método híbrido entre redes neurais e BBO para a previsão de energia elétrica setorial. Os autores treinaram a rede neural com o BBO, de forma a obter, os melhores parâmetros de procura. Eles demonstraram que o seu método conseguia prever energia elétrica setorial a longo prazo. Similarmente, Krishnasamy e Najundappan [24] apresentaram um novo método, WPNNBBO, que hibridizava uma rede neural probabilística com o BBO, para um despacho económico aplicado a unidades eólicas integradas com vários tipos de combustíveis. Outro método integrava um sistema de informação imprecisa (*fuzzy*) paralelo (PFS) e um BBO não homogéneo (NHBBO) para a seleção de tecnologias de acesso à rádio (RAT), em redes sem fios. Onde o PFS era utilizado para calcular a probabilidade de seleção do RAT, que por sua vez era utilizada como os dados de entrada do NHBBO. Os resultados demonstraram que este método era mais eficiente e robusto que os outros métodos utilizados nesta área.

2.1.2 Modificações do BBO

As modificações feitas no próprio algoritmo BBO são, normalmente, aplicadas nos operadores de migração e mutação. Simon [5], inicialmente, representou as taxas de migração com uma reta linear, o que nem sempre é verdade, na vida real. As modificações feitas neste ramo, tentam, de forma aproximada, simular o que se passa na realidade. No ramo da mutação, as modificações tentam sempre representar melhor os fatores externos, como relações presa-predador ou doenças.

No caso dos operadores de migração, Ma [10] apresentou cinco novos modelos, inspirados pelo comportamento natural de espécies migradoras. Neste estudo, o autor concluiu que o modelo com curvas sinusoidais apresentou melhores resultados, mas para superar a sensibilidade que um modelo possa ter em relação a um problema, o mesmo autor criou um método conjunto [25], que combina vários modelos de migração. Os resultados provaram-se bem-sucedidos, já que este operador apresentou resultados melhores em várias funções

de referência, quando comparado ao BBO base. O autor, Ma, continuou a trabalhar no desenvolvimento do BBO e em [26] ele apresentou três novas variações do BBO: o BBO baseado em emigração total, o BBO baseado em imigração total e o BBO baseado em emigração parcial. Destes três o BBO baseado em emigração total, apresentou os melhores resultados.

Chen et al. [27] desenvolveram um operador de migração BBO baseado numa matriz de covariância, de forma a reduzir a dependência do BBO relativamente ao sistema de coordenadas do problema. Feng et al. [28] melhoraram a habilidade de exploração do BBO com a migração ortogonal, que foi inspirada no crossover ortogonal dos algoritmos genéticos. De forma similar, Vanita e Deep [29] introduziram um operador de migração Laplaciano baseado no crossover Laplaciano real utilizado em algoritmos genéticos. Finalmente, Xiong et al. [30] incorporaram um modelo de migração polifilético, ou seja, que não mantém o ancestral comum de todos os indivíduos, para melhorar a exploração do algoritmo, neste modelo a migração utiliza as características de quatro soluções promissoras para construir uma solução mais promissora. Os mesmos autores combinaram a migração sinusoidal com operadores de migração de mistura (*blending*) para construírem um algoritmo BBO multi-estratégico, que obteve bons resultados.

Quanto às mudanças nos operadores de mutação, utilizados, maioritariamente, de forma a melhorar a diversidade da população, a inspiração veio de outros algoritmos evolucionários. Gong et al. [31] integraram três operadores de mutação no BBO e demonstraram a sua superioridade, relativamente ao operador do BBO base. Bansal [32] incorporou mutação polinomial num BBO com mistura (*blending*) para melhorar o seu desempenho, os resultados obtidos confirmaram que esse método era superior ao operador base. Niu et al. [33] utilizaram estratégias de mutação do algoritmo DE em conjunto com a teoria do caos, no BBO, e os resultados obtidos demonstraram os melhores resultados, para o seu problema. Na tentativa de acelerar a convergência do BBO, Lohokare et al. [34]-[35], utilizaram uma função de mutação modificada, onde a mutação vizinha do algoritmo DE foi incorporada no BBO, estes resultados também foram positivos, quando comparados ao BBO original.

Ergezer et al. [36] aplicaram aprendizagem baseada em oposição (OBL), algo comumente aplicado em outros algoritmos evolucionários e criou o BBO de oposição. Os resultados obtidos foram promissores, tendo apresentado uma boa taxa de sucesso. Roy e Mandal [37] utilizaram um novo método OBL, chamado semi-oposição, e incorporou-o no BBO. Esta modificação apresentou melhorias na taxa de convergência e a pesquisa global. Guo et al. [38] criaram um modelo novo chamado *backtracking BBO*, onde uma população secundária externa é utilizada para gravar o histórico de soluções. Esta população era, mais tarde, combinada com a população principal, criando assim a geração de soluções. Isto melhorava a capacidade de procura do algoritmo. Os resultados obtidos provaram ser competitivos, quando comparados a outros algoritmos evolucionários. O método BBO em conjunto, foi desenvolvido para resolver o problema de planeamento da área de localização

em [39]. Este método usa um grupo de abordagens diferentes para a migração e mutação de forma a preservar grupos de populações. Os resultados experimentais demonstraram que a sua abordagem era bastante robusta, na resolução do problema, apresentando um tempo de computação menor. Zheng et al. [40] utilizaram um BBO com diferentes topologias, de forma a limitar a migração e a convergência prematura e melhorar a capacidade de procura. Os seus resultados demonstraram resultados superiores aos outros métodos utilizados.

Capítulo 3

Algoritmos Evolucionários

Este capítulo irá apresentar as bases teóricas dos algoritmos evolucionários. Primeiramente, serão apresentadas informações sobre os Algoritmos Genéticos e sobre alguns dos seus operadores, como o crossover e a mutação. Depois será apresentada alguma teoria e pontos fulcrais, relativamente aos algoritmos meméticos e à sua história e desenvolvimento, tal como, o algoritmo de pesquisa local e os processos de aprendizagem. Finalmente, será apresentada informação relativamente à otimização baseada em biogeografia (BBO), tal como, os modelos matemáticos e definições deste método, alguns dos operadores apresentados na literatura, relativamente à migração e mutação, e uma breve menção a adaptações realizadas ao BBO, para problemas com restrições e otimização multiobjectivo.

3.1 Algoritmos Genéticos

Em 1858 Charles Darwin, em conjunto com Alfred Russel, apresentou a teoria da evolução por seleção natural [41]. Em 1859 Darwin publica um livro [42] onde este explica a sua teoria suportada por várias provas observadas durante a sua viagem, a bordo do Beagle. Em 1865, Gregor Mendel apresentou os princípios básicos de hereditariedade, que mais tarde em 1900, em conjunto com a teoria de Darwin, deu origem à “Teoria da Evolução”. Esta teoria combina a genética com as ideias de seleção natural de Darwin, criando o princípio básico de população genética, que define que a variabilidade entre os indivíduos numa população de organismos sexualmente reprodutores é produzida através de mutações e recombinação genética. Em 1975, Holland [1] apresentou o Algoritmo Genético (GA) e criou uma base teórica para a adaptação da evolução biológica à resolução de problemas de otimização.

Os Algoritmos Genéticos utilizam uma analogia direta do comportamento natural. Eles utilizam uma população de indivíduos, cada uma representando uma solução possível. Cada indivíduo recebe uma classificação de acordo com a sua aptidão, os indivíduos mais aptos têm uma oportunidade para reproduzir, através do crossover com outros indivíduos e, assim, novos indivíduos são produzidos. Os membros da população menos aptos têm uma probabilidade menor de serem escolhidos para a reprodução, por isso, as suas características

começam a desaparecer. Neste tipo de algoritmos, é comum utilizarem-se termos biológicos para descrever entidades análogas. Por exemplo, o termo cromossoma, em algoritmos genéticos, refere-se a um candidato a solução de um problema. Genes são blocos de bits ou apenas um bit que codificam um elemento particular do candidato a solução. Um alelo numa sequência de bits é 0 ou 1. A maioria dos GAs utilizam indivíduos haploides, indivíduos com apenas um cromossoma, ao contrário da realidade onde a maioria dos indivíduos sexualmente reproduzidos são diploides.

A Figura 3.1 apresenta uma representação esquemática em diagrama de fluxo de um algoritmo genético genérico.

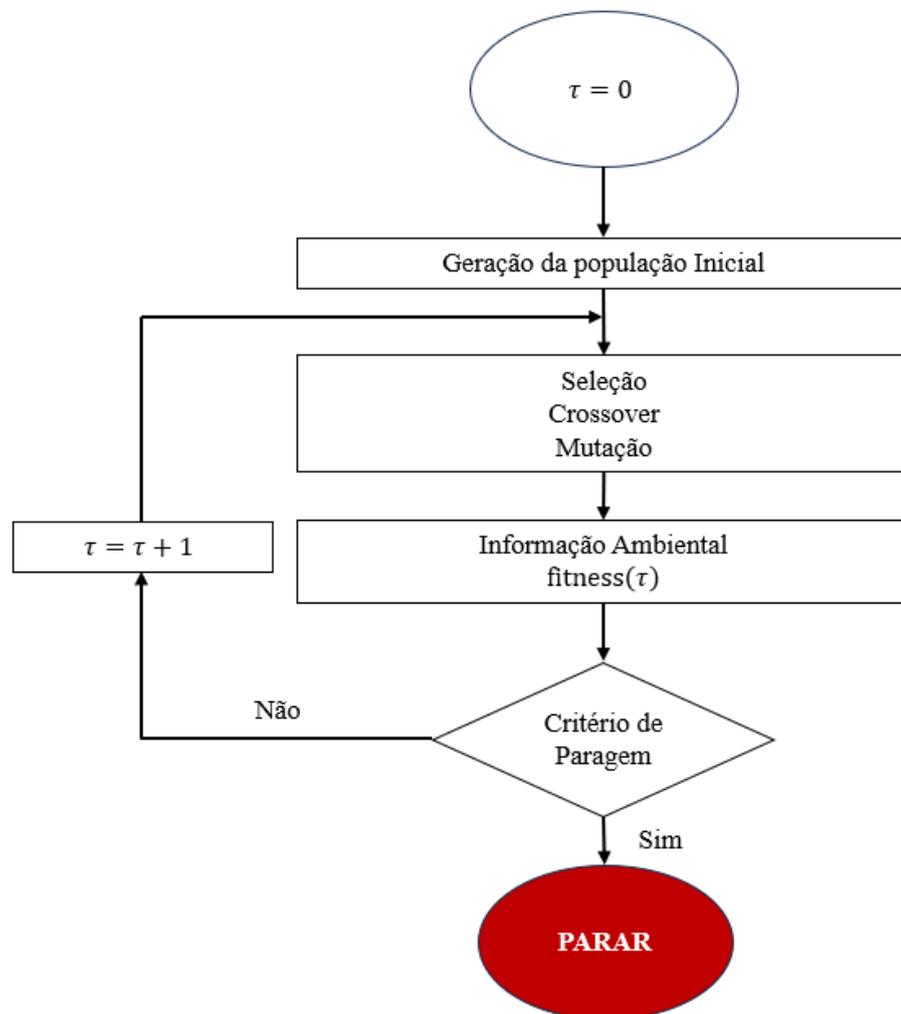


Figura 3.1: Diagrama para um GA genérico [6].

3.1.1 Definição de mérito ou aptidão

Mérito ou aptidão são as designações representativas do desempenho da solução/indivíduo, no contexto da população. De forma geral, esta avaliação é realizada usando uma função, chamada função de mérito ou função de aptidão, construída a partir da função objetivo e

das funções representando as restrições. Esta definição pode ser apresentada de três formas diferentes: o valor da função de mérito é igual ao valor da avaliação; o valor da função de mérito é definido através de uma normalização do valor da função objetivo e funções de restrição; e o valor da função de mérito é um valor ajustado, também podendo ser chamado de mérito relativo.

O valor do mérito é igual ao valor avaliado

É o método, mais simples de calcular o mérito. Neste caso, para um problema sem restrições, o valor da função de mérito é igual ao valor a ser avaliado [6],

$$\text{Mer}(\mathbf{x}) = f(\mathbf{x}) \quad (3.1)$$

O valor do mérito é definido por um processo de normalização

Considerando que a função objetivo pode ser positiva ou negativa e o problema de otimização pode pedir a sua maximização ou minimização. O método do mérito normalizado, torna-se uma melhor opção para a avaliação da aptidão de uma solução. De uma forma geral, a função de mérito pode ser definida através da seguinte expressão, num cenário onde seja requerido a minimização da solução [6],

$$\text{Mer}(\mathbf{x}_i) = C_{max} - [f(\mathbf{x}_i) + \Gamma(\mathbf{x}_i)] \quad (3.2)$$

onde $f(\mathbf{x}_i)$ representa o valor da função objetivo do problema de otimização, $\Gamma(\mathbf{x}_i) \geq 0$ é um termo restrição relacionado à violação das restrições do problema e C_{max} um escalar, definido de acordo com o âmbito do problema, para tornar a função de mérito positiva.

Alternativamente, a função de mérito também pode ser definida através do seguinte método [6],

$$\text{Mer}(\mathbf{x}_i) = \frac{1}{f(\mathbf{x}_i) + \Gamma(\mathbf{x}_i)}, \quad \text{com } f(\mathbf{x}_i) + \Gamma(\mathbf{x}_i) \neq 0 \quad (3.3)$$

Mérito relativo ou ajustado

Este tipo de avaliação é utilizado em problemas onde a convergência prematura da solução possa ocorrer, recorrendo ao mapeamento do mérito numa nova escala de valores o mérito relativo consegue contrariar esse efeito. Existem duas metodologias possíveis: o *mapeamento explícito* e o *mapeamento implícito*.

Mapeamento explícito inclui a *normalização do mérito*, o seu *ajustamento* e o seu *ordenamento*. O *processo de normalização* pode ser descrito em três passos [6]:

1. Ordenar a população, N_{pop} , de acordo com a ordem decrescente do valor de avaliação, F_i , definido em 3.2. Calcular o valor máximo e mínimo de F_i , respetivamente F_{max} e F_{min} .
2. Criar méritos a partir de um valor de referência constante, variando linearmente.

3. Os valores de referência e a taxa de mudança são parâmetros do processo.

Através deste método, o mérito do indivíduo é dado através da seguinte expressão,

$$\text{Mer}(\mathbf{x}_i) = F_{min} + \frac{F_{max} - F_{min}}{N_{pop} - 1} (\text{Rank}(i) - 1) \quad (3.4)$$

onde $\text{Rank}(i)$ é a posição relativa do indivíduo na população ordenada.

O *ajustamento do mérito* é bastante similar à *normalização do mérito*, sendo a única diferença o método para a escolha do valor a subtrair. Neste caso, o valor de mérito mínimo é gravado em cada geração e a quantidade subtraída é o mérito mínimo obtido nas últimas n gerações, sendo n , normalmente, igual a 10 [6]. Isto permitirá que a procura da solução varie não só de problema para problema, mas também a cada utilização do algoritmo. O desempenho do programa irá sofrer, na existência de indivíduos extremos, se este for excepcionalmente extremo. A presença de um indivíduo super-apto poderá causar convergência prematura, enquanto que a presença de um indivíduo super-inapto irá causar más taxas de convergência. O *ajustamento do mérito* é aplicado da seguinte forma:

1. Obter o mínimo da população avaliada

$$A_{min} = \min [A_i(\mathbf{x}), i = \dots, N_{pop}] \quad (3.5)$$

onde N_{pop} representa o tamanho da população e $A_i(\mathbf{x}) = f(\mathbf{x}_i) + \Gamma(\mathbf{x}_i)$ com $A_i(\mathbf{x}) \geq 0$.

2. Atribuir um valor de mérito a cada indivíduo \mathbf{x}_i da seguinte forma,

$$\text{Mer}(\mathbf{x}_i) = A_i - A_{min} \quad (3.6)$$

3. É recomendado usar um valor de fitness mínimo superior ao valor mínimo encontrado de forma a permitir a reprodução dos cromossomas menos aptos

O *processo de ordenação* não depende de um indivíduo extremo. Indivíduos são ordenados de acordo com o seu mérito puro e depois são designados valores de mérito reprodutivos correspondentes. Isto pode ser feito de forma linear ou de forma exponencial. Este processo assemelha-se ao processo de normalização, exceto que o processo de ordenação garante que os mapeamentos do mérito de indivíduos intermédios é realizado regularmente, reduzindo assim o efeito dos indivíduos extremos.

O *mapeamento implícito* é aplicado na seleção de progenitores, sem entrar na fase intermédia de remapeamento de mérito. Seleção por torneio é, normalmente, o método utilizado para este tipo de mapeamento, sendo o mais simples o torneio de seleção binário, onde pares de indivíduos são escolhidos aleatoriamente da população e aqueles que têm o mérito mais elevado são copiados para o grupo de reprodução. Assim, em vez da população inteira ser substituída, apenas alguns dos indivíduos serão afetados. Este modelo é mais parecido com o que ocorre na natureza, promovendo a competição entre progenitores e a sua descendência.

3.1.2 Aplicação de Restrições

Um problema de otimização pode ser formulado da seguinte forma,

$$\min_{\mathbf{x} \in \mathbb{R}^n} \{f(\mathbf{x}) : \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\} \quad (3.7)$$

onde o espaço de projeto \mathbf{X} é definido como

$$\mathbf{X} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\} \quad (3.8)$$

Assumem-se dois tipos de restrições: restrições de igualdade ($\mathbf{h}(\mathbf{x}) = \mathbf{0}$) e restrições de desigualdade ($\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$). Um indivíduo da população é considerado uma solução admissível se ele satisfizer todas as restrições. Uma restrição de desigualdade é considerada ativa num ponto \mathbf{x} se $g_i(\mathbf{x}) = 0$ e inativa em \mathbf{x} se $g_i(\mathbf{x}) < 0$. Uma restrição de igualdade é considerada ativa a qualquer ponto admissível. A restrição ativa num ponto admissível \mathbf{x} restringe a admissibilidade do domínio na vizinhança de \mathbf{x} , enquanto os outros pontos, inadmissíveis, não têm qualquer influência na vizinhança de \mathbf{x} .

Condições necessárias de Primeira Ordem para problemas com restrições

Se um ponto \mathbf{x}^* , denotado como um ponto regular, garante que os vetores $\nabla h_i(\mathbf{x}^*)$ e $\nabla g_j(\mathbf{x}^*)$ são linearmente independente. Sendo que $j \in \mathbf{J}$ e \mathbf{J} representa todos os índices j onde $g_j(\mathbf{x}^*) = 0$.

As condições de otimalidade de Karush-Khun-Tucker, que afirmam que se um ponto \mathbf{x}^* é um ponto regular e um mínimo relativo do problema 3.7, então deverá existir um vetor $\lambda \in \mathbb{R}^m$ e um vetor $\mu \in \mathbb{R}^p$ com $\mu \geq \mathbf{0}$ tal que,

$$\nabla f(\mathbf{x}^*) + \lambda^T \nabla \mathbf{h}(\mathbf{x}^*) + \mu^T \nabla \mathbf{g}(\mathbf{x}^*) = \mathbf{0} \quad (3.9)$$

$$\mu^T \mathbf{g}(\mathbf{x}^*) = 0 \quad (3.10)$$

Condições de Segunda Ordem

Condições Necessárias de Segunda Ordem: assumindo que as funções $\mathbf{f}, \mathbf{g}, \mathbf{h} \in C^2$ e que \mathbf{x}^* é um ponto regular das restrições. Se \mathbf{x}^* é um mínimo relativo do problema, então existem dois vetores $\lambda \in \mathbb{R}^m$ e $\mu \in \mathbb{R}^p$, com $\mu \geq \mathbf{0}$ tal que 3.9 e 3.10 sejam verificadas e que,

$$\mathbf{H}_L(\mathbf{x}^*) = \mathbf{H}_f(\mathbf{x}^*) + \lambda^T \mathbf{H}_h(\mathbf{x}^*) + \mu^T \mathbf{H}_g(\mathbf{x}^*) \quad (3.11)$$

é positiva semidefinida no sub espaço tangente das restrições ativas em \mathbf{x}^* , onde $\mathbf{H}_f, \mathbf{H}_h$ e \mathbf{H}_g são as matrizes Hessianas das funções f, g, h , respetivamente.

Condições Suficientes de Segunda Ordem: considere-se que $\mathbf{f}, \mathbf{g}, \mathbf{h} \in C^2$. As condições suficientes para um ponto \mathbf{x}^* satisfazer as restrições ser um ponto mínimo estri-

tamente relativo do problema é que existam dois vetores $\lambda \in \mathbb{R}^m$ e $\mu \in \mathbb{R}^p$, tal que,

$$\mu \geq \mathbf{0} \quad (3.12)$$

$$\mu^T \mathbf{g}(\mathbf{x}^*) = 0 \quad (3.13)$$

$$\nabla f(x^*) + \lambda^T \nabla \mathbf{h}(\mathbf{x}^*) + \mu^T \nabla \mathbf{g}(x^*) = 0 \quad (3.14)$$

e a matriz Hessiana

$$\mathbf{H}_L(\mathbf{x}^*) = \mathbf{H}_f(\mathbf{x}^*) + \lambda^T \mathbf{H}_h(\mathbf{x}^*) + \mu^T \mathbf{H}_g(\mathbf{x}^*) \quad (3.15)$$

é definida positiva semidefinida no subespaço

$$M' = \{\mathbf{y} : \nabla \mathbf{h}(\mathbf{x}^*) \mathbf{y} = \mathbf{0}, \nabla g_j(\mathbf{x}^*) \mathbf{y} = 0 \text{ para todos os } j \in J\} \quad (3.16)$$

onde

$$J = \{j : g_j(\mathbf{x}^*) = 0, \mu_j > 0\} \quad (3.17)$$

Aplicação em Algoritmos Genéticos

Existem vários métodos para lidar com a aplicação de restrições em problemas de otimização com Algoritmos Genéticos, e podem ser agrupados em cinco classes:

- Métodos baseados em preservar a viabilidade das soluções;
- Métodos baseados em funções de penalidade;
- Métodos baseados na procura de soluções admissíveis;
- Métodos baseados em decodificadores;
- Métodos híbridos.

Os métodos mais comuns, são os métodos baseados em funções penalidade, que penalizam as soluções não admissíveis, através de uma função penalidade exterior. De forma geral, estas funções são determinadas através da distância da solução a ser analisada ao espaço de soluções admissíveis Ω . Este processo é realizado ao usar um conjunto de funções u_j ($j = 1, \dots, m$) para impor a penalidade, onde u_j é uma medição da violação do conjunto de restrições de ordem j ,

$$u_j = \begin{cases} \max[0, g_j(\mathbf{x})], & \text{se } 1 \leq j \leq q \\ |h_j(\mathbf{x})|, & \text{se } q + 1 \leq j \leq m \end{cases} \quad (3.18)$$

Na literatura estes métodos são referidos, normalmente, como métodos de penalidade. Estes apresentam diferenças significantes entre si, tanto na função de penalidade usada

quanto na forma como esta é aplicada. Por exemplo, Joines et al [43] propôs um método de penalidade dinâmica, onde, num problema de minimização, as soluções são avaliadas, por cada k iteração, pela expressão,

$$Aval(\mathbf{x}) = f(\mathbf{x}) + (Ck)^\alpha \sum_{j=1}^m [u_j(\mathbf{x})]^\beta \quad (3.19)$$

onde C, α, β são constantes.

Outro processo que utiliza penalidades adaptáveis, o método de Hadj-Alouane e Bean [44], utiliza uma penalidade dependente do processo evolucionário e da avaliação de cada indivíduo através da seguinte fórmula,

$$Aval(\mathbf{x}) = f(\mathbf{x}) + \lambda(k) \sum_{j=1}^m [u_j(\mathbf{x})]^2 \quad (3.20)$$

com $\lambda(k)$ a ser atualizado em cada k gerações, de acordo com o estado da pesquisa.

3.1.3 Operador Crossover

Inicialmente, a combinação genética era o procedimento mais importante no desenvolvimento do operador crossover, sendo o seu objetivo a partilha de informação entre os cromossomas. Nos dias de hoje, a análise baseada no “*Scheme Theorem*” de Holland [45] tornou-se complicada devido à existência de estratégias elitistas, regras heurísticas e otimizadores locais. Este facto implica a identificação de mecanismos de recombinação e uma análise taxonómica para entender os esquemas de operador crossover propostos. Para isso, eis alguns aspetos a considerar quando se desenvolve um operador de crossover:

- Hereditariedade: considerado o aspeto mais importante do operador crossover, este deve herdar o material genético de ambos os progenitores. Um operador só é um operador de recombinação pura se dois indivíduos idênticos criarem descendências semelhantes, antes da mutação inserir diversidade na população. Um operador de crossover OX é consistente se as propriedades comuns em ambos os parentes são preservados e ordenados se a distância entre ambos os progenitores e o descendente é inferior ou igual à distância entre os parentes:

$$d(\mathbf{p}_1, \mathbf{s}) \leq d(\mathbf{p}_1, \mathbf{p}_2) \quad \text{e} \quad d(\mathbf{p}_2, \mathbf{s}) \leq d(\mathbf{p}_1, \mathbf{p}_2), \forall \mathbf{s} \in \mathbf{O}(\mathbf{p}_1, \mathbf{p}_2, OX) \quad (3.21)$$

onde $\mathbf{O}(\mathbf{p}_1, \mathbf{p}_2, OX)$ representa o conjunto de todos os possíveis descendentes gerados pelo operador de crossover OX, \mathbf{p} representa os progenitores e \mathbf{s} a descendência.

- Validade: o operador deve produzir soluções válidas, sendo esta condição apenas violável em problemas de otimização com restrições.

Mecanismos de seleção

- **Seleção dos Pares Acasalamento (MSM):** define o processo de seleção dos grupos de cromossomas dos progenitores antes de ser aplicado o crossover. Divide-se em dois grupos: seleção proporcional, que seleciona indivíduos baseados no seu mérito e seleção baseada na ordenação, que seleciona indivíduos baseados na sua classificação/ordenação na população ordenada.
- **Geração de Descendentes (OGM):** Tenta produzir novos cromossomas a partir dos pais selecionados no MSM. Este mecanismo permite que material genético seja transferido de pais para filhos. Isto pode ser realizado no domínio binário ou real.
- **Seleção de Sobreviventes na próxima geração (OSM):** Começando pelo descendente gerado por cada conjunto de pais, este mecanismo escolhe os indivíduos que irão ser membros da população.

A maioria dos operadores de crossover propostos na literatura são baseados na geração de dois cromossomas descendentes por cada par de progenitores. No entanto, também existem alguns casos onde operadores baseados em operadores multi-progenitor e crossover multi-descendentes foram utilizados.

Análise de Taxonomia dos operadores de crossover [46]

- **Operadores Discretos (DCO):** estes operadores assumem que o valor de cada gene no cromossoma do filho coincide com o valor do mesmo gene num dos pais. Para obter valores dos genes de um descendente dos genes dos pais não existe nenhuma transformação numérica. Exemplos de operadores pertencentes a esta categoria são o single-point crossover (ponto único), two-point crossover (duplo ponto) e o multi-point crossover (múltiplos pontos).
- **Operadores Baseados em Agregação (ABCO):** esta categoria de operadores engloba os operadores que utilizam uma função agregadora para combinar os valores dos genes dos pais, numericamente, e gera os valores dos genes dos descendentes.
- **Operadores Baseados em Vizinhanças (NBCO):** neste conjunto de operadores, os genes dos filhos são obtidos por intervalos definidos na vizinhança associada aos genes dos pais utilizando uma função de distribuição de probabilidade. Operadores que se enquadram nesta categoria são o BLX-a, recombinação fuzzy e o crossover binário simulado.

Operadores de Crossover no domínio binário

Esta operação é simulada através de uma máscara, representada por um vetor \mathbf{M}_i cujos componentes são valores binários. Este vetor tem uma dimensão igual ao número de

cromossomas a serem combinados, l_c . A Figura 3.2 apresenta um algoritmo demonstrativo do uso do vetor máscara.

```

If  $M_i = 0$  then
     $s_1 \leftarrow P_1$ 
     $s_2 \leftarrow P_2$ 
Else If
     $s_1 \leftarrow P_2$ 
     $s_2 \leftarrow P_1$ 
End If

```

Figura 3.2: Pseudo algoritmo de aplicação do vetor máscara [6].

Single-Point Crossover (SP)

Este operador consiste em dois pontos:

1. Considerar dois progenitores com cromossomas de comprimento l_c
2. Selecione aleatoriamente um número k , designado de ponto de corte (*cut-off*), tal que, $0 < k < l_c$,
 - (a) O primeiro filho receberá todos os genes do primeiro pai de 1 a k e todos os genes do segundo pai de $k + 1$ a l_c
 - (b) O segundo filho irá receber os genes de ambos os parentes de forma inversa ao primeiro filho.

Este tipo de operador costuma ter os piores resultados de todos os operadores tradicionais.

Multi-Point Crossover (MP)

O Multi-Point Crossover (MP) pode ser interpretado como uma generalização do operador Single-Point obtido para n pontos de corte. De forma geral, o MPX pode ser visto como um caso especial de um com $m > n$ pontos, onde um ou mais $(m - n)$ pontos de corte selecionados estão localizados no final do cromossoma.

Uniform Crossover (U) [47]

Num SP ou MP a troca de material genético é feito de acordo com certas posições no cromossoma, chamados pontos de corte. No Uniform Crossover (U) este aspeto é generalizado de forma que cada cromossoma possa ser considerado um ponto de corte. Para cada gene da descendência, o crossover U escolhe um gene de um dos progenitores. Este processo acontece de gene em gene de forma sequencial. O primeiro filho recebe o

gene do primeiro pai se a máscara, que foi gerada de forma aleatória previamente, nesse cromossoma tiver um valor igual a 0, ou gene do segundo pai se o valor da máscara for igual a 1. Normalmente, os genes do segundo filho são gerados de forma complementar ao primeiro filho, ou seja, segue a ordem inversa da substituição dos genes do primeiro filho.

Operadores de Crossover no domínio real

Para representações em valores reais, existem outros operadores, além dos que já foram apresentados. Os mais comuns são a recombinação centrada na média e o crossover centrado nos pais. Na recombinação centrada na média, a descendência é gerada o mais próximo o possível do centroide dos seus progenitores. Os operadores que seguem esta filosofia são os seguintes:

- **Crossover Intermédio:** estes tentam fazer a média dos elementos genéticos dos pais. Dados dois pais, o crossover aritmético cria um descendente cujos genes são definidos usando uma média ponderada

$$s_i = \alpha x_{1i} + (1 - \alpha) x_{2i} \quad (3.22)$$

onde x_{1i} e x_{2i} representam o material genético dos pais e α é o fator de ponderação.

- **Crossover Geométrico [48]:** este processo gera filhos da seguinte forma,

$$s_i = ((x_{11}x_{21})^{0,5}, \dots, (x_{1n}x_{2n})^{0,5}) \quad (3.23)$$

- **Crossover com Distribuição Normal Unimodal:** neste operador, $(\mu - 1)$ pais \mathbf{x}^i são selecionados aleatoriamente da população. O valor médio $\bar{\mathbf{x}}$ dos indivíduos selecionados são calculados. De seguida, $(\mu - 1)$ vetores direcionais são gerados, $\mathbf{d}^i = \mathbf{x}^i - \bar{\mathbf{x}}$. A variável \mathbf{e}^i designa o cosseno do diretor $\mathbf{d}^i / \|\mathbf{d}^i\|$. Os descendentes são gerados da seguinte forma,

$$\mathbf{y} = \bar{\mathbf{x}} + \sum_{i=1}^{\mu-1} w_i \|\mathbf{d}^i\| \mathbf{e}^i + \sum_{i=\mu}^n \tau_i D \mathbf{e}^i \quad (3.24)$$

onde w_i e τ_i representam variáveis distribuídas normalmente com média de zero e variância σ_w^2 e σ_τ^2 , respetivamente

- **Crossover Simplex [49]:** este operador gera descendência em torno da média. O operador retém o descendente numa região predefinida, um simplex similar, mas com um fator $\beta = \sqrt{\mu + 1}$ maior que o simplex do pai. Este operador utiliza uma função de distribuição normal padronizada, para a geração do descendente

Recombinação centrada nos progenitores

Na recombinação centrada nos progenitores, os descendentes são gerados o mais próximo possível aos seus progenitores. Cada progenitor tem uma probabilidade igual de gerar descendência na sua vizinhança. Os seguintes operadores podem ser utilizados:

- **Crossover Binário Simulado** [50]: um número aleatório μ_i entre 0 e 1 é gerado, seguindo a seguinte função de distribuição probabilística,

$$P(\beta) = \begin{cases} 0,5(\tau + 1)\beta_i^\tau, & \text{se } \beta_i \leq 1 \\ 0,5(\tau + 1)\beta_i^{\frac{1}{\tau+2}}, & \text{se } \beta_i > 1 \end{cases} \quad (3.25)$$

onde o índice de distribuição τ dá uma probabilidade maior de criar um filho mais próximo dos seus progenitores. O βq_i é calculado de forma que a área por baixo da curva de probabilidade de 0 a βq_i é igual ao número aleatoriamente gerado através de,

$$\beta q_i = \begin{cases} 2\mu_i^{\frac{1}{\tau+1}}, & \mu_i \leq 0,5 \\ \left[\frac{1}{2(1-\mu_i)} \right]^{\frac{1}{\tau+1}}, & \mu_i > 0,5 \end{cases} \quad (3.26)$$

Para um único objetivo, τ é normalmente igual a 2, enquanto que no caso multi-objetivo $\tau \in [5, 10]$. Os dois filhos serão calculados da seguinte forma.

$$y_i^1 = 0,5 [(1 + \beta q_i)x_i^1 + (1 - \beta q_i)x_i^2] \quad (3.27)$$

$$y_i^2 = 0,5 [(1 - \beta q_i)x_i^1 + (1 + \beta q_i)x_i^2] \quad (3.28)$$

onde x_i^1 e x_i^2 são os progenitores e y_i^1 e y_i^2 são os filhos.

- **Crossover centrado nos pais** [51]: o valor médio $\bar{\mathbf{x}}$ dos indivíduos selecionados é calculado. Um pai \mathbf{x}^p é selecionado para cada filho, com probabilidade igual. De seguida o vetor direcional $\mathbf{d}^p = \mathbf{x}^p - \bar{\mathbf{x}}$ é determinado. De cada um dos $(\mu - 1)$ pais, distâncias perpendiculares D_i da linha \mathbf{d}^p são calculadas e a sua média \bar{D} é calculada. O descendente é gerado da seguinte forma,

$$y = \mathbf{x}^p + w_\delta \|\mathbf{d}^p\| + \sum_{i=1, i \neq p}^{\mu} \omega_\tau \bar{D} \mathbf{e}^i \quad (3.29)$$

onde w_δ e ω_τ são variáveis distribuídas normalmente com média zero e uma variância σ_δ^2 e σ_τ^2 , respetivamente

- **Crossover mistura BLX-alpha** [52]: este operador combina os genes de dois progenitores para gerar os filhos, amostrando um valor obtido no intervalo $[\min_i - Ia, \max_i + Ia]$ a cada posição i . Neste caso, \min_i e \max_i são o valor mínimo e máximo dos genes dos pais na posição i . O valor de I é igual a $\max_i - \min_i$. O descendente é gerado

da seguinte forma,

$$y_i = (\min_i - Ia) + \alpha |(\max_i + Ia) - (\min_i - Ia)| \quad (3.30)$$

onde, normalmente, $\alpha = 0,5$

3.1.4 Mutação

Operadores de mutação são operadores unários, que atuam em apenas um indivíduo. Este processo representa pequenas mudanças, nos indivíduos selecionados. A probabilidade de mutação, normalmente denotado por P_m , está definida para cada gene. De forma geral, valores pequenos ($P_m \in [0,001; 0,01]$) são preferíveis para este parâmetro. Embora também existam alguns casos em que a probabilidade de mutação é calculada com $1/k$, em que k é o número de variáveis do problema. A seleção de um operador tem de ter em conta os seguintes aspetos:

Ergodicidade: o operador de mutação deve poder adotar todas as soluções do espaço de pesquisa possam ser alcançadas

Validade: o operador deve produzir soluções válidas, o que nem sempre será possível em problemas de otimização com restrições

Localidade: a propriedade mais importante do operador de mutação. Localidade refere-se ao efeito no fenótipo, quando este realiza mudanças no genótipo. Quando mudanças pequenas ocorrem no genótipo, o fenótipo deve revelar mudanças pequenas. Isto implica uma localidade forte. Localidade fraca é caracterizada pelos grandes efeitos no fenótipo quando ocorrem mudanças pequenas no genótipo. Em casos extremos, a pesquisa irá convergir numa pesquisa aleatória no domínio do problema.

Operadores de Mutação para o domínio binário

Mutação de troca genética

Neste caso, o material genético de um indivíduo é sujeito a uma modificação aleatória, com uma certa probabilidade P_m de forma a preservar diversidade (prevenir a solução de ficar presa num mínimo/máximo local). A Figura 3.3 demonstra uma representação deste operador de mutação. Neste caso, os vetores de cima representam as soluções antes de serem mutadas e os de baixo as soluções após serem mutadas. Os *bits* a negrito e rodeados a vermelho representam os genes selecionados a serem mudados. Estes irão ser modificados através do processo *Bit Flip*, onde se o valor for zero ele passa para um e vice-versa.



Figura 3.3: Mutação de troca genética do tipo "flip-bit".

Mutação para o domínio real

O caso mais utilizado de operadores de mutação no domínio real tem o aspecto seguinte,

$$\mathbf{x}' = \mathbf{x} + \mathbf{M} \quad (3.31)$$

onde \mathbf{M} é uma variável aleatória. O valor de \mathbf{M} pode ser calculado de formas diferentes formas.

O primeiro caso, a mutação uniforme aleatória, gera um valor aleatório uniforme no intervalo $[a, b]$, onde o parâmetro a é, normalmente, igual a $-b$. O produto da mutação é gerado dentro dos limites da seguinte forma,

$$\mathbf{x}' = \mathbf{x} + \mathbf{U}(-b, b)^n \quad (3.32)$$

onde b é uma constante definida pelo utilizador.

A segunda opção, a mutação distribuída normalmente, utiliza uma distribuição normal $\mathbf{M} = \mathbf{N}(\mathbf{0}, \sigma)$, em que $\mathbf{N}(\mathbf{0}, \sigma)$ é um vetor independente de números aleatórios que seguem a distribuição de Gauss com uma média igual a zero e desvio padrão σ , sendo este o parâmetro mais crucial.

O terceiro caso, a mutação polinomial, o descendente é criado da seguinte forma,

$$x'_i = x_i + (x_i^U - x_i^L)\delta_i \quad (3.33)$$

onde x_i^U e x_i^L representam os limites superiores e inferiores de x_i , respetivamente. O parâmetro δ_i é calculado a partir do polinômio da função da distribuição da probabilidade:

$$P(\delta) = 0,5(\tau_m + 1)(1 - |\delta|^{\tau_m}) \quad (3.34)$$

$$\delta_i = \begin{cases} (2r_i)^{\frac{1}{\tau_m+1}-1}, & r_i < 0,5 \\ 1 - (2(1 - r_i))^{\frac{1}{\tau_m+1}}, & r_i \geq 0,5 \end{cases} \quad (3.35)$$

onde τ_m representa o índice da distribuição normal (normalmente $\tau_m \approx 20$) e r_i é um número aleatório definido no intervalo $[0, 1]$.

3.1.5 Outros parâmetros

Mecanismos de seleção de sobreviventes

Este parâmetro é utilizado após a geração da descendência, pelos pais. Como o tamanho da população é, quase sempre, constante, será necessário escolher os indivíduos que passarão para a nova geração. Esta decisão é, normalmente, baseada no mérito, favorecendo aqueles de maior aptidão, mas o conceito de idade também poderá ser utilizado. Ao contrário da escolha de pais, que é estocástica, a seleção de sobreviventes é determinística. A seleção por idade irá utilizar apenas os filhos criados no crossover [6].

Substituição por similaridade fenotópica ou genética

Monitorizar a diversidade da população, é um aspeto muito importante, visto que, este permite antecipar a necessidade de tomada de ação se existir a possibilidade de diversidade genética. A diversidade da população será alta na fase inicial, estável na fase intermédia e baixa na fase final do processo de otimização. Seguindo esta perspectiva deveria haver um equilíbrio entre a diversificação e a intensificação.

Geração da População Inicial

A inicialização da população é um processo crucial, visto que, esta cria a base do desempenho do algoritmo. A população inicial, normalmente gerada de forma aleatória, é essencial para manter a diversidade das soluções, impedindo a convergência prematura num ótimo local. O outro método utilizado para a criação da população, o método heurístico, tende a gerar boas soluções, mas carece de diversidade. Com isto, também é possível gerar uma população de forma híbrida, usando os métodos heurísticos para a geração de alguns membros da população, que serão boas soluções, e complementando-os com a geração aleatória dos restantes indivíduos.

O tamanho da população inicial também um ponto de extrema importância, já que, uma população enorme irá diluir as soluções, resultando numa convergência mais lenta, mas uma população muito pequena, poderá não oferecer um processo de acasalamento diverso o suficiente.

3.2 Algoritmos Meméticos

Em 1976, Richard Dawkins introduziu uma nova versão da interpretação do Darwinismo clássico, que afirma que o gene é a unidade fundamental da evolução, em vez do cromossoma. Na publicação de “*The Selfish Gene*” [2] Dawkins usa o conceito de meme para analisar a transmissão cultural, da mesma forma, que o gene é utilizado na transmissão bi-

ológica. Este parâmetro pode ser replicado de acordo com a sua utilidade ou popularidade relativamente à sua comunidade, sendo transmitido através dos meios de comunicação.

O primeiro algoritmo memético (MA), era considerado uma fusão entre procura global baseada em populações e procura local heurística. Assim, o aspeto global é controlado pela natureza evolucionária do algoritmo e o aspeto local pela abordagem heurística. Os algoritmos meméticos podem ser divididos pelos vários tipos de abordagem e estas são:

- Algoritmo Evolucionário Híbrido [53]
- Procura Local Genética [54]
- Algoritmo Cultural [55]
- Algoritmo Evolucionário Lamarckiano [56]
- Algoritmo Evolucionário Baldwiniano [56]
- Algoritmo Memético de Difusão [57]

Em 1998, surgiu o Algoritmo do Gene Egoísta (SG) [58], onde a sobrevivência do mais apto é considerada uma batalha entre genes e não entre indivíduos. O processo de aprendizagem é baseado no desempenho mútuo da informação de cada gene.

A Teoria do Gene Egoísta, apresentada por Dawkins, utilizava o termo gene egoísta para retratar uma vista evolutiva centrada nos genes. Esta teoria sugere que a evolução é melhor interpretada ao considerar que esta age ao nível do gene, assim o mecanismo de seleção é dominado por seleção à base de genes. O termo 'meme' também havia sido introduzido por Dawkins, como uma unidade de evolução cultural humana análoga ao gene, sugerindo que uma reprodução egoísta também poderá modificar a transmissão cultural. Exemplos práticos de memes são: modas, frases de impacto ou ideias etc.

Genes propagam-se ao juntar-se com outros genes, saltando de corpo a corpo, da mesma forma, memes propagam-se ao juntar-se a outros memes, saltando de cérebro para cérebro, num processo que num sentido mais amplo pode ser chamado de replicação. Dawkins integrou este método no processo de evolução e designou esta fusão de Darwinismo Universal. Em particular, este método sugere que a evolução não é exclusiva dos sistemas biológicos ou dos genes, mas também é aplicável a qualquer sistema complexo que demonstre os princípios de hereditariedade, variação e seleção. Os parâmetros necessários para se cumprir os requerimentos de um sistema evolutivo.

3.2.1 Processos de Aprendizagem

Os dois processos, Lamarckiano [3] e Baldwiniano [4], são os processos mais básicos de aprendizagem, presentes em MAs. O processo Lamarckiano propõe que o melhoramento obtido através da procura local é refletido no genótipo, através da inclusão do novo indivíduo na população original, de forma a permitir oportunidades de reprodução. Este processo

de aprendizagem é baseado no mecanismo de transmissão hereditária das características obtidas, apresentado por Lamarck no século XIX.

No outro lado, o efeito de Baldwin sugere um mecanismo, onde a evolução é dirigida até um ponto favorável, com a adaptação sem a necessidade da hereditariedade. Assim, neste processo existem apenas mudanças no mérito dos indivíduos e o novo genótipo melhorado não é incluído na população geral. O processo Lamarckiano acelera a convergência da pesquisa evolucionária, aumentando as chances de convergência prematura. Enquanto que o processo Baldwiniano mantém um pouco da diversidade populacional, com o custo da redução da velocidade do algoritmo.

O Algoritmo do gene egoísta, utiliza um terceiro processo, baseado numa população virtual (PV). Neste método, a PV é atualizada em função das regras, baseadas em recompensas e castigos para os vencedores e perdedores do torneio, respetivamente. Em versões mais recentes, esta população virtual contém vetores de frequências acumuladas de alelos seguindo uma distribuição de probabilidade para cada gene.

3.2.2 Aspectos Essenciais dos Algoritmos Meméticos

O método mais comum na construção de algoritmos meméticos é a integração de evolução cultural baseada no conceito de meme de Dawkins. Esta adaptação é feita ao aplicar uma ou mais fases de melhoria no mérito nos novos indivíduos da população da geração. Ou seja, a pesquisa local é aplicada nos indivíduos gerados a partir da mutação ou recombinação genética. O uso de conhecimento específico ao problema e o uso de métodos heurísticos dentro do algoritmo, é algo que deverá ser aplicado baseado no comportamento do algoritmo e a experiência. Este conceito leva a uma série de questões:

- Qual será a melhor a priorizar, pesquisa global ou pesquisa local?
- Onde e quando deverá ser aplicada a pesquisa local, no ciclo evolucionário?
- Quais indivíduos serão os melhores a ser escolhidos, e como deverão estes ser selecionados?
- Quanto esforço computacional deverá ser colocado na procura local?
- Como é que se deve aplicar os operadores genéticos, na pesquisa local, de forma a promover a sinergia?

Sendo um dos processos de maior importância, na história e desenvolvimento dos algoritmos meméticos, o algoritmo de pesquisa local e os seus processos, serão expostos sucintamente, neste trabalho.

Algoritmo de Pesquisa Local

Os três componentes a considerar quando se implementa um algoritmo de pesquisa local são a regra pivô, a profundidade de pesquisa local e a função de geração de vizinhança.

A **regra pivô** define o critério usado para aceitar um indivíduo com características benéficas. Num problema de maximização, este processo, determina o final do loop interno, apenas após ter pesquisado a função de vizinhança inteira ou assim que encontrar um valor melhor. Na prática, poderá haver casos onde a função vizinhança considerada é limitada aleatoriamente, caso esta seja muito grande para ser pesquisada.

A **profundidade de procura local** define a condição de paragem do loop externo, baseado no valor da melhoria obtida através da pesquisa local. Vários estudos foram realizados relativamente à influência deste parâmetro no desempenho da pesquisa local, no contexto do MAs, tanto quanto ao tempo demorado quanto à qualidade da solução produzida. A **função de geração de vizinhança**, $f(i)$, define um conjunto de pontos que poderão ser obtidos ao aplicar um operador de deslocação ao ponto i , dentro do espaço de procura. Esta aplicação pode ser representada com um gráfico $G = (\nu, e)$ onde os vértices ν são pontos no espaço de pesquisa e os eixos e estão relacionados com o operador de deslocamento, por exemplo, $e_{ij} \in G \Leftrightarrow j \in f(i)$.

A utilização de múltiplos operadores de pesquisa local em MAs é algo comum. O problema está relacionado ao facto de que cada, operador realiza uma pesquisa local tendenciosa, ou seja, o método torna-se mais eficiente para algumas classes de problemas e menos eficiente para outras. Assim, conclui-se que a pesquisa local é algo dependente do problema. Como desenvolver operadores locais evitando escolhas inadequadas, é algo que tem recebido bastante atenção. Logo, a maioria dos autores prefere utilizar vários operadores de procura local. O Algoritmo 1 demonstra um exemplo de um pseudocódigo de um otimizador local.

Algoritmo 1: Pseudocódigo de um operador de procura local [6].

```

/* Receber uma solução inicial  $i$  e uma função de vizinhança  $f$  */
Set best  $\leftarrow i$ 
Set iter  $\leftarrow 0$ 
while condição de profundidade não é satisfeita do
  Set contar  $\leftarrow 0$ 
  while regra pivô não é satisfeita do
    Próxima geração  $j$  em  $f(i)$  Set contar  $\leftarrow$  contar + 1
    if  $f(j) > f(\text{best})$  then
      Set best  $\leftarrow j$ 
    end if
  end while
  Set  $i \leftarrow$  best
  Set iter  $\leftarrow$  iter + 1
end while

```

Aprendizagem Lamarckiana e o Efeito de Baldwin

A pesquisa local descrita anteriormente, assume que a solução atualmente em análise é sempre substituída pela solução vizinha com um mérito superior, se for encontrada. No contexto do algoritmo, a pesquisa local pode ser considerada uma fase de desenvolvimento de conhecimento, dentro de cada geração. Desta forma, é possível escolher considerar as mudanças sofridas por um indivíduo durante o processo ou recompensar o membro original da população, atribuindo-lhe uma melhoria no seu mérito, resultante do processo de procura local. A abordagem baseada no efeito de Baldwin é a que se mais assimila à teoria de Darwin, assumindo que não existe a herança de características adquiridas, mas sim a introdução da plasticidade fenotópica no modelo.

Plasticidade fenotópica, é definida como a flexibilidade e criatividade do organismo ou sistema ao adaptarem o seu comportamento ao ambiente onde eles estão envolvidos durante toda a sua vida. Em termos práticos, os estudos mais recentes tendem a apresentar ou um modelo puramente Lamarckiano, ou uma combinação probabilística das duas abordagens.

Preservar a diversidade populacional

Convergência prematura, ocorre quando a população converge na vizinhança de um ponto ótimo local. Este fenômeno pode ocorrer, em MAs, quando ocorre uma perda de diversidade, chamado procura local total e quando ocorre uma convergência prematura para regiões de grande atração, chamado procura local incompleta. Existem várias soluções para contornar este problema, a seguinte lista apresenta algumas dessas soluções:

- Durante a geração da população, inclui-se apenas uma pequena percentagem de indivíduos conhecidos com bom mérito.
- Aplicar procura local apenas a uma pequena fração da população. Isto permite garantir diversidade no resto da população.
- Utilizar múltiplos otimizadores locais, onde cada um induz um espaço de procura diferente.
- Utilizar m critério que explicitamente controla a diversidade, tal como, a regra pivô faz com a pesquisa local.
- Modificar o operador de seleção, para prevenir a duplicação de indivíduos da população.

Seleção de operadores

O fator mais importante na criação de MAs é a escolha de uma heurística. É necessário escolher um método de procura local, cujo operador de deslocamento não seja o mesmo que o operador usado no *crossover* e na mutação da pesquisa evolucionária. Assim é possível obter, no âmbito local e global, uma pesquisa diferenciada. Em alguns casos, a informação

relacionada com o domínio de procura pode ser usada para guiar a pesquisa, de acordo, com a sua estrutura vizinha. Mas, a escolha ótima de operadores depende não só do tipo de problema, mas também do ponto da pesquisa evolucionária nesse momento.

Uma forma simples, de contornar estes problemas é o uso de diversos operadores de pesquisa local, como foi mencionado anteriormente. O uso de um conjunto de estratégias diferentes é análogo à definição de meme, apresentada por Dawkins. Num outro ponto de vista, uma meme pode ser transferida de um indivíduo para o outro, a qualquer instante no ciclo de vida, sugerindo que este mecanismo tem um grande impacto na distribuição de memes pela população.

3.3 Otimização baseada em Biogeografia

3.3.1 Modelos Matemáticos

A Figura 3.4, apresenta um modelo de abundância de espécies num único habitat. O rácio de emigração μ e o rácio de imigração λ são funções do número de espécies no habitat. Considerando a curva de imigração, o seu máximo ocorrerá quando a taxa de migração do habitat é I , que ocorrerá quando o número de espécies no habitat é zero. Com o aumento do número de espécies, o processo de imigração tornar-se-á mais competitivo, diminuindo, assim, o número de espécies que conseguirão completar o processo, resultando no decréscimo da taxa de imigração. O número máximo de espécies presentes num habitat é definido como S_{max} , nesse ponto a taxa de imigração será zero. O oposto se passará na curva de emigração, o seu máximo, E , irá ocorrer quando a população tiver atingido o seu máximo e o seu mínimo, quando não houver nenhuma espécie.

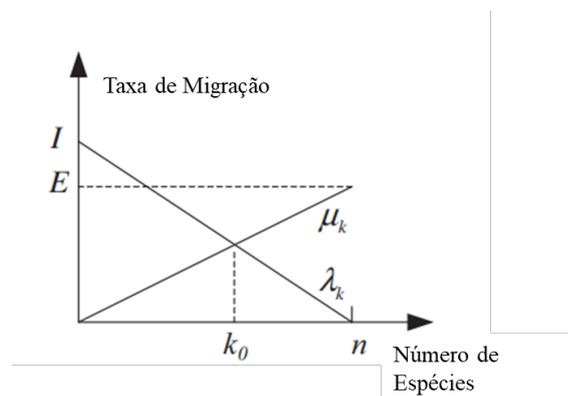


Figura 3.4: Representação gráfica do modelo de migração linear ([5]).

O equilíbrio S_0 , o ponto em que a taxa de emigração e imigração são iguais, poderá diferir, ocasionalmente, devido a efeitos temporais, como no caso das doenças. A Figura 3.4, apresenta as curvas de imigração e emigração, como linhas retas, mas na realidade elas serão curvas mais complexas. Com isto estabelecido, esta representação simplificada apresenta uma boa aproximação ao processo de migração. Agora, considere uma probabilidade P_s

que o habitat contém exatamente S espécies. A probabilidade varia de tempo t a tempo $t + \Delta t$, seguindo a seguinte equação [5]:

$$P_s(t + \Delta t) = P_s(t)(1 - \lambda_s \Delta t - \mu_s \Delta t) + P_{s-1} \lambda_{s-1} \Delta t + P_{s+1} \mu_{s+1} \Delta t \quad (3.36)$$

onde λ_s e μ_s são as taxas de imigração e emigração quando existem S espécies no habitat. Para ter S espécies no tempo $(t + \Delta t)$, uma das seguintes condições tem de ser verdadeira.

1. Existiam S espécies no instante de tempo t e não ocorreu nenhuma imigração ou emigração entre t e $(t + \Delta t)$;
2. Existiam $(S - 1)$ espécies no instante de tempo t e uma espécie imigrou;
3. Existiam $(S + 1)$ espécies no instante de tempo t uma espécie emigrou.

Assume-se que o intervalo de tempo Δt é pequeno o suficiente, que a probabilidade de mais de uma imigração ou emigração possa ser ignorada. Utilizando um limite na equação 3.36 com o intervalo de tempo a tender para zero, surge a equação,

$$\dot{P}_s = \begin{cases} -(\lambda_s + \mu_s) P_s + \mu_{s+1} P_{s+1}, & S = 0 \\ -(\lambda_s + \mu_s) P_s + \lambda_{s-1} P_{s-1} + \mu_{s+1} P_{s+1}, & 1 < S \leq S_{max} - 1 \\ -\lambda_s + \mu_s P_s + \lambda_{s-1} P_{s-1}, & S = S_{max} \end{cases} \quad (3.37)$$

De forma a simplificar a notação as seguintes definições foram tomadas, $n = S_{max}$ e $P = [P_0 \dots P_n]^T$. Assim será possível rearranjar a equação anterior em forma matricial,

$$\dot{P} = AP \quad (3.38)$$

onde a matriz A é dada por,

$$A = \begin{bmatrix} -(\lambda_0 + \mu_0) & \mu_1 & 0 & \dots & 0 \\ \lambda_0 & -(\lambda_1 + \mu_1) & \mu_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \lambda_{n-2} & -(\lambda_{n-1} + \mu_{n-1}) & \mu_n \\ 0 & \dots & 0 & \lambda_{n-1} & -(\lambda_n + \mu_n) \end{bmatrix} \quad (3.39)$$

Considerando as linhas retas da Figura 3.4 temos,

$$\mu_k = E \frac{k}{n}; \quad \lambda_k = I \left(1 - \frac{k}{n}\right) \quad (3.40)$$

De uma forma geral, a equação 3.40 poderá ser apresentada de uma forma diferente se se

substituir o número de espécies pelo valor calculado na função de mérito f , da seguinte forma,

$$\lambda_i = I \frac{f_{\max} - f(H_i) + \varepsilon}{f_{\max} - f_{\min} + \varepsilon}; \quad \mu_i = E \frac{f(H_i) - f_{\min} + \varepsilon}{f_{\max} - f_{\min} + \varepsilon} \quad (3.41)$$

onde H_i representa a solução a ser estudada. Da mesma forma, o cálculo das taxas de migração também pode ser obtidos através da ordem do mérito da solução, após estas terem sido colocadas em ordem decrescente, através da fórmula,

$$\lambda_i = I \frac{i}{n}; \quad \mu_i = E \left(1 - \frac{i}{n} \right) \quad (3.42)$$

Agora, considerando o caso em que $E = I$, temos

$$\lambda_k + \mu_k = E \quad (3.43)$$

e a matriz A transforma-se em,

$$A = E \begin{bmatrix} -1 & \frac{1}{n} & 0 & \dots & 0 \\ \frac{n}{n} & -1 & \frac{2}{n} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \frac{2}{n} & -1 & \frac{n}{n} \\ 0 & \dots & 0 & \frac{1}{n} & -1 \end{bmatrix} = EA' \quad (3.44)$$

É possível notar que zero é um valor próprio da matriz A' com os correspondentes vetores próprios,

$$v = [v_1 \dots v_{n+1}] \quad (3.45)$$

$$v_i = \begin{cases} \frac{n!}{(n-1-i!)(i-1)!}, & (i = 1, \dots, i') \\ v_{n+2-i}, & i = i' + 1, \dots, n + 1 \end{cases} \quad (3.46)$$

O valor, em regime permanente, da probabilidade do número de cada espécie é dada por,

$$P(\infty) = \frac{v}{\sum_{i=1}^{n+1} v_i} \quad (3.47)$$

Ma [10] propôs a utilização de uma nova fórmula, para o cálculo da probabilidade de mutação, que dependia nas taxas de migração calculadas anteriormente, expressa da seguinte forma,

$$P_i = \begin{cases} P_0 = \frac{1}{1 + \sum_{k=1}^n \frac{\lambda_0 \lambda_1 \dots \lambda_{k-1}}{\mu_1 \mu_2 \dots \mu_k}}, & i = 0 \\ P_i = \frac{\lambda_0 \lambda_1 \dots \lambda_{i-1}}{\mu_1 \mu_2 \dots \mu_i \left(1 + \sum_{k=1}^n \frac{\lambda_0 \lambda_1 \dots \lambda_{k-1}}{\mu_1 \mu_2 \dots \mu_k} \right)}, & 1 \leq i \leq n \end{cases} \quad (3.48)$$

O primeiro método traz um problema à tona, o limite do valor real em programação, ou seja, ambos os casos conseguem ultrapassar o limite numérico imposto pela linguagem de programação. Isto, irá criar um limite no número de indivíduos que poderá ser utilizado, que por sua vez poderá trazer consequências como a falta de diversidade na população. De uma forma inicial, esta é uma das grandes limitações imposta pelo método. O segundo método, também implicará o mesmo problema, mas para um número de indivíduos, muito mais elevado que o recomendado. Simon [5], recomenda o uso de $E = I = 1$, $N = 50$ e $m_{\max} = 0,01$ como parâmetros iniciais para o BBO, embora, mais tarde, possa ser necessário ajustar estes parâmetros, dependendo do comportamento do algoritmo, para o problema específico.

3.3.2 Migração

Supondo que a população de soluções, de um problema, pode ser representada como um vetor de números inteiros. Cada um dos números inteiros é considerado um SIV (*suitability index variable*), cuja definição foi apresentada no capítulo 2. O HSI (*habitat suitability index*) é análogo ao fitness, utilizado em outros processos de otimização baseados em populações. Soluções com HSI elevado representam habitats com várias espécies e soluções com HSI baixo representam habitats com poucas soluções. É possível assumir que cada solução tem uma curva de migração idêntica (com $E = I$), mas o valor S representado pela solução depende apenas no seu HSI.

As taxas de emigração e imigração de cada solução são usadas para, probabilisticamente, partilhar informação entre os habitats. A probabilidade P_{mod} , irá modificar cada solução baseando-se em outras soluções, se uma dada solução for selecionada para ser modificada. Assim, usa-se a sua taxa de imigração λ para decidir se se irá modificar cada SIV nessa solução. Se um SIV, numa dada solução S_i , for selecionado para ser modificado, então a taxa de emigração μ , das outras soluções, irão decidir qual das soluções irão migrar um SIV, selecionado aleatoriamente, para a solução S_i .

Este processo assemelha-se bastante ao processo de reprodução de outros algoritmos genéticos (AG), onde muitos parentes conseguem contribuir na criação de um descendente. Mas, o BBO difere num aspeto muito importante, no AG a reprodução através de combinação é usada para a criação de novas soluções, enquanto no BBO a migração é usada para modificar soluções existentes. Assim, pode-se dizer que, ao contrário da reprodução, a migração dos BBO é um processo adaptativo. É importante notar que o elitismo ainda é requerido neste tipo de otimização, para garantir que as melhores soluções não sejam corrompidas pela imigração.

Em termos algorítmicos o operador de migração presente no BBO, pode ser descrito da seguinte forma (Algoritmo 2), onde D representa a dimensão do problema. O processo apresentado na linha três, poderá ser realizado através de uma roleta, um processo, já utilizado em outros algoritmos evolucionários.

Algoritmo 2: Pseudocódigo para a migração [59].

```

for  $d = 1, D$  do
  if  $\text{rand} < \lambda_i$  then
    Selecionar outro habitat ( $H_j$ ) da população com uma probabilidade  $\propto \mu_j$ 
    Definir  $H_i(d) = H_j(d)$ 
  end if
end for

```

De um modo geral, a utilização de modelos lineares para descrever as curvas de migração é suficiente. Sendo, esta operação um dos pontos principais do BBO, inicialmente, houve bastantes variações neste operador (de migração), quer seja para melhorar a sua habilidade de procura ou para estender o BBO para outras aplicações. Primeiramente, Ma e Simon [60] apresentaram um novo algoritmo híbrido, chamado Blended BBO (B-BBO), que retira algumas ideias dos operadores de crossover, dos algoritmos genéticos, criando um operador de migração, que mistura cada componente de uma solução imigrante (H_i) com o componente corresponde da solução emigrante (H_j), da seguinte forma,

$$H'_i(d) = \alpha H_i(d) + (1 - \alpha) H_j(d) \quad (3.49)$$

onde α é um número real entre $[0, 1]$, podendo ser um valor predeterminado ou um valor aleatório.

Com o operador de migração base, apenas é possível melhorar as soluções más, utilizando características vindas de soluções boas. Mas, o operador de migração de mistura (*blended* ou *blending*), não só permite isso, como também consegue evitar, mesmo que parcialmente, a degradação de boas soluções, já que, estas mantêm uma parte das suas características originais. Este operador também é capaz de produzir novos componentes, de ambas as soluções, promovendo assim a diversidade.

Uma outra solução para o operador de migração, será a utilização de modelos não lineares para descrever as curvas de migração. Esta configuração permitirá uma simulação, mais próxima da realidade, da biogeografia. Um modelo não linear típico, é o modelo quadrático [10], onde as curvas de migração são funções quadráticas convexas do número de espécies, tal como está apresentado na Figura 3.5. As equações podem ser escritas da seguinte forma,

$$\lambda_s = I \left(1 - \frac{s}{n}\right)^2; \quad \mu_s = E \left(\frac{s}{n}\right)^2 \quad (3.50)$$

onde n representa a população máxima.

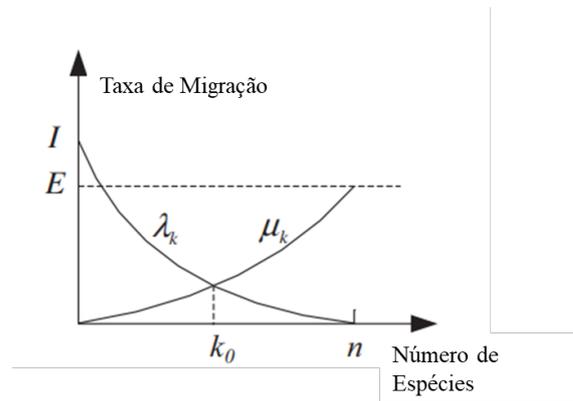


Figura 3.5: Representação gráfica do modelo de migração quadrático ([59]).

Este modelo é baseado, numa teoria testada experimentalmente de biogeografia de ilhas [8], que diz que quando o habitat tem um número de espécies reduzido, a taxa de imigração decresce rapidamente do máximo enquanto a taxa de emigração aumenta lentamente do zero. Outro modelo não linear popular é o modelo sinusoidal [10], em que as equações são,

$$\lambda_s = \frac{I}{2} \left(\cos \left(\frac{s\pi}{n} \right) + 1 \right); \quad \mu_s = \frac{E}{2} \left(-\cos \left(\frac{s\pi}{n} \right) + 1 \right) \quad (3.51)$$

As curvas sinusoidais descrevem melhor o comportamento real, já que, têm em conta fatores como relações presa/predador, mobilidade da espécie e a sua evolução. Quando o habitat tem um número de espécies médio, ambas as taxas de migração, variam rapidamente do valor de equilíbrio, o que fará com que o habitat demore mais tempo a estabilizar no ponto de equilíbrio k_0 . A Figura 3.6 representa as curvas de migração para o caso sinusoidal.

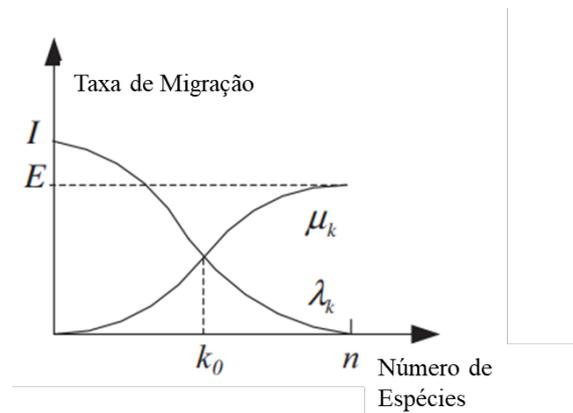


Figura 3.6: Representação gráfica do modelo de migração sinusoidal ([59]).

Ma [10], também recomenda a utilização de um modelo trapezoidal, descrito da seguinte

forma,

$$\lambda_s = \begin{cases} I, & s \leq i' \\ 2I \left(1 - \frac{s}{n}\right), & i' < s \leq n \end{cases} \quad \mu_s = \begin{cases} \frac{2 \cdot E}{n}, & s \leq i' \\ E, & i' < s \leq n \end{cases} \quad (3.52)$$

onde i' é o número inteiro menor mais próximo de $(n + 1)/2$. O gráfico resultante é apresentado, na Figura 3.7.

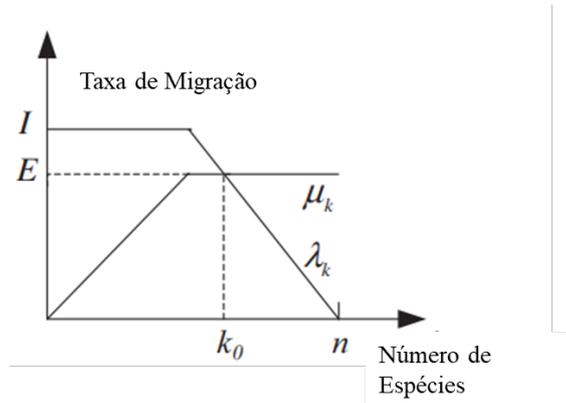


Figura 3.7: Representação gráfica do modelo de migração trapezoidal ([10]).

3.3.3 Mutaç o

Eventos catacl smicos podem mudar drasticamente o HSI de um habitat natural, o que poder  fazer com que o n mero de esp cies difere do ponto de equil brio k_0 . Para representar este aspeto, no BBO,   usado o conceito de muta o, como uma muta o dos SIV. Sendo estes utilizados para determinar a taxa de muta o.

Cada elemento da popula o tem uma probabilidade associada, que indica a probabilidade de se esperar *a priori* que exista como solu o para o problema em causa. Solu es com HSI muito baixo ou muito elevado ser o igualmente improv veis de serem mudadas e solu es com HSI interm dio ser o relativamente prov veis. Se uma solu o S tem uma probabilidade P_s baixa, ent o a probabilidade de esta ser mudada   alta. Pelo lado oposto, uma solu o com uma probabilidade elevada ter  menos oportunidades para ser mudada. Isto pode ser implementado, usando a seguinte f rmula,

$$m_i = m_{\max} \left(\frac{1 - P_i}{P_{\max}} \right) \quad (3.53)$$

onde m_{\max}   um par metro definido pelo pr prio utilizador, que representa a taxa de muta o m xima.

Este m todo permite que solu es com HSI baixo consigam ser modificadas, abrindo assim a possibilidade destas melhorarem. E com o uso de elitismo, que garante que as melhores solu es se mantenham no espa o de estudo, a muta o, que ir  ser designada de muta o normal neste trabalho, permite que haja uma certeza que o programa n o encalhe num ponto  timo local. Algoritmicamente este processo pode ser descrito da seguinte forma

(Algoritmo 3), onde d representa a variável a ser estudada, D a dimensão do problema, H_i a solução em análise e u_d e l_d representam os limites superior e inferior do problema, respetivamente.

Recentemente, o processo de mutação simplificado, onde apenas habitats com HSI baixo são mutados, removendo a necessidade de calcular a probabilidade da contagem da espécie e a taxa de mutação para cada espécie, tem sofrido um maior destaque.

Algoritmo 3: Algoritmo para a mutação normal do BBO.

```

for  $d = 1, D$  do
  if  $\text{rand} < m_i$  then
    Definir  $H_i(d) = l_d + \text{rand} \cdot (u_d - l_d)$ 
  end if
end for

```

Tal como, no caso do operador de migração, vários outros operadores de mutação foram propostos. De realce, o operador proposto em [61] designado de operador de mutação taxonómica. Neste operador, as soluções de HSI elevado são excluídos de serem mutados e o tipo de mutação difere dependendo se o HSI da solução é baixo ou médio. Para soluções de HSI médio, como a sua solução é um pouco útil, será utilizado um processo de mutação diferencial, utilizando uma taxa de crossover P_C para julgar se a solução é cruzada ou não. Para soluções de HSI baixo, será utilizado mutação aleatória, similar à mutação normal do BBO, com um intervalo, auto adaptativo, restrito pela população atual. Ambos os processos, estão representados nos Algoritmos 4 e 5, algoritmicamente, para facilitar a sua visualização.

Algoritmo 4: Pseudocódigo para a mutação diferencial, da mutação taxonómica [61].

```

for  $i = 1$  to  $N_{pop}$  do
  Selecionar 3 soluções ( $H_1, H_2, H_3$ ) e uma dimensão ( $d_r$ ) aleatoriamente
  for  $d=1$  to  $nv$  do
    if  $\text{rand} < m_i$  then
      if  $\text{rand} \leq P_C$  or  $d == d_r$  then
         $H_i(d) = H_1(d) + F \cdot (H_2(d) - H_3(d))$ 
      else
         $H_i(d) = H_{\min}(d) + \text{rand} \cdot (H_{\max}(d) - H_{\min}(d))$ 
      end if
    else
      if  $\text{rand} \leq P_C$  or  $d == d_r$  then
         $H_i(d) = H_1(d)$ 
      else
         $H_i(d) = H_{best}(d) + F \cdot (H_2(d) - H_3(d))$ 
      end if
    end if
  end for
end for

```

Algoritmo 5: Pseudocódigo para a mutação aleatória, da mutação taxonómica [61].

```

for  $i=1$  to  $N_{pop}$  do
  for  $d = 1$  to  $nv$  do
    if  $H_i(d)$  é selecionado com a probabilidade  $m_i$  then
       $H_i(d) = H_{\min}(d) + \text{rand} \cdot (H_{\max}(d) - H_{\min}(d))$ 
    end if
  end for
end for

```

3.3.4 Definições

Nesta secção serão apresentadas algumas das definições necessárias para uma formalização do algoritmo BBO. Estas definições foram tiradas do trabalho original de Dan Simon [5], onde este apresentou o algoritmo BBO pela primeira vez.

1. Um habitat $H \in \text{SIV}^m$ é um vetor de m números inteiros que representa uma possível solução do problema
2. O $\text{SIV} \in C$ é o número inteiro que é permitido num habitat. $C \subset Z$ é o conjunto de todos os números inteiros que são permitidos num habitat.
3. O HSI: $H \rightarrow R$ é uma medida de mérito de uma solução representada pelo habitat. O HSI é, normalmente, designado de mérito, na maioria, dos algoritmos baseados em populações.
4. Um ecossistema H^n é um grupo de n habitats. O tamanho do ecossistema poderá ser constante, como é o caso do tamanho da população em alguns algoritmos genéticos.
5. A taxa de imigração $\lambda : R \rightarrow R$ é uma função monotonicamente não crescente do HSI. λ_i é proporcional à probabilidade de um SIV de habitats vizinhos migrar para o habitat H_i .
6. A taxa de emigração $\mu : R \rightarrow R$ é uma função monotonicamente não crescente do HSI. μ_i é proporcional à probabilidade de um SIV do habitat H_i migrar para um habitat vizinho.

Na prática é assumido que ambas as taxas são lineares com os mesmos valores máximos. No entanto, isto serve só para simplificação matemática e conveniência.

7. Modificação de habitats $\Omega(\lambda, \mu) : H^n \rightarrow H$ é um operador que ajusta o habitat H baseando-se no ecossistema H^n . A probabilidade de H ser modificado é proporcional à sua taxa de imigração e a probabilidade de que esta mudança venha de um habitat H_j é proporcional à taxa de imigração μ_j .

É possível definir $\lambda = 0$ para os p melhores habitats, onde p é um parâmetro definido pelo utilizador, para representar uma forma de elitismo no habitat.

8. Mutaç o $M(\lambda, \mu) : H \rightarrow H$ é um operador que modifica os SIV de um habitat, aleatoriamente, baseado na probabilidade de exist ncia desse habitat.
9. A funç o de transiç o de um ecossistema $\Psi = (m, n, \lambda, \mu, \Omega, M) : H^n \rightarrow H^n$ é um s xtuplo que modifica o ecossistema de uma iteraç o para a pr xima. Esta funç o pode ser escrita da seguinte forma,

$$\Psi = \lambda^n \circ \mu^n \circ \Omega^n \circ \text{HSI}^n \circ M^n \circ \text{HSI}^n \quad (3.54)$$

10. Um algoritmo BBO, $\text{BBO} = (I, \Psi, T)$ é um terno ordenado que prop e uma soluç o a um problema de otimizaç o. $I : \emptyset \rightarrow \{H^n, \text{HSI}^n\}$ é uma funç o que cria um ecossistema inicial de habitats e que computa o HSI, correspondente a cada um. $T : H^n \rightarrow \{\text{true}, \text{false}\}$ é o crit rio de t rmino.

Ambos conceitos já são utilizados em outros algoritmos de otimização baseados em população. A função I pode ser implementada através da geração de números aleatórios ou através de outros métodos dependentes do problema em questão. T tanto se pode basear no número de iterações, como no HSI dos melhores habitats.

3.3.5 Adaptação do BBO para otimização com restrições

Conforme a forma geral de otimização restrita, a função penalidade pode ser descrita como,

$$\phi(\mathbf{x}) = \sum_{i=1}^p M_i |\min(0, g_i(\mathbf{x}))|^\alpha \quad (3.55)$$

onde p é o número de restrições, g_i é a restrição, M_i são constantes positivas altas e α é um coeficiente maior ou igual a um.

Geralmente, é possível usar uma constante M para todas as restrições e $\alpha = 1$. Baseado na função penalidade, a função objetivo original f pode ser transformada numa função penalizada,

$$f'(\mathbf{x}) = f(\mathbf{x}) + \phi(\mathbf{x}) \quad (3.56)$$

de notar que esta fórmula poderá ser modificada, caso o problema peça para a maximização da função objetivo.

O método da função de penalidade é simples, mas é sensível aos valores dos fatores de penalidade M_i e α . Para contrariar este facto, Boussaïd et al. [62], propôs três variantes do algoritmo BBO, chamadas CBBO, CBBO-DM e CBBO-PM. Estes algoritmos usam o método de penalidade em conjunto com um procedimento de classificação estocástico para balançar dinamicamente a dominância das soluções viáveis e não viáveis, ou seja, cada solução é classificada conforme a sua aptidão e a sua violação à restrição. A diferença principal entre os três algoritmos vem da sua filosofia na atualização de soluções.

O CBBO utiliza a filosofia base considerada no BBO. O CBBO-DM substitui a mutação original do BBO pela mutação do método DE (Evolução diferencial) para gerar novas soluções e aplica a migração nas soluções mutadas. O CBBO-PM substitui a mutação do BBO com um novo operador de mutação baseado no método pivot, que gera uma nova solução mutada ao combinar a melhor solução atual com outra solução selecionada aleatoriamente. Entre cada solução original H_i e a sua nova solução $H_{i'}$, o algoritmo seleciona a melhor para a próxima geração considerando três casos:

- Se ambas as soluções são viáveis, escolhe-se a com melhor aptidão;
- Se nenhuma das soluções forem viáveis, seleciona-se a que terá um valor de penalidade inferior
- Se apenas uma das soluções é viável, escolhe-se a solução viável.

Bi [63] propôs um algoritmo BBO que adota o método ε -constraint, chamado de ε -BBO, para lidar com as restrições o problema de otimização. Tal como, no BBO base o ε -BBO

também utiliza o HSI para o cálculo das taxas de migração, mas, ao contrário do BBO, este considera a influência da violação da restrição na classificação do BBO, de forma a beneficiar as soluções viáveis. A estratégia dinâmica do ε -BBO utiliza o vetor diferencial no operador de migração, para produção uma perturbação desviante da seguinte forma,

$$H_i(d) = H_j(d) + (r_{\min} + \lambda_i \cdot r \cdot (r_{\max} - r_{\min})) \cdot (H_{r_1}(d) - H_{r_2}(d)) \quad (3.57)$$

onde $H_i(d)$ e $H_j(d)$ representam a dimensão da solução atual e solução emigrante, respectivamente, r_1 e r_2 são dois números inteiros aleatórios entre $[1, N]$ e o valor de r aumenta com as iterações t dentro do intervalo $[r_{\min}, r_{\max}]$ da seguinte forma,

$$r^{(t)} = (1 - \beta) \cdot r^{(t)} + \beta \quad (3.58)$$

Desta forma, a perturbação será mínima no início da procura, de forma a preservar as características iniciais de soluções promissoras. Enquanto, que na fase final, a perturbação aumentará de forma a melhorar a diversidade da população. O ε -BBO também utiliza um operador de mutação diferente, que explora a vizinhança próxima da solução ótima local mais eficientemente para melhorar a precisão da solução. Inicialmente gera uma sequência caótica de acordo com a equação 3.59 e depois segue o processo de mutação de acordo com a equação 3.60.

$$z_{t+1}(d) = \begin{cases} 4\mu z_t(d) (0,5 - z_t(d)), & 0 \leq z_t(d) < 0,5 \\ 4\mu (z_t(d) - 0,5) (1 - z_t(d)), & 0,5 \leq z_t(d) < 1 \end{cases} \quad (3.59)$$

$$H_i(d) = (2 - 2z_i(d)) \cdot H_{i(d)} + (2z_i(d) - 1) \times H_{\text{best}}(d) \quad (3.60)$$

3.3.6 Adaptação para otimização multiobjetivo

De forma a utilizar o BBO, que foi inicialmente desenvolvido para problemas com um objetivo, em problemas multiobjetivo é necessário modificar o cálculo do HSI, para que este tome em consideração todas as funções objetivo. Ma et al. [64] introduziram o BBMO, em que a ideia principal é correr várias sub-rotinas do BBO em paralelo, cada uma a tentar otimizar o seu objetivo. Durante o processo de otimização em paralelo, as taxas de migração são calculadas da seguinte forma,

$$\lambda_i = I \left(1 - \frac{k(i)}{n} \right); \mu_i = E \frac{k(i)}{n} \quad (3.61)$$

onde $k(i)$ representa a classificação da solução em termos de aptidão.

Bi et al. [65] propuseram o MOBBO onde o HSI de cada solução é calculado de acordo com a sua eficiência de Pareto em relação com as outras soluções da população. Seja $P = \{H_1, H_2, \dots, H_N\}$ a população com N soluções, o valor de aptidão de cada H_i é calculado da seguinte forma

$$f(H_i) = 1 - \frac{1}{N} \sum_{H_j \in P \wedge H_j H_i} |\{H_j \in P | H_j H_i\}| \quad (3.62)$$

O MOBBO utiliza um método auto-adaptável para o cálculo das taxas de migração para cada solução da seguinte forma,

$$\lambda_i = \left(\frac{f_{\max} - f(H_i)}{f_{\max} - f_{\min}} \right)^{1/N^\alpha} ; \mu_i = 1 - \lambda_i \quad (3.63)$$

A cada geração, quando o número de soluções não dominadas é superior à população N o MOBBO utiliza uma abordagem de seleção competitiva, onde este primeiro calcula a distância Euclidiana entre cada par de soluções e continua a selecionar um par de soluções cuja distância é a mais pequena e remove o mais lotado até que o número chegue a N . Este processo permite manter a diversidade, algo importante para a otimização multiobjectivo.

Capítulo 4

Metodologia

Nesta secção será apresentado e descrito todo o processo utilizado para a implementação do algoritmo BBO e da sua hibridização. Inicialmente, será apresentado o algoritmo base do BBO, serão mencionados os modelos de migração e os operadores de mutação utilizados. Mais tarde, serão apresentadas as modificações feitas ao BBO, antes da hibridização. E, finalmente, serão mencionados os novos processos implementados no processo de hibridização do método. Esta informação será complementada com várias figuras, que tentarão representar o algoritmo utilizado, no decorrer desta dissertação.

4.1 Implementação do BBO

A linguagem de programação usada para esta tese, foi o FORTRAN 90. No algoritmo base, a população inicial foi gerada, de forma aleatória, e no domínio real. De uma forma geral, o método pode ser representado de acordo com [4.1](#).

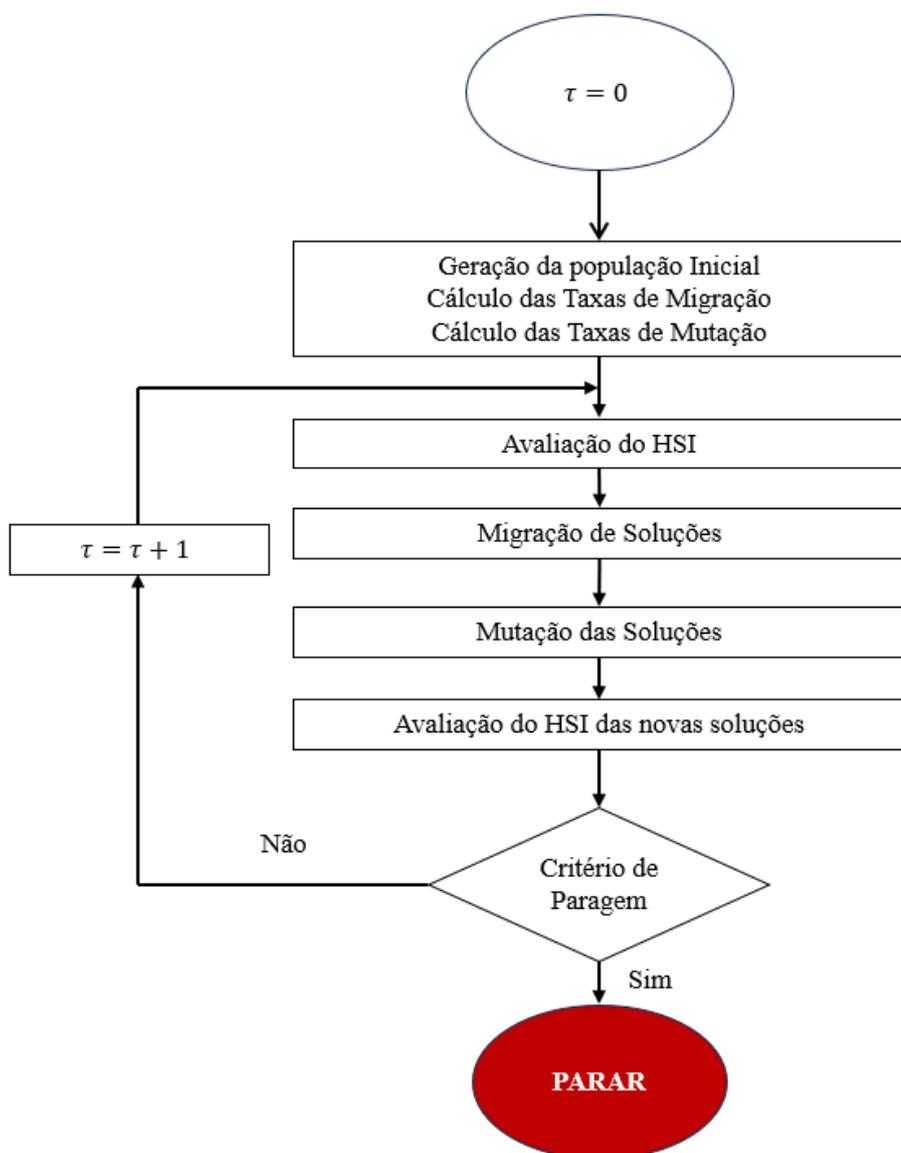


Figura 4.1: Diagrama de fluxo do BBO implementado.

Foram utilizados os quatro modelos de migração (linear, trapezoidal, quadrático e sinusoidal) apresentados na secção 3.3.2. Os operadores de mutação usados foram, a mutação referida pelo primeiro autor a apresentar este modelo [5], designada de mutação normal neste trabalho, e a mutação taxonómica de Hao [61]. O conceito de *blending* (equação 3.49), também foi implementado neste algoritmo. O cálculo das taxas de migração e de mutação, seguiu o mesmo princípio de Ma [10]. Finalmente, o algoritmo do BBO é definido da seguinte forma (Algoritmo 6).

Algoritmo 6: Pseudocódigo do BBO base.

```

/*Iniciação dos parâmetros*/
Cálculo da população inicial
Cálculo dos valores de HSI (mérito)
while critério de paragem não é satisfeito do
    Calcular os valores das taxas de migração de acordo com o modelo escolhido
    /*Migração*/
    Selecionar um  $H_i$  de acordo com a sua taxa de imigração
    if  $H_i$  é selecionado then
        Selecionar um  $H_j$  de acordo com a sua taxa de imigração
        Substituir o  $H_j$  com o  $H_i$ 
    end if
    /*Mutaçãõ*/
    Selecionar um  $H_i$  com a probabilidade baseada na taxa de mutaçãõ
    if  $H_i$  é selecionado then
        Substituir  $H_i$  com uma soluçãõ gerada aleatoriamente
    end if
    Cálculo dos valores de HSI (mérito)
end while

```

Para aumentar a diversidade populacional foi ainda implementado uma geração aleatória de soluções, a cada geração. Estas soluções serão depois inseridas na população, substituindo as piores soluções (últimos 10%). Também foi implementado, para o mesmo propósito, um ciclo, no operador de mutação. Mas, ao contrário do processo anterior, estas soluções só são aceites na nova população se tiverem um mérito superior à solução original.

Numa iteração inicial foram utilizados 50 indivíduos, como a população inicial, que se manterá constante por todas as gerações. Foi definido um valor de emigração (E) e imigração (I) igual a 1, um valor máximo de mutação (m_{max}) de 0.01, um número de ciclos de mutação inicial de 3, que irá aumentar por um a cada 100 gerações, até um valor máximo de 20.

4.2 Implementação da Hibridização

Nesta secção serão apresentados os operadores utilizados na hibridização e os parâmetros usados na sua implementação.

4.2.1 Aplicação de operadores de crossover

Este operador será utilizado para melhorar a troca de informação genética, na população inteira. Com a adição deste operador, espera-se que exista uma perda de informação genética menor, algo que o BBO tenta abordar parcialmente com o conceito de *blending*.

Mesmo existindo vários exemplos de crossover no domínio real, é recomendado o uso do domínio binário, para a implementação dos operadores, devido à sua simplicidade, representação natural e pelo facto dos operadores criados para este tipo de codificação serem mais eficientes. Para isso, foi preciso desenvolver um decodificador binário e um codificador real, para facilitar a transição entre os operadores no domínio real e os operadores no domínio binário. Os operadores, apresentados na secção 3.1.3, foram implementados, sobre duas estratégias diferentes.

A **primeira estratégia**, utiliza os valores das taxas de migração como a probabilidade de crossover e esta é usada para a escolha do primeiro progenitor. O segundo progenitor, é escolhido a partir da lista de melhores soluções das gerações anteriores. Os filhos criados, serão então avaliados e colocados na nova geração segundo a sua posição na classificação de mérito geral.

A **segunda estratégia** [66], utiliza o conceito de elitismo. A população atual é dividida em dois grupos, as soluções superiores (elite) e as inferiores. É escolhido um progenitor de cada um dos grupos, através do método roleta (equação 4.1), para a criação de filhos. Após isso, os filhos criados são todos colocados na população, substituindo as soluções que não estejam na elite.

$$P(\mathbf{x}_j^k) = \frac{f(\mathbf{x}_j^k)}{\sum_{j=1}^{N_{pop}} f(\mathbf{x}_j^k)} \quad (4.1)$$

onde $f(\mathbf{x}_j^k)$ representa o valor do mérito da solução e N_{pop} o tamanho da população.

Os dois métodos, apresentam ideias diferentes, não só para a seleção dos pais, mas também para a criação da nova população. Uma tenta acrescentar diversidade, na nova geração, respeitando as elites, existentes. Enquanto a outra, beneficia sempre as melhores soluções, quer estas tenham sido recentemente criadas ou quer estas já existam desde a primeira geração. Inicialmente, o operador crossover será utilizado a cada 100 gerações, sendo introduzido antes da migração e mutação.

4.2.2 Aplicação do otimizador local (Hook-Jeeves)

Com a adição de um otimizador local, é esperado que o método melhore a sua pesquisa local, algo que já foi reconhecido como um ponto fraco do BBO. Este operador irá tentar aumentar a velocidade, com que este algoritmo chega ao ponto ótimo, ao explorar a vizinhança das soluções obtidas pela pesquisa global do método. Em adição, à enorme diversidade garantida pelos outros operadores já acrescentados, espera-se que este método se destaque na pesquisa global e local.

O otimizador local utilizado, foi inspirado no método Hooke-Jeeves, um método comum para este tipo de aplicações, que permite a pesquisa eficiente em espaços de grande dimensão, mesmo quando a função objetivo é complexa e não diferenciável. Este foi transformado e os seus parâmetros foram adaptados de forma a conseguir melhorar o seu desempenho de acordo com o algoritmo e os restantes operadores. O Algoritmo 7 apresenta o pseudocódigo para o método de pesquisa local.

Algoritmo 7: Pseudocódigo do método de pesquisa local de Hooke-Jeeves

```

/* Iniciação dos parâmetros */
Calcular tamanho do passo  $h$ 
for  $i = 1$  to  $N_{pop}$  do
  Seleção do ponto a ser explorado,  $H_i$ 
  Realizar a primeira iteração
  /* Avaliar as soluções geradas,  $H_{new}$  */
  if  $H_{new}$  melhor que  $H_i$  then
     $H_i \leftarrow H_{new}$ 
  end if
  Guardar a informação da solução,  $H_{old}$ 
  /* Loop de procura local */
  while critério de paragem não é atingido do
    Calcular o novo valor de  $h$ 
    Calcular a nova solução,  $H_{new}$ 
    /* Avaliar nova solução */
    if  $H_{new}$  melhor que  $H_{old}$  then
       $H_{old} \leftarrow H_{new}$ 
      if  $H_{new}$  melhor que  $H_i$  then
         $H_i \leftarrow H_{new}$ 
      end if
    end if
  end while
end for

```

O primeiro passo, h^1 , foi calculado da seguinte forma,

$$h^1(j) = H_1(j) - H_{N_{pop}}(j) \quad (4.2)$$

onde j representa a variável, H_1 a melhor solução da população atual e $H_{N_{pop}}$ a pior solução atual. A redução do passo, é dado da seguinte fórmula,

$$h^i(j) = (1/i) \cdot h^1(j) \quad (4.3)$$

onde i representa o número de ciclos que a solução passou no processo de pesquisa local.

É importante referir que o passo será adicionado e subtraído à solução, sendo avaliado para as duas hipóteses. O ciclo tem como critério de paragem o número de ciclos, definido como 100. Relativamente à escolha de soluções, foi utilizado uma probabilidade igual para todas as soluções, sendo estas escolhidas aleatoriamente, exceto para as duas melhores, que serão sempre selecionadas. O valor para a probabilidade foi definido com 0,1. Este processo ocorrerá para todas as gerações após a realização da migração, mutação e crossover. O pseudocódigo do método hibridizado está apresentado no Algoritmo 8.

Finalmente, o diagrama de fluxo que representa o algoritmo hibridizado está representado na Figura 4.2.

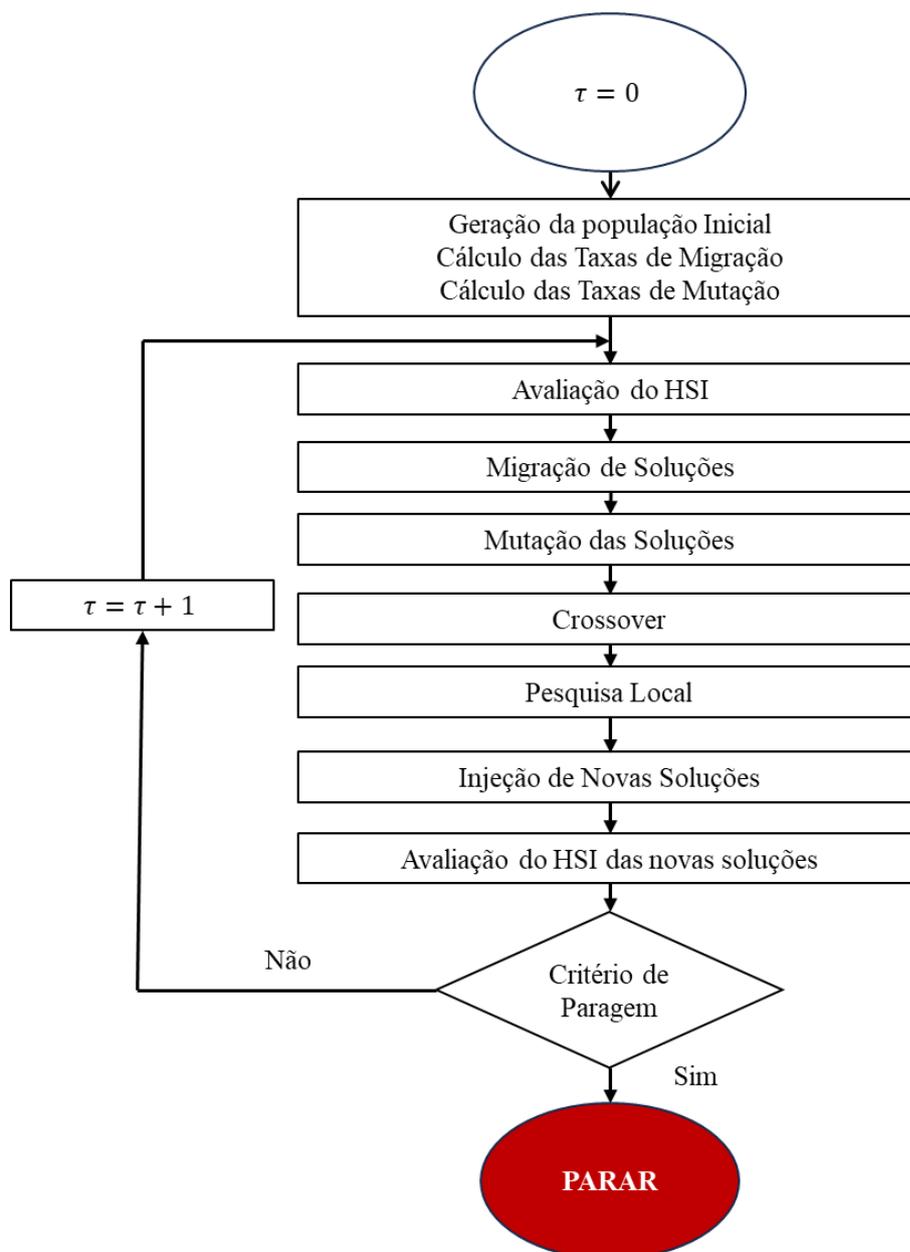


Figura 4.2: Diagrama de fluxo completo do método hibridizado.

Algoritmo 8: Pseudocódigo do BBO hibridizado com pesquisa local.

```

/* Iniciação dos parâmetros */
Criação da população inicial
Calcular os valores das taxas de migração e mutação de acordo com o modelo escolhido
Cálculo dos valores de HSI (mérito)
while critério de paragem não é satisfeito do
  /* Crossover */
  if RESTO(iteração, 100) == 0 then
    Cálculo da probabilidade de crossover de acordo com a estratégia escolhida
    Realizar crossover
  end if
  /* Migração */
  if  $H_i$  é selecionado then
    Selecionar um  $H_j$  de acordo com a sua taxa de emigração
    Substituir o  $H_j$  pelo  $H_i$ 
  end if
  /* Mutação */
  contar  $\leftarrow$  0
  while contar < n do
    Selecionar um  $H_i$  com a probabilidade baseada na taxa de mutação
    if  $H_i$  é selecionado then
      Criar a nova solução,  $H_n$ 
      Avaliar a nova solução criada
      if  $H_n$  tem um mérito melhor que  $H_i$  then
        Substituir  $H_i$  com a nova solução
      else
        Manter  $H_i$ 
      end if
    end if
    contar  $\leftarrow$  contar + 1
  end while
  Cálculo dos valores de HSI (mérito)
  Classificação das soluções
  /* Pesquisa Local */
  Seguir pseudocódigo método de pesquisa local de Hooke-Jeeves, apresentado anteriormente
  Classificação das soluções
  /* Injeção de novas soluções */
  for  $i = 0.9 * N_{pop}$  to  $N_{pop}$  do
    Criar as novas soluções
    Substituir as piores soluções
  end for
  Cálculo dos valores de HSI (mérito)
end while

```

Capítulo 5

Resultados e Discussão

Nesta secção serão apresentados os exemplos de estudo, os seus resultados e a sua análise.

5.1 Exemplo de Teste: Função Quadrática

Este problema foi concebido, num estágio inicial, para testar a implementação do algoritmo e a influência do *blending*, no método BBO atualizado (com a injeção de soluções e o ciclo de mutação, secção 4.1). A equação de estudo é definida da seguinte forma,

$$f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 2)^2 - 1 \quad (5.1)$$

Sendo uma equação tão simples, é possível resolvê-la analiticamente, utilizando o método seguinte,

$$\nabla f(x_1, x_2) = 0 \quad (5.2)$$

onde,

$$\nabla f(x_1, x_2) = [2x_1 - 2, \quad 2x_2 - 4] \quad (5.3)$$

Resolvendo a equação 5.2, é possível chegar-se ao ponto óptimo $(x_1^*, x_2^*) = (1, 2)$ cujo valor da função objetivo será $f^*(x_1, x_2) = -1$. Este problema requer a minimização da função. Para este caso, a função de mérito será igual à expressão do problema.

5.1.1 Resultados

Este exemplo serviu, inicialmente, como uma função de teste para o algoritmo base do BBO e dos vários modelos de migração implementados, sendo estes modelos estudados com e sem o efeito *blending*, apresentado no capítulo 3. Aqui serão apresentados os resultados obtidos nesses testes. Sendo a solução ótima conhecida e tratando-se de uma função simples, o código foi pedido para parar assim que este chegasse ou estivesse muito próximo ($tol = 10^{-8}$) do valor óptimo ou quando fosse atingido o valor máximo de gerações, 10000 neste caso. Os resultados estão apresentados na tabela 5.1.

Tabela 5.1: Resultados para o primeiro exemplo.

Modelo	<i>Blending</i>	Gerações	x_1	x_2	Objetivo
Linear	Não	1814	1	2	-1
Linear	Sim	93	1,000023	2,000143	-1
Trapezoidal	Não	10000	1,052513	2,089691	-0,9892
Trapezoidal	Sim	7697	1	2	-1
Quadrático	Não	10000	1,052513	2,089691	-0,9892
Quadrático	Sim	121	0,999885	1,999948	-1
Sinusoidal	Não	10000	1,052513	2,089691	-0,9892
Sinusoidal	Sim	141	0,999879	2,000101	-1

Como é possível ver, a adição do *blending* traz imensas vantagens ao processo de migração, resultando num algoritmo muito mais rápido. Todos os modelos com *blending* conseguiram chegar à solução ótima antes do limite de gerações máximas. Em termos do método, o linear, provou ser o mais rápido, para este caso, isto deve-se, provavelmente, ao facto de que a função a ser estudada é bastante simples. O que poderá resultar numa boa relação com este modelo, que também é o mais simples dos modelos apresentados.

5.2 Exemplo de Teste Polinomial

Este exemplo foi retirado de [67] e foi utilizado para o teste da implementação da codificação binária, e dos processos de codificação e decodificação. Também servirá para demonstrar a influência da codificação binária de soluções e a sua interação com os restantes operadores do BBO, que utilizam, maioritariamente, codificação real. A fórmula dada para este problema foi a seguinte:

$$f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2 \quad (5.4)$$

Mais uma vez sendo uma equação simples o método dado em 5.2, também poderá ser utilizado para prever o resultado esperado desta equação. Este problema requer a minimização da função. Para este caso, a função objetivo será igual à expressão do problema. Assim, o ponto ótimo é definido como,

$$f^*(-1, 1.5) = -1.25 \quad (5.5)$$

5.2.1 Resultados

Este exemplo serviu, inicialmente, como uma função de teste para o algoritmo base do BBO. Ele foi utilizado para continuar o estudo dos modelos de migração, dos operadores de mutação e dos domínios de codificação da população. Sendo a solução ótima conhecida e tratando-se de uma função simples, o código foi pedido para parar assim que este chegasse ou estivesse muito próximo ($tol = 10^{-5}$) ao valor ótimo ou quando fosse atingido o valor

máximo de gerações, 10000 neste caso.

Os resultados obtidos, relativamente aos modelos de migração, estão apresentados na tabela 5.2.

Tabela 5.2: Resultados do estudo dos modelos de migração para o segundo exemplo.

Modelo	<i>Blending</i>	Gerações	x_1	x_2	Objetivo
Linear	Não	10000	-0,94557	1,448334	-1,24703
Linear	Sim	10000	-1,00165	1,505161	-1,24999
Trapezoidal	Não	10000	-0,94557	1,448334	-1,24703
Trapezoidal	Sim	10000	-0,89611	1,450836	-1,23621
Quadrático	Não	10000	-0,94557	1,448334	-1,24703
Quadrático	Sim	10000	-0,96624	1,43272	-1,24774
Sinusoidal	Não	10000	-0,91311	1,515244	-1,23202
Sinusoidal	Sim	132	-1,00237	1,501187	-1,24999

A codificação real atingiu o valor pretendido, apenas num caso, com o modelo sinusoidal e *blending*. Mas, conseguiu, de forma geral, criar boas soluções, estando a maioria próxima da solução ótima. Os resultados obtidos, relativamente à codificação binária, estão apresentados na Tabela 5.3.

Tabela 5.3: Resultados obtidos através de codificação binária de soluções para o segundo exemplo.

Modelo	<i>Blending</i>	Gerações	x_1	x_2	Objetivo
Linear	Não	10000	-0,94557	1,448334	-1,24703
Linear	Sim	121	-1,00195	1,501623	-1,25
Trapezoidal	Não	10000	-0,89611	1,450836	-1,23621
Trapezoidal	Sim	10000	-0,91311	1,515244	-1,23202
Quadrático	Não	10000	-0,94557	1,448334	-1,24703
Quadrático	Sim	94	-1,00007	1,499306	-1,25
Sinusoidal	Não	10000	-0,94557	1,448334	-1,24703
Sinusoidal	Sim	110	-0,99713	1,496622	-1,24999

Como é possível notar, a codificação binária conseguiu atingir a solução ótima, antes do limite de gerações, três vezes, para os modelos sinusoidal, quadrático e linear com a assistência do operador de migração com *blending*, uma melhoria relativamente ao método de codificação real. Repara-se também que o modelo trapezoidal continuou a apresentar os piores resultados, sendo os resultados obtidos aqui, piores ao método anterior. Mas, de forma geral, a codificação binária conseguiu atingir e até melhorar os resultados obtidos anteriormente. Assim, é possível concluir que a codificação binária de soluções poderá ser uma boa ferramenta a acrescentar ao método.

5.3 Dimensionamento de uma Viga Soldada

O problema da viga soldada é um problema clássico na área da otimização, onde o objetivo é minimizar o custo de execução da junta soldada, sujeito a restrições na tensão de corte na soldadura e na tensão de flexão na viga. É um problema bastante não-linear, tanto no caso da função objetivo, quanto na aplicação das restrições. A Figura 5.1, apresenta a geometria ser estudada.

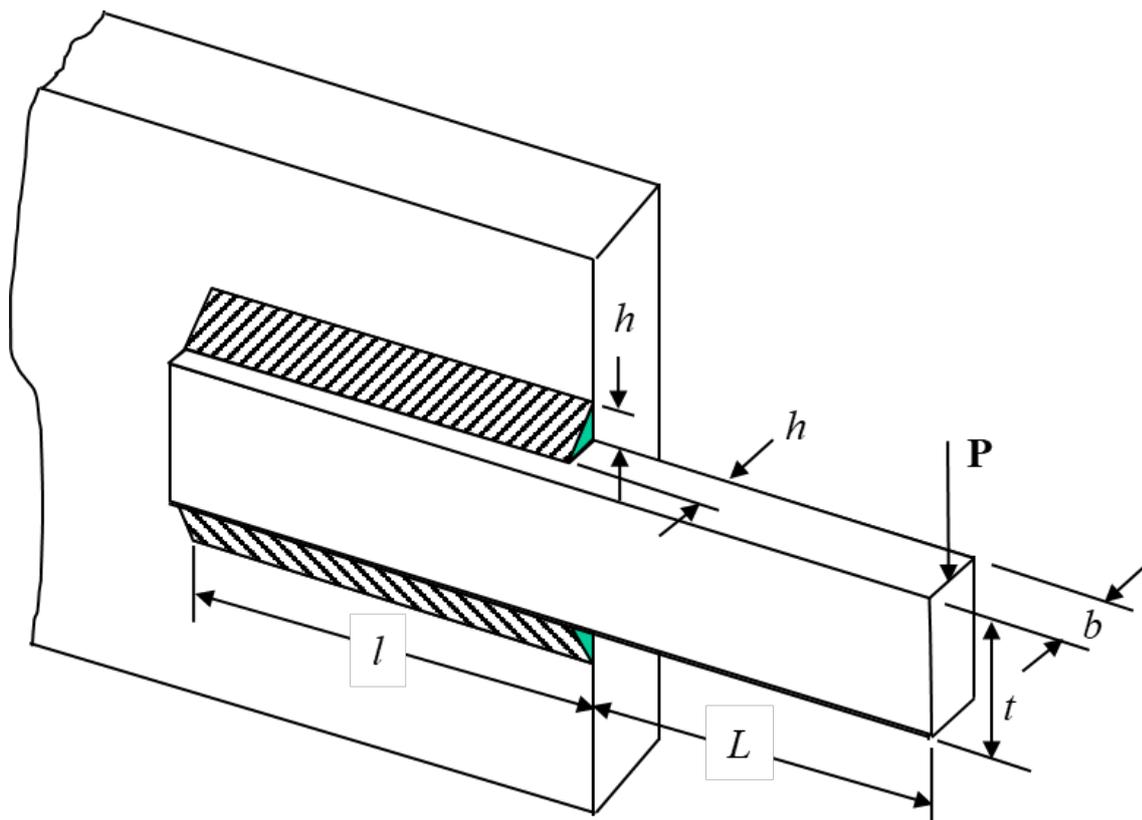


Figura 5.1: Geometria do problema da viga soldada.

De um ponto de vista geométrico, a função objetivo pode ser descrita da seguinte forma,

$$f(x_1, x_2, x_3, x_4) = (C_1 + C_3) \cdot V_{soldadura} + C_2 \cdot V_{viga} \quad (5.6)$$

com $(x_1, x_2, x_3, x_4) = (h, l, t, b)$ e onde $C_1 = 0.10471$, $C_2 = 0.04811$, $C_3 = 1$ e

$$V_{soldadura} = x_1^2 \cdot x_2 \quad (5.7)$$

$$V_{viga} = x_3 \cdot x_4 \cdot (L + x_2) \quad (5.8)$$

As restrições impostas sobre o problema são as seguintes,

$$g_1(\mathbf{x}) = \tau(\mathbf{x}) - \tau_{max} \leq 0 \quad (5.9)$$

$$g_2(\mathbf{x}) = \sigma(\mathbf{x}) - \sigma_{max} \leq 0 \quad (5.10)$$

$$g_3(\mathbf{x}) = x_1 - x_4 \leq 0 \quad (5.11)$$

$$g_4(\mathbf{x}) = C_1 \cdot x_1^2 + C_2 \cdot x_3 \cdot x_4 \cdot (L + x_2) - 5 \leq 0 \quad (5.12)$$

$$g_5(\mathbf{x}) = 0,125 - x_1 \leq 0 \quad (5.13)$$

$$g_6(\mathbf{x}) = \delta(\mathbf{x}) - \delta_{max} \leq 0 \quad (5.14)$$

$$g_7(\mathbf{x}) = P - P_c(\mathbf{x}) \leq 0 \quad (5.15)$$

$$g_8 \text{ to } g_{11} : 0,1 \leq x_i \leq 2, \quad i = 1,4 \quad (5.16)$$

$$g_{12} \text{ to } g_{15} : 0,1 \leq x_i \leq 10, \quad i = 2,3 \quad (5.17)$$

onde,

$$\tau(\mathbf{x}) = \sqrt{(\tau')^2 + 2 \cdot \tau' \tau'' \frac{x_2}{2R} + (\tau'')^2} \quad (5.18)$$

$$\tau' = \frac{P}{\sqrt{2} \cdot x_1 \cdot x_2} \quad (5.19)$$

$$\tau'' = \frac{M \cdot R}{J} \quad (5.20)$$

$$M = P \left(L + \frac{x_2}{2} \right) \quad (5.21)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2} \right)^2} \quad (5.22)$$

$$J = 2 \left[x_1 \cdot x_2 \sqrt{2} \cdot \left(\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2} \right)^2 \right) \right] \quad (5.23)$$

$$\sigma(\mathbf{x}) = \frac{6PL}{x_4 \cdot x_3^2} \quad (5.24)$$

$$\delta(\mathbf{x}) = \frac{4PL^3}{E x_4 \cdot x_3^2} \quad (5.25)$$

$$P_c(\mathbf{x}) = \frac{4.013E \cdot \sqrt{\frac{x_3^2 \cdot x_4^6}{36}}}{L^2} \cdot \left(1 - \frac{x_3}{2 \cdot L} \cdot \sqrt{\frac{E}{4G}} \right) \quad (5.26)$$

onde $P = 6000$ lb, $E = 30 \cdot 10^6$ psi, $G = 12 \cdot 10^6$ psi, $L = 14$ in, $\sigma_{max} = 30000$ psi, $\tau_{max} = 13600$ psi, $\delta_{max} = 0.25$.

Neste caso, o mérito será definido da seguinte forma,

$$\text{HSI}(\mathbf{x}) = C - (f(\mathbf{x}) + \lambda \cdot \Gamma(\mathbf{x})) \quad (5.27)$$

onde $f(\mathbf{x})$ é a função objetivo, $\Gamma(\mathbf{x}) \geq 0$ é a função penalidade e C e λ são duas constantes igual a 10 e 100, respetivamente. A função penalidade, inicialmente, será calculada,

somente se a restrição for violada, da seguinte forma,

$$\Gamma(\mathbf{x}) = \sum_{i=1}^7 g_i(\mathbf{x})^2 \quad (5.28)$$

A solução ótima deste problema foi retirada de [68] e é,

$$f^*(0.20572963, 3.4704886, 9.0366239, 0.20572963) = 1.7248523085 \quad (5.29)$$

5.3.1 Resultados

Os testes apresentados nesta secção, tentarão ilustrar a variação do comportamento do algoritmo com os diferentes operadores e modelos. Estes serão comparados não só individualmente, mas também em conjunto.

Testes com Migração e *Blending*

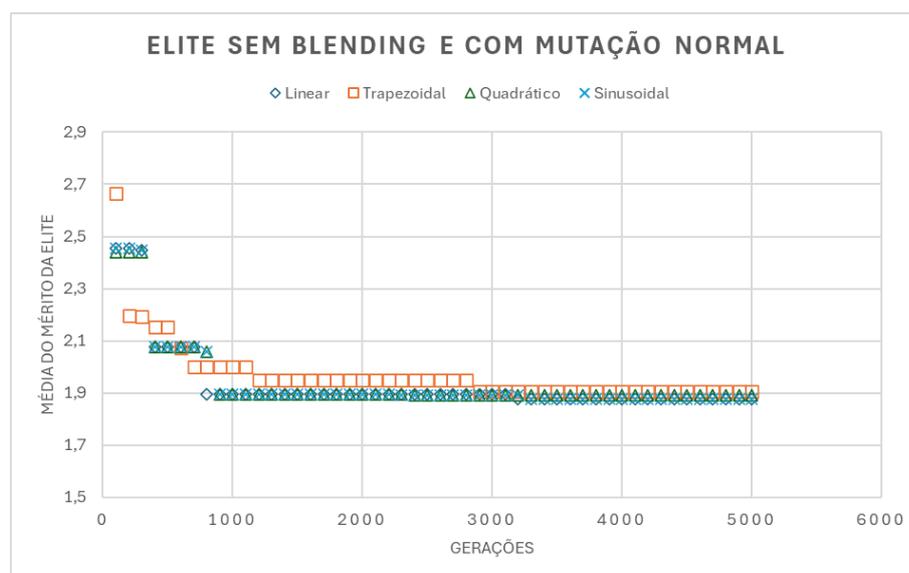
Para avaliar e classificar o melhor modelo de migração, estes serão comparados não só à melhor solução, mas também relativamente à média do mérito das soluções da elite. Os resultados serão comparados, com e sem a utilização do *blending*, para também tentar entender a influência desta estratégia e a sua relação e influência com os diferentes modelos de migração. Este estudo também irá incluir a utilização do operador de mutação taxonómica, e a sua comparação relativamente ao operador de mutação normal do BBO. Para tal, todos os testes irão utilizar a mesma população inicial e o mesmo critério de paragem, que, neste caso, será o número de gerações máxima, que será 5000, para este teste.

A Tabela 5.4 apresenta os resultados para as melhores soluções ao fim das 5000 gerações para cada caso.

É possível notar, de imediato, que o método não chegou ao ponto ótimo com nenhum dos modelos. Também é possível notar que a mutação taxonómica não apresenta bons resultados quando utilizada sem o *blending*. Este facto, deve estar relacionado com o facto deste operador modificar a população inteira, exceto a elite, o que, sem a estratégia de mistura, faz com que se percam bons genes ou SIVs, muito cedo. Assim, poderá existir uma convergência prematura num ótimo local, como é o caso neste exemplo, ou, no pior dos casos, em divergência de soluções. Quanto aos melhores resultados, realça-se as soluções geradas pelo modelo sinusoidal e linear, com a estratégia de mistura e a mutação taxonómica. O modelo trapezoidal com *blending* e a mutação taxonómica, apresenta a pior solução gerada, quando não se consideram as soluções geradas com a mutação taxonómica e sem *blending*. As Figuras 5.2-5.5, apresentam a evolução da média da elite, para cada um dos modelos e combinações entre o tipo de mutação e o tipo de migração.

Tabela 5.4: Melhores soluções para o teste dos modelos de migração e mutação, sem o crossover e o otimizador local, para o exemplo da viga soldada.

Modelo	Mutação	Blending	x_1	x_2	x_3	x_4	Objetivo
Linear	Normal	Não	0,179	4,638	9,032	0,211	1,876
	Taxonómica	Não	1,100	1,100	4,100	1,100	4,747
	Normal	Sim	0,225	3,012	8,954	0,211	1,837
	Taxonómica	Sim	0,216	3,333	8,873	0,212	1,745
Trapezoidal	Normal	Não	0,179	4,638	9,032	0,211	1,876
	Taxonómica	Não	1,100	1,100	4,100	1,100	4,747
	Normal	Sim	0,190	4,043	9,257	0,213	1,877
	Taxonómica	Sim	0,222	6,270	7,468	0,293	2,553
Quadrático	Normal	Não	0,225	3,417	8,249	0,244	1,891
	Taxonómica	Não	1,100	1,100	4,100	1,100	4,747
	Normal	Sim	0,189	3,939	8,849	0,214	1,792
	Taxonómica	Sim	0,200	3,654	8,812	0,217	1,788
Sinusoidal	Normal	Não	0,179	4,638	9,032	0,211	1,876
	Taxonómica	Não	1,100	1,100	4,100	1,100	4,747
	Normal	Sim	0,170	4,643	9,097	0,206	1,833
	Taxonómica	Sim	0,222	3,208	8,987	0,206	1,743

Figura 5.2: Evolução da média do mérito da elite em 5000 gerações, sem *blending*.

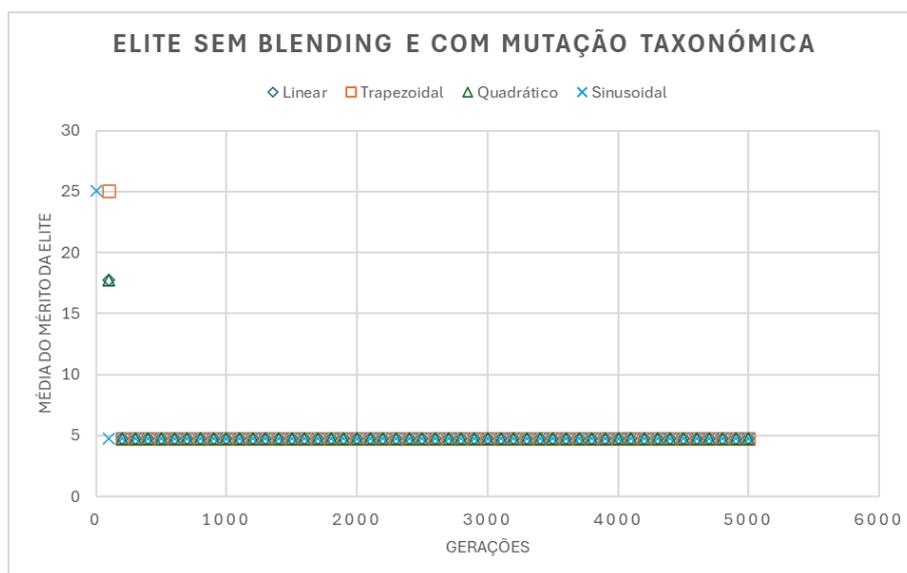


Figura 5.3: Evolução da média do mérito da elite em 5000 gerações, sem *blending* e com mutação taxonômica.

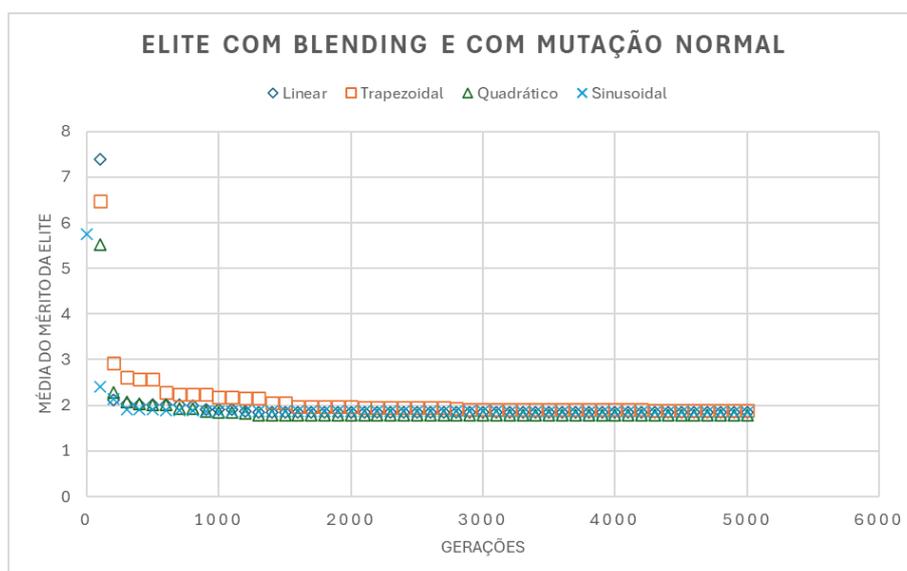


Figura 5.4: Evolução da média do mérito da elite em 5000 gerações, com *blending*.

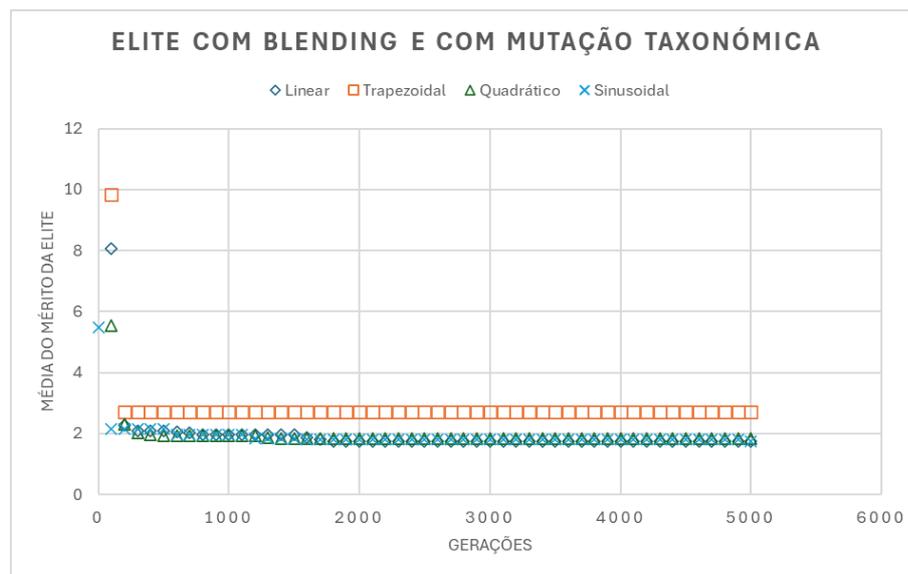


Figura 5.5: Evolução da média do mérito da elite em 5000 gerações, com *blending* e mutação taxonômica.

É possível notar a descida rápida do algoritmo, chegando à proximidade do valor final, em poucas gerações. Mas isso também demonstra o enorme problema que este algoritmo apresenta na pesquisa local, havendo ocasiões onde ele fica parado na mesma solução por várias gerações, em alguns casos mantendo-se por mais de 1000 gerações. Estas figuras também permitem ver o comportamento dos diferentes modelos, por exemplo, a Figura 5.4 permite ver o comportamento do modelo de migração trapezoidal e como ele tende a atrasar-se, relativamente às outras soluções. Esse fator pode ser uma das causas para a sua má relação com a mutação taxonômica, que de forma geral apresenta melhores resultados, mas torna o algoritmo um pouco mais lento. Mas, além deste modelo, é muito complicado interpretar qual dos modelos se destaca, para a resolução deste caso.

Para tentar corrigir a baixa convergência do método e a sua velocidade baixa nas últimas gerações, foi tentado aumentar a diversidade da população através da injeção de soluções ou da implementação do ciclo de mutações. Mas, ambos não resolveram totalmente este impedimento, por isso, foram implementados processos de crossover, de forma a facilitar a troca de informação genética e aumentar a diversidade populacional, tentando manter a informação genética das boas soluções espalhadas pela população inteira.

Testes com Operadores de Crossover

Para a análise dos operadores de crossover binários (Single-Point, Multi-Point, Uniform), apresentados no Capítulo 3, utilizando as duas estratégias descritas no capítulo anterior (seção 4.2.1), e a sua relação com os modelos de migração e o tipo de mutação. Serão utilizados os mesmos parâmetros que a última análise, ou seja, condição de paragem a 5000 gerações e população inicial idêntica. Desta vez, a população inicial será criada no domínio binário e devido aos maus resultados, obtidos anteriormente, a combinação entre

mutação taxonómica e migração sem *blending* não será apresentada. A Tabela 5.5 apresenta as melhores soluções obtidas para a primeira estratégia (secção 4.2.1) de crossover, ao fim das 5000 gerações.

Os valores obtidos pela primeira estratégia (secção 4.2.1), também não chegaram ao valor óptimo, já conhecido. E os melhores valores obtidos, neste caso, não superaram os valores obtidos anteriormente. Em termos de operador de crossover, todos conseguiram apresentar bons resultados, destacando-se o operador MP (Multi-Point Crossover), que conseguiu o maior número de bons resultados. Ambos os operadores restantes, apresentaram um comportamento semelhante, tendo gerado um número semelhante de boas soluções. Para a segunda estratégia (secção 4.2.1), as melhores soluções obtidas estão apresentadas na Tabela 5.6.

Tabela 5.5: Resultados obtidos para os operadores de crossover utilizando a primeira estratégia (secção 4.2.1) de seleção e codificação, para o exemplo da viga soldada.

Modelo	Blending	Mutação	Crossover	x_1	x_2	x_3	x_4	Objetivo
Linear	Não	Normal	SP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	MP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	U	0,179	4,638	9,032	0,211	1,876
	Sim	Normal	SP	0,206	3,702	8,958	0,213	1,799
	Sim	Normal	MP	0,211	3,521	8,883	0,212	1,761
	Sim	Normal	U	0,216	3,392	8,866	0,212	1,755
	Sim	Taxonómica	SP	0,206	3,513	8,747	0,216	1,811
	Sim	Taxonómica	MP	0,212	3,329	9,088	0,209	1,749
	Sim	Taxonómica	U	0,195	3,562	9,369	0,205	1,772
Trapezoidal	Não	Normal	SP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	MP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	U	0,179	4,638	9,032	0,211	1,876
	Sim	Normal	SP	0,179	4,638	9,032	0,211	1,876
	Sim	Normal	MP	0,196	3,725	9,156	0,209	1,791
	Sim	Normal	U	0,207	3,479	8,760	0,212	1,869
	Sim	Taxonómica	SP	1,100	1,100	4,100	1,100	4,747
	Sim	Taxonómica	MP	1,100	1,100	4,100	1,100	4,747
	Sim	Taxonómica	U	1,100	1,100	4,100	1,100	4,747
Quadrático	Não	Normal	SP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	MP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	U	0,179	4,638	9,032	0,211	1,876
	Sim	Normal	SP	0,228	3,117	8,805	0,213	1,779
	Sim	Normal	MP	0,217	3,467	9,136	0,205	1,770
	Sim	Normal	U	0,217	3,327	8,886	0,220	1,806
	Sim	Taxonómica	SP	0,208	3,462	8,880	0,212	1,750
	Sim	Taxonómica	MP	0,179	4,238	8,789	0,216	1,826
	Sim	Taxonómica	U	0,200	3,778	8,892	0,208	1,794
Sinusoidal	Não	Normal	SP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	MP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	U	0,179	4,638	9,032	0,211	1,876
	Sim	Normal	SP	0,192	4,254	9,095	0,205	1,825
	Sim	Normal	MP	0,170	4,545	8,873	0,212	1,828
	Sim	Normal	U	0,201	4,153	9,096720	0,205	1,818
	Sim	Taxonómica	SP	0,176	4,844	8,795	0,215	1,892
	Sim	Taxonómica	MP	0,178	4,158	9,071	0,208	1,793
	Sim	Taxonómica	U	0,217	3,648	9,087	0,207	1,795

Tabela 5.6: Resultados obtidos para os operadores de crossover utilizando a segunda estratégia (secção 4.2.1) de seleção e codificação, para o exemplo da viga soldada.

Modelo	Blending	Mutação	Crossover	x_1	x_2	x_3	x_4	Objetivo
Linear	Não	Normal	SP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	MP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	U	0,179	4,638	9,032	0,211	1,876
	Sim	Normal	SP	0,227	3,299	8,555	0,227	1,816
	Sim	Normal	MP	0,206	3,702	8,958	0,213	1,799
	Sim	Normal	U	0,206	3,702	8,958	0,213	1,799
	Sim	Taxonómica	SP	0,177	4,386	9,052	0,205	1,800
	Sim	Taxonómica	MP	0,172	4,503	9,040	0,207	1,814
	Sim	Taxonómica	U	0,215	3,628	8,998	0,208	1,779
Trapezoidal	Não	Normal	SP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	MP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	U	0,179	4,638	9,032	0,211	1,876
	Sim	Normal	SP	0,196	3,725	9,156	0,209	1,791
	Sim	Normal	MP	0,179	4,638	9,032	0,211	1,876
	Sim	Normal	U	0,218	4,263	9,068	0,207	1,884
	Sim	Taxonómica	SP	1,100	1,100	4,100	1,100	4,747
	Sim	Taxonómica	MP	1,100	1,100	4,100	1,100	4,747
	Sim	Taxonómica	U	1,100	1,100	4,100	1,100	4,747
Quadrático	Não	Normal	SP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	MP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	U	0,179	4,638	9,032	0,211	1,876
	Sim	Normal	SP	0,191	3,939	9,066	0,206	1,773
	Sim	Normal	MP	0,228	3,117	8,805	0,213	1,779
	Sim	Normal	U	0,191	3,751	9,139	0,211	1,797
	Sim	Taxonómica	SP	0,222	3,421	8,825	0,215	1,782
	Sim	Taxonómica	MP	0,208	3,462	8,880	0,212	1,750
	Sim	Taxonómica	U	0,177	4,304	8,919	0,210	1,800
Sinusoidal	Não	Normal	SP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	MP	0,179	4,638	9,032	0,211	1,876
	Não	Normal	U	0,179	4,638	9,032	0,211	1,876
	Sim	Normal	SP	0,196	3,728	9,080	0,205	1,754
	Sim	Normal	MP	0,192	4,254	9,095	0,205	1,825
	Sim	Normal	U	0,227	2,917	9,748	0,206	1,845
	Sim	Taxonómica	SP	0,207	3,595	9,115	0,208	1,771
	Sim	Taxonómica	MP	0,176	4,844	8,795	0,215	1,892
	Sim	Taxonómica	U	0,203	3,549	9,647	0,201	1,898

Com a segunda estratégia (secção 4.2.1), o operador que mais se destacou foi o operador Single-Point (SP), mesmo sendo o operador MP a produzir a melhor solução, o operador SP,

conseguiu, consistentemente, chegar a bons resultados. O operador de crossover uniforme (U), não apresentou maus resultados, mas quando comparado aos outros operadores ele apresentou os piores. Destaca-se também a melhoria, nas soluções finais obtidas pelo modelo de migração quadrático, que conseguiu obter vários resultados positivos.

De forma geral, o operador crossover não criou melhores soluções, continuando longe do valor de referência. Mas, este operador conseguiu melhorar as soluções dos vários modelos de migração, como foi o caso do modelo quadrático, cujos resultados tiveram uma melhoria significativa. Por outro lado, a implementação do operador crossover destacou, mais uma vez, a má relação entre o modelo trapezoidal e a mutação taxonómica, chegando a valores piores. Pode-se dizer, que para este exemplo o efeito do operador crossover não é significativo o suficiente.

Testes com otimizador Local (Hooke-Jeeves)

Os parâmetros iniciais e de paragem irão manter-se para este estudo, que visa estudar o efeito do operador de otimização local no algoritmo e a sua relação com os restantes operadores. Para este caso a mutação taxonómica sem *blending*, não será aplicada aos testes com os operadores de crossover, visto que, esta combinação continuou a demonstrar maus resultados.

Inicialmente, as soluções obtidas para o estudo do otimizador local, sem a utilização do operador de crossover, são apresentadas na Tabela 5.7. De forma geral, os resultados não sofreram uma melhoria significativa, quando comparados aos resultados obtidos anteriormente, mas olhando para os resultados obtidos no modelo trapezoidal, é possível reparar que o operador conseguiu ultrapassar o ponto óptimo definido anteriormente. O que demonstra, obviamente, o efeito positivo que este operador pode oferecer.

Tabela 5.7: Resultados do teste com otimizador local, sem crossover, para o exemplo da viga soldada.

Modelo	<i>Blending</i>	Mutação	x_1	x_2	x_3	x_4	Objetivo
Linear	Não	Normal	0,179261	4,638446	9,031602	0,21132	1,876
	Não	Taxonómica	0,159694	6,97437	9,071216	0,208317	2,103
	Sim	Normal	0,203113	3,616415	8,825093	0,214749	1,773
	Sim	Taxonómica	0,259288	2,886056	8,112492	0,25473	1,895
Trapezoidal	Não	Normal	0,179261	4,638446	9,031602	0,21132	1,876
	Não	Taxonómica	0,159694	6,97437	9,071216	0,208317	2,103
	Sim	Normal	0,211764	3,336497	9,027827	0,205671	1,719
	Sim	Taxonómica	0,211725	3,336896	9,028142	0,205674	1,719
Quadrático	Não	Normal	0,192266	3,951089	8,85521	0,217564	1,825
	Não	Taxonómica	0,159694	6,97437	9,071216	0,208317	2,103
	Sim	Normal	0,216438	3,304898	8,871531	0,214421	1,755
	Sim	Taxonómica	0,231468	3,307541	8,157366	0,250357	1,903
Sinusoidal	Não	Normal	0,179261	4,638446	9,031602	0,21132	1,876
	Não	Taxonómica	0,159694	6,97437	9,071216	0,208317	2,103
	Sim	Normal	0,163693	4,722018	8,960535	0,209618	1,831
	Sim	Taxonómica	0,264861	2,769915	8,195406	0,249321	1,889

Para facilitar a visualização do impacto do otimizador local, ao longo da evolução da população, foram criados gráficos que representam a evolução da média da elite ao longo das gerações. As figuras 5.6-5.9, mostram esses gráficos.

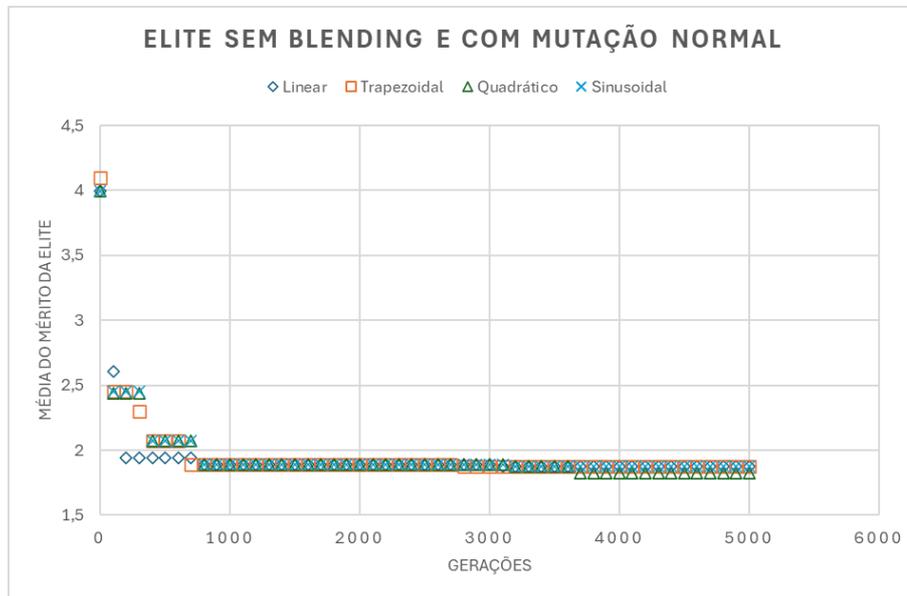


Figura 5.6: Evolução da média do mérito da elite em 5000 gerações com a utilização do otimizador local, sem *blending*.

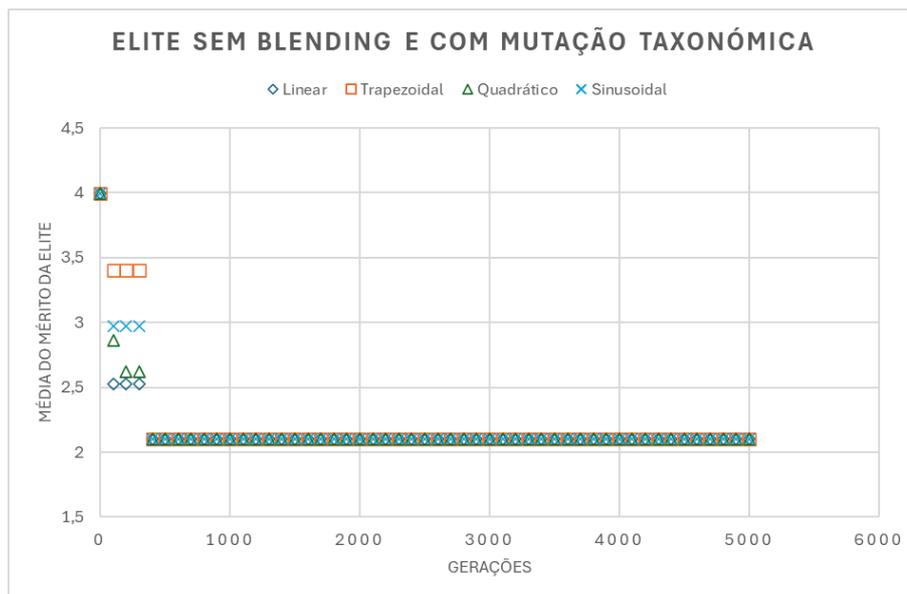


Figura 5.7: Evolução da média do mérito da elite em 5000 gerações com a utilização do otimizador local, com mutação taxonômica.

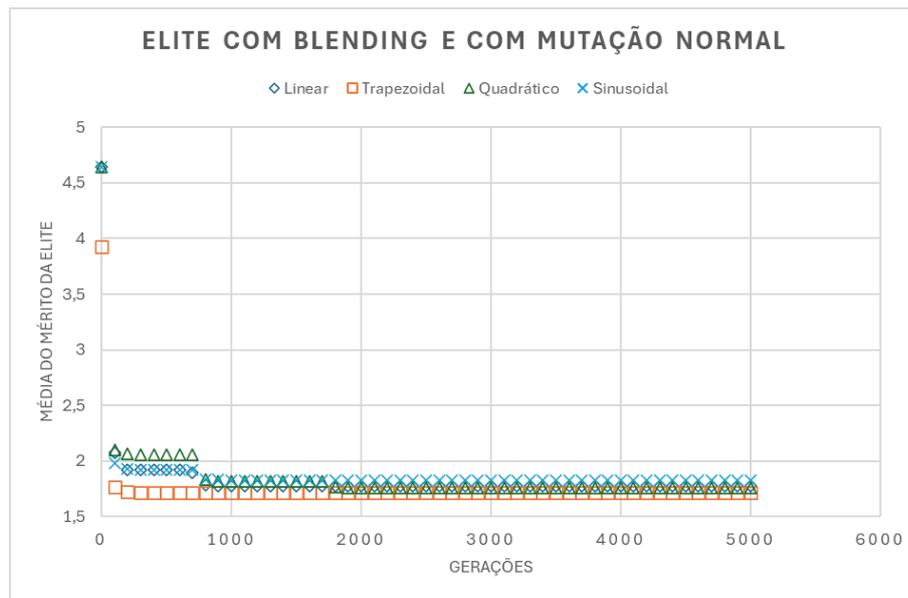


Figura 5.8: Evolução da média do mérito da elite em 5000 gerações com a utilização do otimizador local, com *blending*.

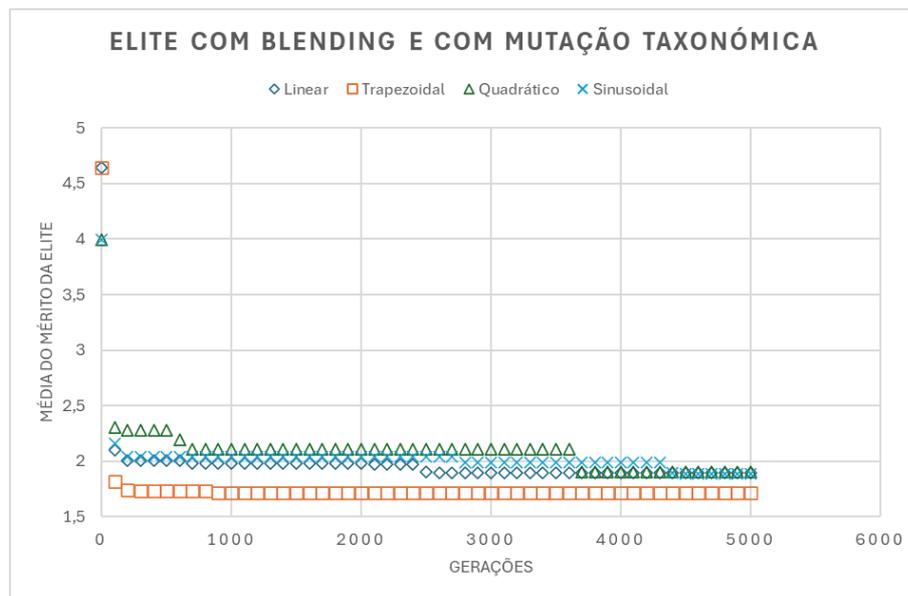


Figura 5.9: Evolução da média do mérito da elite em 5000 gerações com a utilização do otimizador local, com *blending* e mutação taxonômica.

As figuras apresentadas, representam, mais uma vez, o comportamento excelente do método na pesquisa global, apresentando uma descida rápida até soluções próximas da ótima. Mas, com a adição do otimizador local, nota-se que o algoritmo perdeu um pouco da tendência de encavar num ponto que não o ótimo, por várias gerações. Sendo esse aspecto bastante evidente para o teste com mutação taxonômica e *blending* (Figura 5.9). Agora é possível reparar, um pouco melhor, no impacto do otimizador local. Finalmente, juntaram-

se os operadores de crossover ao otimizador local. As melhores soluções obtidas, para as duas estratégias (secção 4.2.1), estão apresentados nas Tabelas 5.8-5.9, respetivamente.

Tabela 5.8: Resultados do teste com otimizador local, com crossover utilizando a primeira estratégia (secção 4.2.1), para o exemplo da viga soldada.

Modelo	Blending	Mutação	Crossover	x_1	x_2	x_3	x_4	Objetivo	Gerações
Linear	Não	Normal	SP	0,1793	4,6384	9,0316	0,2113	1,8760540	-
	Não	Normal	MP	0,1967	4,6484	8,9602	0,2108	1,893675	-
	Não	Normal	U	0,1793	4,6384	9,0316	0,2113	1,876054	-
	Sim	Normal	SP	0,2122	3,3282	9,0273	0,2057	1,719216	1223
	Sim	Normal	MP	0,2118	3,3363	9,0257	0,2057	1,719250	629
	Sim	Normal	U	0,2117	3,3388	9,0273	0,2057	1,719196	3564
	Sim	Taxonómica	SP	0,2118	3,3353	9,0276	0,2057	1,719196	1814
	Sim	Taxonómica	MP	0,2118	3,3355	9,0311	0,2056	1,719238	2645
	Sim	Taxonómica	U	0,2118	3,3357	9,0274	0,2057	1,719196	3175
Trapezoidal	Não	Normal	SP	0,1793	4,6384	9,0316	0,2113	1,8760540	-
	Não	Normal	MP	0,1793	4,6384	9,0316	0,2113	1,876054	-
	Não	Normal	U	0,1793	4,6384	9,0316	0,2113	1,876054	-
	Sim	Normal	SP	0,2117	3,3377	9,0269	0,2057	1,719200	297
	Sim	Normal	MP	0,2149	3,3020	8,9551	0,2089	1,730779	-
	Sim	Normal	U	0,2118	3,3381	9,0208	0,2059	1,719744	365
	Sim	Taxonómica	SP	0,2119	3,3357	9,0235	0,2057	1,719256	3334
	Sim	Taxonómica	MP	0,2120	3,3319	9,0276	0,2057	1,719202	765
	Sim	Taxonómica	U	0,2117	3,3378	9,0279	0,2057	1,719197	1495
Quadrático	Não	Normal	SP	0,1793	4,6384	9,0316	0,2113	1,8760540	-
	Não	Normal	MP	0,2250	3,4175	8,2485	0,2443	1,891274	-
	Não	Normal	U	0,1793	4,6384	9,0316	0,2113	1,876054	-
	Sim	Normal	SP	0,2118	3,3357	9,0280	0,2057	1,719196	472
	Sim	Normal	MP	0,2116	3,3400	9,0275	0,2057	1,719198	376
	Sim	Normal	U	0,2118	3,3350	9,0273	0,2057	1,719196	361
	Sim	Taxonómica	SP	0,2117	3,3363	9,0292	0,2057	1,719204	1148
	Sim	Taxonómica	MP	0,7834	1,3514	4,6124	0,7837	3,591797	-
	Sim	Taxonómica	U	0,2116	3,3400	9,0274	0,2057	1,719198	1455
Sinusoidal	Não	Normal	SP	0,1793	4,6384	9,0316	0,2113	1,8760540	-
	Não	Normal	MP	0,1663	5,3258	9,2080	0,2051	1,918331	-
	Não	Normal	U	0,1793	4,6384	9,0316	0,2113	1,876054	-
	Sim	Normal	SP	0,2116	3,3401	9,0278	0,2057	1,719198	307
	Sim	Normal	MP	0,2118	3,3350	9,0275	0,2057	1,719196	392
	Sim	Normal	U	0,2118	3,3357	9,0281	0,2057	1,719196	355
	Sim	Taxonómica	SP	0,2118	3,3364	9,0276	0,2057	1,719196	3328
	Sim	Taxonómica	MP	1,0000	1,1995	3,2531	1,6281	5,201549	-
	Sim	Taxonómica	U	0,2117	3,3376	9,0273	0,2057	1,719195	2690

Tabela 5.9: Resultados do teste com otimizador local, com crossover utilizando a segunda estratégia (secção 4.2.1), para o exemplo da viga soldada.

Modelo	Blending	Mutação	Crossover	x_1	x_2	x_3	x_4	Objetivo	Gerações
Linear	Não	Normal	SP	0,1793	4,6384	9,0316	0,2113	1,876054	-
	Não	Normal	MP	0,1793	4,6384	9,0316	0,2113	1,876054	-
	Não	Normal	U	0,1793	4,6384	9,0316	0,2113	1,876054	-
	Sim	Normal	SP	0,2121	3,3301	9,0277	0,2057	1,719208	3509
	Sim	Normal	MP	0,2122	3,3282	9,0273	0,2057	1,719216	1223
	Sim	Normal	U	0,2125	3,3208	9,0295	0,2057	1,719288	441
	Sim	Taxonómica	SP	0,2118	3,3355	9,0279	0,2057	1,719196	2777
	Sim	Taxonómica	MP	0,2117	3,3371	9,0321	0,2056	1,719328	2303
	Sim	Taxonómica	U	0,2117	3,3375	9,0274	0,2057	1,719195	4538
	Trapezoidal	Não	Normal	SP	0,1793	4,6384	9,0316	0,2113	1,876054
Não		Normal	MP	0,1793	4,6384	9,0316	0,2113	1,876054	-
Não		Normal	U	0,1793	4,6384	9,0316	0,2113	1,876054	-
Sim		Normal	SP	0,2118	3,3371	9,0273	0,2057	1,719196	357
Sim		Normal	MP	0,2117	3,3377	9,0269	0,2057	1,719200	297
Sim		Normal	U	0,2117	3,3381	9,0276	0,2057	1,719195	1605
Sim		Taxonómica	SP	0,2117	3,3372	9,0277	0,2057	1,719195	1774
Sim		Taxonómica	MP	0,2117	3,3377	9,0266	0,2057	1,719203	1079
Sim		Taxonómica	U	0,2116	3,3361	9,0407	0,2055	1,719830	624
Quadrático		Não	Normal	SP	0,1793	4,6384	9,0316	0,2113	1,876054
	Não	Normal	MP	0,1793	4,6384	9,0316	0,2113	1,876054	-
	Não	Normal	U	0,1793	4,6384	9,0316	0,2113	1,876054	-
	Sim	Normal	SP	0,2114	3,3447	9,0274	0,2057	1,719212	430
	Sim	Normal	MP	0,2118	3,3357	9,0280	0,2057	1,719196	472
	Sim	Normal	U	0,2116	3,3390	9,0274	0,2057	1,719196	792
	Sim	Taxonómica	SP	0,2117	3,3367	9,0301	0,2056	1,719221	960
	Sim	Taxonómica	MP	0,2117	3,3353	9,0324	0,2056	1,719280	2344
	Sim	Taxonómica	U	0,2118	3,3363	9,0277	0,2057	1,719195	3053
	Sinusoidal	Não	Normal	SP	0,1793	4,6384	9,0316	0,2113	1,876054
Não		Normal	MP	0,1793	4,6384	9,0316	0,2113	1,876054	-
Não		Normal	U	0,1793	4,6384	9,0316	0,2113	1,876054	-
Sim		Normal	SP	0,2117	3,3370	9,0279	0,2057	1,719196	344
Sim		Normal	MP	0,2116	3,3401	9,0278	0,2057	1,719198	307
Sim		Normal	U	0,2116	3,3386	9,0295	0,2057	1,719210	3839
Sim		Taxonómica	SP	0,2115	3,3425	9,0269	0,2057	1,719210	278
Sim		Taxonómica	MP	0,2118	3,3364	9,0276	0,2057	1,719196	3328
Sim		Taxonómica	U	0,2118	3,3368	9,0261	0,2057	1,719206	3580

Como é possível ver, com a inclusão do crossover, foi possível não só chegar ao valor óptimo de referência, como também foi possível ultrapassá-lo. Assim, também é possível testemunhar a velocidade tremenda que o algoritmo conseguiu alcançar, tendo chegado a ótimas soluções em menos de 1000 gerações em alguns casos. Em termos de operadores de crossover, o operador MP foi o menos consistente, para esta estratégia, sendo dos poucos operadores que não conseguiu chegar a soluções ótimas em 5000 gerações. Em termos de modelo de migração destaca-se o modelo quadrático que apresentou alguns dos

resultados mais rápidos. É importante destacar que os testes sem mutação taxonómica e com *blending*, foram os mais rápidos. Mas, com a utilização da mutação taxonómica os resultados, de forma geral, tendem a ser mais precisos e a população evolui de uma forma mais estável. Reforça-se mais uma vez que a mutação taxonómica utiliza dados da população atual, no processo de mutação, por isso, este operador tende a priorizar estabilizar a população a cada geração, deixando a procura para os operadores de migração. É importante realçar, também, que a mutação taxonómica tem uma enorme sinergia com este operador de otimização local, visto que, ele utiliza informação da pior e melhor solução para o cálculo do passo inicial.

As soluções obtidas, para a segunda estratégia (Tabela 5.9), também obtiveram resultados excelente, conseguindo atingir o ponto óptimo de referência repetidamente. Mais uma vez, o operador SP, conseguiu destacar-se, em termos de velocidade, para este problema. Mas, realça-se também o operador MP, que conseguiu melhorar significativamente, seguindo esta estratégia. Em termos de modelo, o sinusoidal, aparenta ser o melhor. Embora a melhor solução tenha sido obtida pelo modelo trapezoidal.

Desta vez, a mutação taxonómica apresentou uma velocidade superior, mas perdeu um pouco da precisão, que havia sido antes mencionada. Pelo contrário, os resultados para os testes só com *blending*, perderam um pouco da sua velocidade, mas notou-se uma precisão superior.

De forma geral, a segunda estratégia (secção 4.2.1) foi a mais rápida, além disso, também foi a mais consistente, não conseguindo apresentar resultados óptimos para os testes sem *blending* e mutação normal, algo que também ocorreu para a primeira estratégia. Finalmente, em relação ao otimizador local. O seu impacto já era sentido nos testes anteriores, mas agora a sua utilidade é inquestionável. A adição do otimizador local, conseguiu melhorar a maioria dos resultados obtidos nos testes anteriores, melhorando até o valor óptimo de referência, numa grande parte dos casos. Os gráficos, que representam a evolução do mérito médio da elite ao longo das gerações para estes testes e os testes apenas para os operadores de crossover estão apresentados no apêndice A.

5.4 Dimensionamento de um Reforço Soldado Gusset

Uma geometria com um perfil L (100x65x11[mm]) e com uma área de secção $A = 1.7 \times 10^{-3}[\text{m}^2]$, tem de ser soldado a uma placa com uma espessura de $1.4 \times 10^{-3}m$ de forma que o perfil possa ser sujeito apenas a tensão, tal como está apresentado na Figura 5.10. O material considerado para o perfil é um aço St37, com uma tensão admissível $\sigma_{adm} = 137.34$ MPa, e o metal utilizado na soldagem tem a tensão admissível $\sigma_{adm,weld} = 107.91$ MPa.

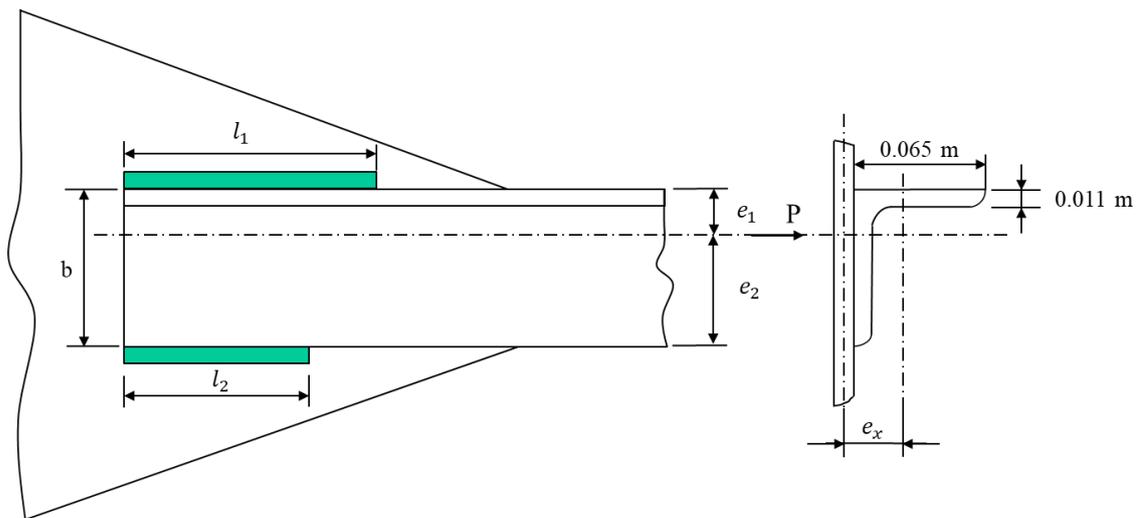


Figura 5.10: Dimensionamento de um reforço soldado Gusset.

O custo para a soldadura inclui o custo da preparação da junta, o custo do material depositado e o custo associado à técnica de soldadura. Supondo que o custo da operação possa ser expresso em função das dimensões da geometria do problema, a função objetivo é descrita da seguinte forma,

$$Cost = K (l_1 \cdot a_1 + l_2 \cdot a_2) \quad (5.30)$$

onde K é o custo associado ao material e à mão de obra, l_1 e l_2 são os comprimentos dos filetes de soldadura, a_1 e a_2 são as larguras respectivas.

Para avaliar as tensões causadas pelo momento de flexão é necessário avaliar o módulo de flexão, Z_f ,

$$Z_f = Z_{f1} + Z_{f2} = \frac{1}{6} \cdot (l_1^2 \cdot a_1 + l_2^2 \cdot a_2) \quad (5.31)$$

A tensão normal, resultante do momento de flexão, é definido pela seguinte expressão,

$$n = \frac{M_{max}}{Z_f} \quad (5.32)$$

onde $M_{max} = 5533.428$ Nm.

A tensão nos filetes de soldadura, paralela ao seu eixo, é dada por,

$$t_{//} = \frac{P}{l_1 \cdot a_1 + l_2 \cdot a_2} \quad (5.33)$$

onde $P = 233.478$ kN.

Finalmente, de acordo com o Regulamento de Estruturas de Aço para Edifícios (REAE) [69] utilizadas em edifícios, a tensão equivalente é dada pela seguinte fórmula,

$$\sigma_{eq} = \sqrt{1.4n^2 + 1.8t_{//}^2} \quad (5.34)$$

A restrição do problema associada à tensão equivalente é dada da seguinte forma,

$$\sigma_{eq} \leq \bar{\alpha} \sigma_{adm,well} \quad (5.35)$$

onde $\bar{\alpha}$ é um fator de segurança definido de acordo com a REAE (art^o77) [69], da seguinte forma,

$$\alpha_i = 0.8 \left(1 + \frac{1}{\bar{a}_i} \right) \quad (5.36)$$

com

$$\bar{a}_1 = a_1 \quad [\text{mm}] \quad (5.37)$$

$$\bar{a}_2 = a_2 \quad [\text{mm}] \quad (5.38)$$

sendo que o art^o35 da REAE define que,

$$a_i \geq 3 \quad [\text{mm}], \quad i = 1, 2 \quad (5.39)$$

O fator de segurança é calculado através da seguinte fórmula,

$$\bar{\alpha} = \text{Min}(\alpha_i, i = 1, 2) \quad (5.40)$$

Finalmente, as variáveis deverão cumprir estas limitações,

$$3 \leq a_1 \leq 20[\text{mm}] \quad (5.41)$$

$$3 \leq a_2 \leq 20[\text{mm}] \quad (5.42)$$

$$20 \leq l_1 \leq 400[\text{mm}] \quad (5.43)$$

$$20 \leq l_2 \leq 250[\text{mm}] \quad (5.44)$$

O mérito será calculado utilizando o mesmo método utilizado no problema anterior (Equação 5.27), com $C = 10^4$ e $\lambda = 10^{16}$, de forma a garantir que nenhuma solução obtida viola a restrição. O ponto ótimo para este problema não é conhecido, mas devido a um estudo realizado anteriormente, o ponto que será utilizado como referência, retirado de [70], será,

$$f^*(5.13333, 5.13333, 400.0, 248.196) = 3228.13 \quad (5.45)$$

5.4.1 Resultados

Agora com um melhor conhecimento sobre o comportamento dos operadores do BBO, aplicados num problema com restrições, irá ser continuado o estudo do comportamento do algoritmo, para um novo exemplo.

Testes com Migração e *Blending*

Tal como no exemplo anterior, foi estabelecido um critério de paragem a 5000 gerações.

A população inicial, foi mantida de teste a teste, de forma, a não contaminar o processo de evolução dos diferentes modelos e operadores. Os resultados para a melhor seleção gerada estão apresentados na Tabela 5.10.

Tabela 5.10: Melhores soluções, para o exemplo da viga Gusset.

Modelo	<i>Blending</i>	Mutação	x_1	x_2	x_3	x_4	Objetivo
Linear	Não	Normal	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Taxonómica	5,433475	5,833137	385,9308	220,3023	3382
	Sim	Normal	5,332847	5,285214	377,2821	240,7241	3284,266
	Sim	Taxonómica	5,761571	6,164008	397,1027	173,0973	3354,906
Trapezoidal	Não	Normal	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Taxonómica	5,433475	5,833137	385,9308	220,3023	3382
	Sim	Normal	5,336918	5,16522	394,6165	231,1687	3300,07
	Sim	Taxonómica	5,527724	6,063087	395,4822	190,2226	3339,453
Quadrático	Não	Normal	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Taxonómica	5,433475	5,833137	385,9308	220,3023	3382
	Sim	Normal	5,578946	5,466615	378,1866	218,714	3305,508
	Sim	Taxonómica	6,013534	5,332894	390,5891	188,3892	3353,477
Sinusoidal	Não	Normal	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Taxonómica	5,433475	5,833137	385,9308	220,3023	3382
	Sim	Normal	5,152382	5,446071	383,2993	240,9044	3286,891
	Sim	Taxonómica	5,272211	5,255558	383,5891	242,9305	3299,102

Primeiramente nota-se que nenhum dos testes, conseguiu atingir o valor óptimo de referência, sendo o mais próximo o modelo linear com *blending*, sendo este o modelo de migração com mais destaque. Mais uma vez, a mutação taxonómica apresentou alguns dos piores resultados e, surpreendentemente, o efeito do *blending* não foi muito significativo, para os testes com a mutação normal. Isto, provavelmente, está relacionado com as constantes de restrição usadas na função de mérito, que restringem bastante o problema, enfraquecendo a pesquisa local do algoritmo, que, normalmente, já tem bastantes dificuldades nesta área. As Figuras 5.11-5.14 apresentam a representação gráfica da evolução do mérito médio da elite ao longo das 5000 gerações.

As figuras mostram, mais uma vez, a excelente evolução inicial do método. Nota-se na Figura 5.14, que o algoritmo apresenta um comportamento similar ao comportamento de um algoritmo evolucionário normal, com uma espécie de evolução em escada, bastante aparente, ao longo das gerações. Pelo contrário, na Figura 5.11, nota-se uma evolução brusca nas gerações iniciais e uma estabilização, nas anteriores. Ainda nessa figura, nota-se um pouco a diferença entre o modelo trapezoidal e os restantes, que evoluiu mais lentamente, fazendo o seu último salto numa geração mais tardia.

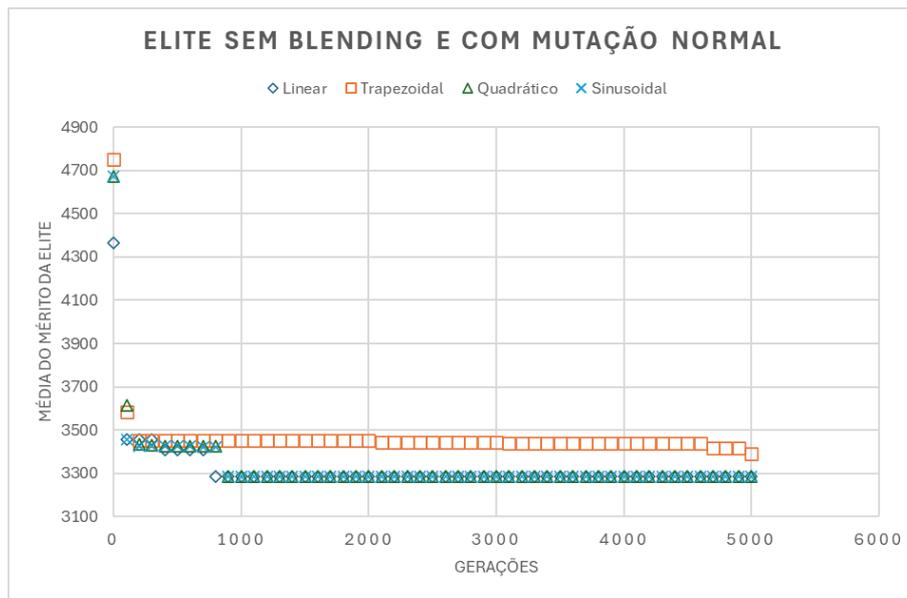


Figura 5.11: Evolução da média do mérito da elite ao longo das gerações, para o algoritmo sem *blending* e com mutação normal.

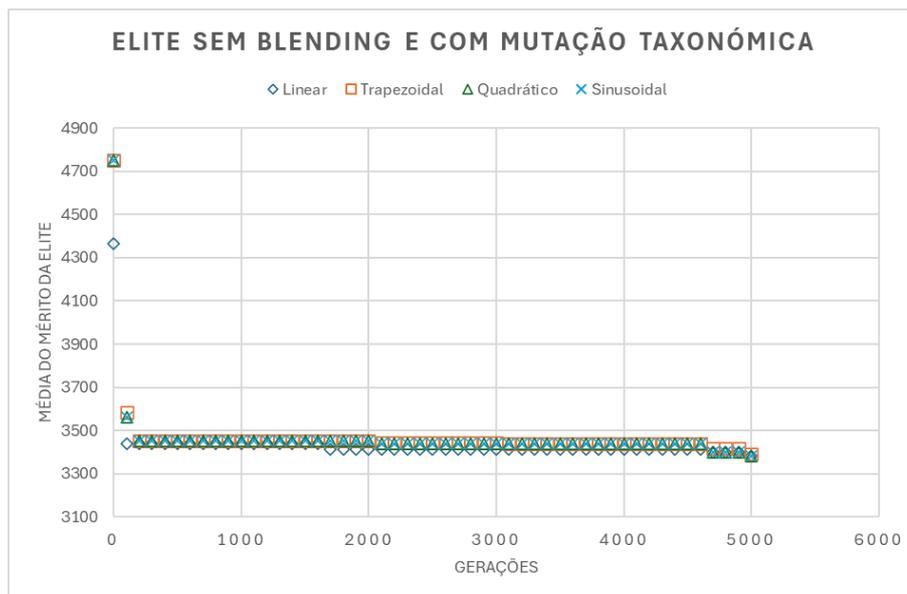


Figura 5.12: Evolução da média do mérito da elite ao longo das gerações, para o algoritmo com mutação taxonômica e sem *blending*.

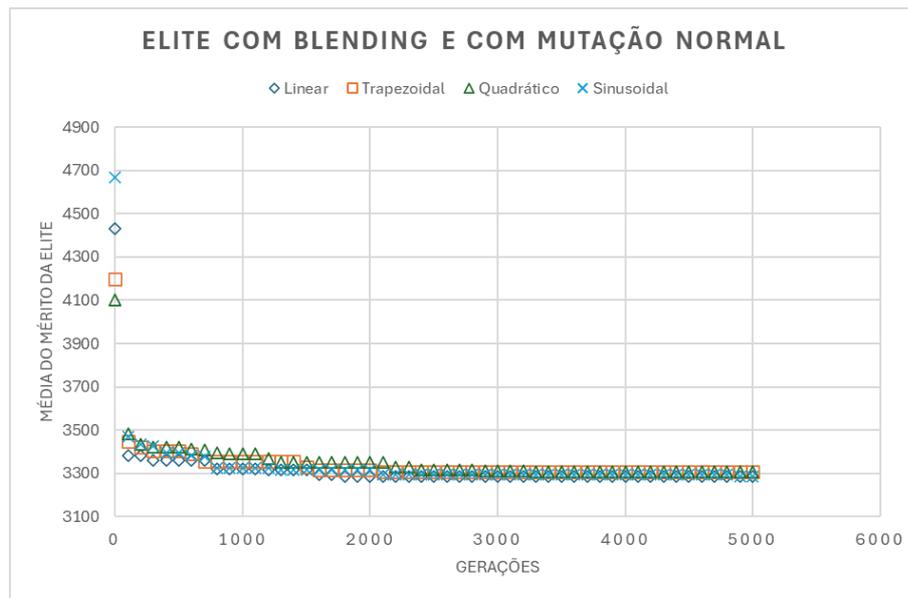


Figura 5.13: Evolução da média do mérito da elite ao longo das gerações, para o algoritmo com *blending* e com mutação normal.

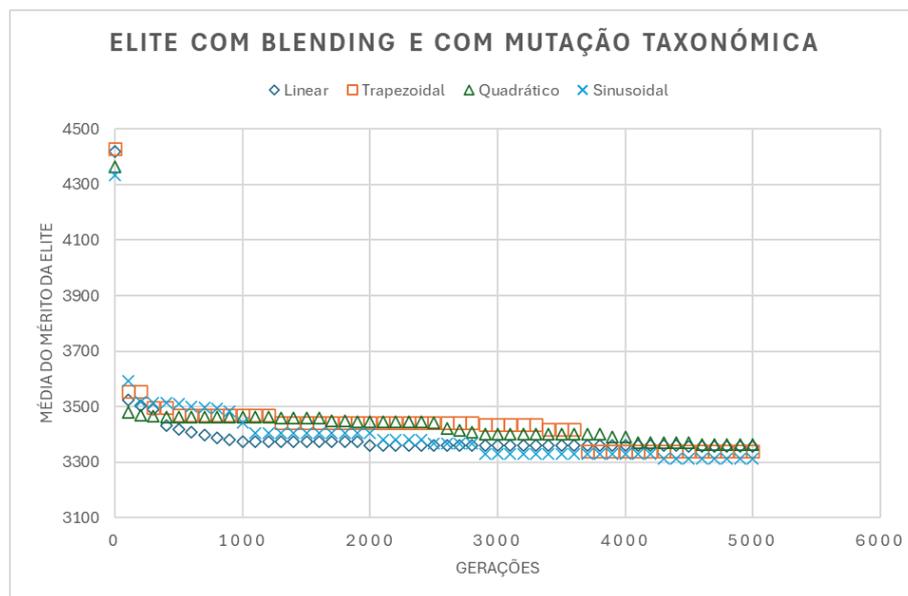


Figura 5.14: Evolução da média do mérito da elite ao longo das gerações, para o algoritmo com mutação taxonômica e *blending*.

Testes com Operadores de Crossover

Para a análise dos operadores de crossover e a sua relação com os modelos de migração e o tipo de mutação. Será utilizado o mesmo critério de paragem, o número de gerações, sendo igual a 5000 gerações. Desta vez, a população inicial será criada no domínio binário e devido aos maus resultados, obtidos durante os testes, a combinação entre mutação taxonômica e migração sem *blending* não será apresentada. Os resultados, relativamente

às melhores soluções obtidas, para os testes com os operadores de crossover, para a primeira estratégia (secção 4.2.1), na tabela 5.11.

Tabela 5.11: Melhores soluções, para o exemplo da viga Gusset, quando é utilizado o crossover, seguindo a primeira estratégia (secção 4.2.1).

Modelo	Blending	Mutação	Crossover	x_1	x_2	x_3	x_4	Objetivo
Linear	Não	Normal	SP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	MP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	U	5,152382	5,446071	383,2993	240,9044	3286,891
	Sim	Normal	SP	5,28487	5,304443	382,2594	235,7599	3270,766
	Sim	Normal	MP	5,185553	5,427181	383,0136	238,9084	3282,734
	Sim	Normal	U	5,295429	5,28798	381,4539	236,4282	3270,188
	Sim	Taxonômica	SP	3,000251	15,28378	20,00001	246,7791	3831,977
	Sim	Taxonômica	MP	5,328488	5,1643	375,5126	249,3213	3288,484
	Sim	Taxonômica	U	7,19135	4,761993	399,9995	105,1286	3377,156
Tapezoidal	Não	Normal	SP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	MP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	U	5,152382	5,446071	383,2993	240,9044	3286,891
	Sim	Normal	SP	5,939138	4,153412	397,253	229,1929	3311,273
	Sim	Normal	MP	5,309774	5,172671	396,0756	229,8542	3292,031
	Sim	Normal	U	5,309774	5,172671	396,0756	229,8542	3292,031
	Sim	Taxonômica	SP	18,97327	14,54497	60,8547	215,9698	4295,883
	Sim	Taxonômica	MP	7,42662	17,35605	20,01605	220,9203	3982,953
	Sim	Taxonômica	U	7,42662	17,35605	20,01605	220,9203	3982,953
Quadrático	Não	Normal	SP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	MP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	U	5,341187	6,122549	397,9062	198,8815	3342,953
	Sim	Normal	SP	5,265386	5,26238	382,2057	238,2136	3266,031
	Sim	Normal	MP	5,272646	5,272579	381,5501	238,1405	3267,391
	Sim	Normal	U	5,272425	5,269811	382,8992	236,7708	3266,547
	Sim	Taxonômica	SP	8,354168	6,607609	352,4554	83,4235	3495,703
	Sim	Taxonômica	MP	3	16,61221	20	231,6545	3908,289
	Sim	Taxonômica	U	3,434247	17,05105	20	226,6188	3932,773
Sinusoidal	Não	Normal	SP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	MP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	U	5,152382	5,446071	383,2993	240,9044	3286,891
	Sim	Normal	SP	5,562807	4,898472	395,8378	220,1601	3280,414
	Sim	Normal	MP	5,663686	4,750486	395,848	220,0231	3287,172
	Sim	Normal	U	5,369855	5,370972	380,8661	229,9198	3280,086
	Sim	Taxonômica	SP	6,628329	6,024014	334,9306	202,3407	3438,93
	Sim	Taxonômica	MP	8,397807	3	399,6862	20	3416,484
	Sim	Taxonômica	U	5,958023	5,656478	356,1715	218,6031	3358,602

Estas soluções ainda não se aproximaram do valor de referência, mas já apresentam alguma melhoria. Em termos de modelo o quadrático apresentou as três melhores soluções quando é usado o *blending* e a mutação normal, continuando esta a demonstrar-se superior à mutação taxonômica para este exemplo. Relativamente aos operadores de crossover o

operador Uniforme (U), apresentou várias boas soluções, mas os outros dois operadores também conseguiram um bom desempenho. Os resultados, relativamente às melhores soluções obtidas, para os testes com os operadores de crossover, para a segunda estratégia (secção 4.2.1), na Tabela 5.12.

Tabela 5.12: Melhores soluções, para o exemplo da viga Gusset, quando é utilizado o crossover, seguindo a segunda estratégia (secção 4.2.1).

Modelo	Blending	Mutação	Crossover	x_1	x_2	x_3	x_4	Objetivo
Linear	Não	Normal	SP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	MP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	U	5,712572	4,640891	396,8366	220,6024	3290,75
	Sim	Normal	SP	5,188367	5,431679	382,1897	239,4814	3283,727
	Sim	Normal	MP	5,28487	5,304443	382,2594	235,7599	3270,766
	Sim	Normal	U	5,328237	5,343098	381,7004	232,4861	3275,984
	Sim	Taxonómica	SP	9,453555	5,732154	348,8891	36,99755	3510,32
	Sim	Taxonómica	MP	3,000251	15,28378	20,00001	246,7791	3831,977
	Sim	Taxonómica	U	3	15,5824	20	243,1355	3848,633
Tapezoidal	Não	Normal	SP	5,712572	4,640891	396,8366	220,6024	3290,75
	Não	Normal	MP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	U	5,152382	5,446071	383,2993	240,9044	3286,891
	Sim	Normal	SP	5,336918	5,16522	394,6165	231,1687	3300,07
	Sim	Normal	MP	5,939138	4,153412	397,253	229,1929	3311,273
	Sim	Normal	U	5,309774	5,172671	396,0756	229,8542	3292,031
	Sim	Taxonómica	SP	7,42662	17,35605	20,01605	220,9203	3982,953
	Sim	Taxonómica	MP	10,2	3	400	20	4140
	Sim	Taxonómica	U	18,97327	14,54497	60,85468	215,9698	4295,883
Quadrático	Não	Normal	SP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	MP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	U	5,152382	5,446071	383,2993	240,9044	3286,891
	Sim	Normal	SP	5,269006	5,284339	382,4406	237,1335	3268,172
	Sim	Normal	MP	5,265386	5,26238	382,2057	238,2136	3266,031
	Sim	Normal	U	5,242824	5,240814	382,2359	240,2572	3263,141
	Sim	Taxonómica	SP	8,339838	3,10007	387,4195	61,55547	3421,844
	Sim	Taxonómica	MP	6,606286	17,48962	20	220,2101	3983,516
	Sim	Taxonómica	U	8,043692	7,782488	395,7964	28,81529	3407,922
Sinusoidal	Não	Normal	SP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	MP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	U	5,152382	5,446071	383,2993	240,9044	3286,891
	Sim	Normal	SP	5,228034	5,349194	382,2309	238,6838	3275,078
	Sim	Normal	MP	5,562807	4,898472	395,8378	220,1601	3280,414
	Sim	Normal	U	5,292537	5,292792	380,7311	237,189	3270,422
	Sim	Taxonómica	SP	6,488429	3,885157	377,7202	232,1468	3352,734
	Sim	Taxonómica	MP	6,072137	5,257766	365,9592	215,1126	3353,164
	Sim	Taxonómica	U	6,066547	5,159811	361,5228	224,9558	3353,922

Estas soluções continuam a não atingir o valor óptimo de referência. E, mais uma vez, o modelo quadrático apresentou as melhores soluções, com os modelos sinusoidal e linear

um pouco atrás. Em termos de operadores de crossover, para a segunda estratégia (secção 4.2.1), nenhum teve um grande destaque, tendo eles desempenhos bastante semelhantes para os vários testes.

Em relação às estratégias, os resultados obtidos pelas duas, foram bastante similares, mas como a segunda estratégia, conseguiu obter o melhor resultado, ela tem uma pequena vantagem. De forma geral, a inclusão do crossover apresentou melhores resultados, quando comparado ao algoritmo base, mas ainda continua longe do ponto óptimo. Revelando assim a necessidade, de aprofundar a hibridização.

Testes com otimizador Local (Hooke-Jeeves)

Agora, adicionando o otimizador local, espera-se que o algoritmo consiga, finalmente, alcançar o valor de referência apresentado. Este algoritmo será testado com e sem a adição de crossover e será exposto aos mesmos parâmetros expostos anteriormente. Os resultados, relativamente às melhores soluções obtidas, para os testes com o operador de otimização local, na tabela 5.13.

Tabela 5.13: Melhores soluções, para o exemplo da viga Gusset, quando é utilizado o otimizador local.

Modelo	<i>Blending</i>	Mutação	x_1	x_2	x_3	x_4	Objetivo
Linear	Não	Normal	5,161634	4,750654	400	250	3252,32
	Não	Taxonómica	5,370295	4,459973	400	248,2571	3255,336
	Sim	Normal	4,956563	4,934667	399,9075	249,9393	3215,531
	Sim	Taxonómica	5,763888	5,189463	383,371	212,4753	3312,344
Trapezoidal	Não	Normal	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Taxonómica	5,370295	4,459973	400	248,2571	3255,336
	Sim	Normal	4,939753	4,966836	399,9711	249,905	3217,008
	Sim	Taxonómica	6,208322	3,822512	389,2778	236,8709	3322,203
Quadrático	Não	Normal	5,087959	4,965643	400	250	3276,594
	Não	Taxonómica	5,370295	4,459973	400	248,2571	3255,336
	Sim	Normal	4,955686	4,943101	399,8264	249,6869	3215,641
	Sim	Taxonómica	5,144434	5,144147	385,1816	246,3299	3248,703
Sinusoidal	Não	Normal	5,12902	4,818823	400	250	3256,312
	Não	Taxonómica	5,370295	4,459973	400	248,2571	3255,336
	Sim	Normal	4,966445	4,940598	399,3975	249,659	3217,047
	Sim	Taxonómica	6,17873	5,318602	394,5579	168,9074	3336,219

A adição do otimizador local, provou-se uma mais-valia, já que, o algoritmo conseguiu não só atingir, como também melhorar, o valor de referência, mesmo sem a utilização do operador de crossover. Estes resultados foram obtidos para todos os modelos, quando foi aplicado o *blending* sem a mutação taxonómica, sendo o melhor resultado o resultado

obtido pelo modelo linear. As Figuras 5.15-5.18 apresentam a representação gráfica, da evolução da média da elite ao longo das gerações.

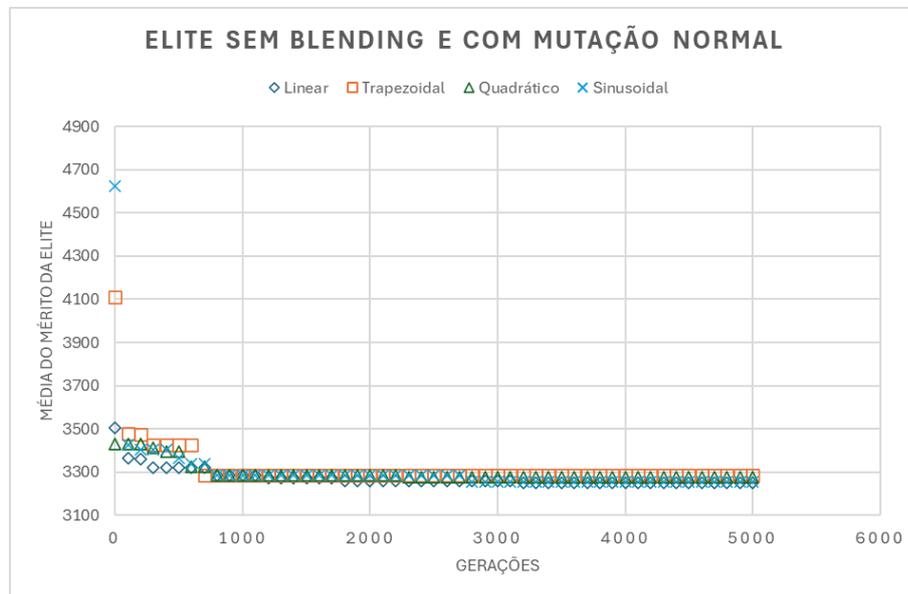


Figura 5.15: Evolução da média do mérito da elite ao longo das gerações, para o algoritmo com otimizador local, sem *blending* e com mutação normal.

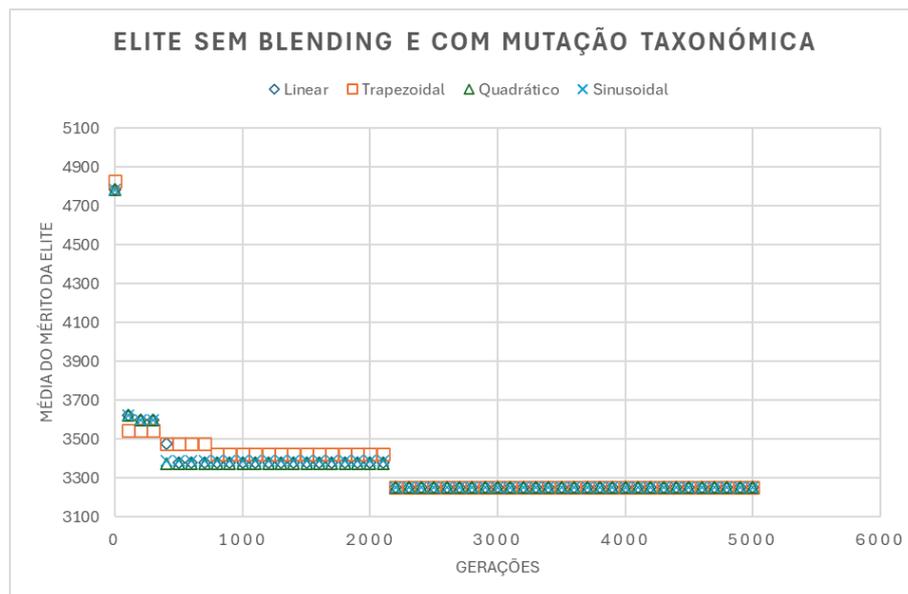


Figura 5.16: Evolução da média do mérito da elite ao longo das gerações, para o algoritmo com otimizador local e mutação taxonômica e sem *blending*.

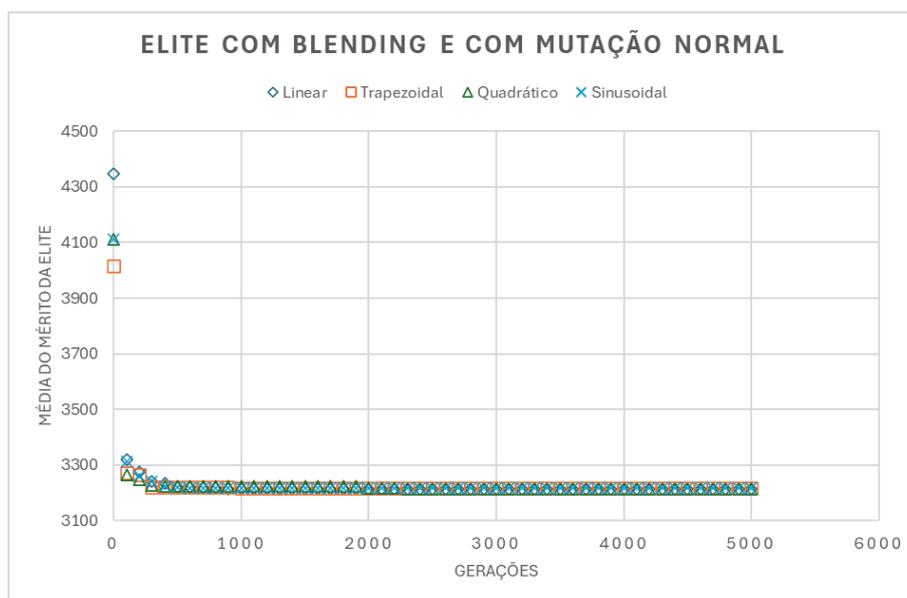


Figura 5.17: Evolução da média do mérito da elite ao longo das gerações, para o algoritmo com otimizador local, mutação normal e *blending*.

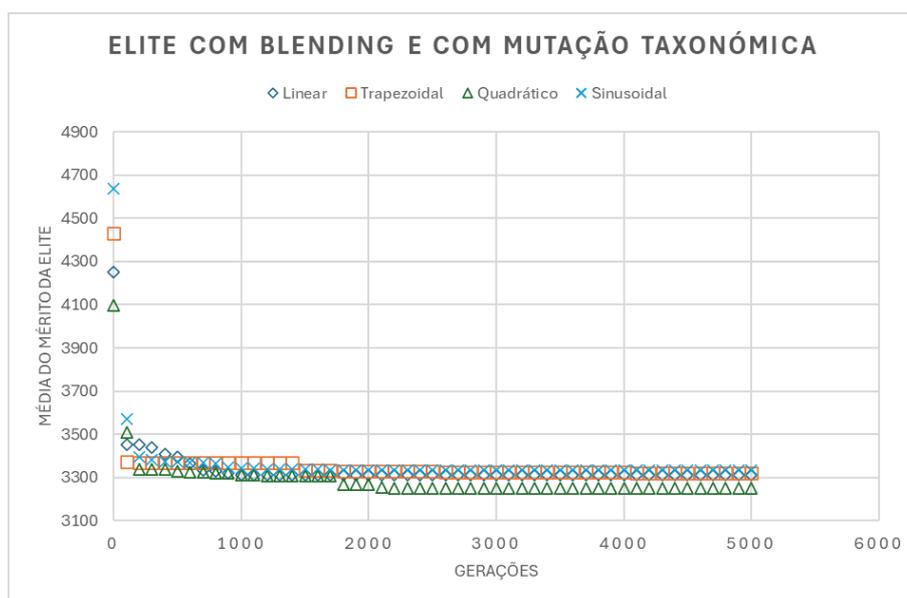


Figura 5.18: Evolução da média do mérito da elite ao longo das gerações, para o algoritmo com otimizador local, mutação taxonómica e *blending*.

Devido à escala, visto que as diferenças entre soluções é pequena, não é possível notar no gráfico a influência do otimizador local, na resolução do maior problema que o método BBO sofre, o facto de ter uma enorme tendência de encalhar em pontos óptimos locais, algo que já descrito neste relatório várias vezes. Nota-se também no comportamento mais lento do modelo trapezoidal, para este exemplo, que ocorre repetidamente para os vários testes. Adicionando agora os operadores de crossover, é esperado que o algoritmo continue

o seu bom comportamento e atinja o valor de referência. Os resultados para os testes com o operador de otimização local e crossover, seguindo a primeira estratégia (secção 4.2.1), estão representados na Tabela 5.14.

Tabela 5.14: Melhores soluções, para o exemplo da viga Gusset, para o algoritmo com otimizador local e crossover, seguindo a primeira estratégia (secção 4.2.1).

Modelo	Blending	Mutação	Crossover	x_1	x_2	x_3	x_4	Objetivo
Linear	Não	Normal	SP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	MP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	U	5,152382	5,446071	383,2993	240,9044	3286,891
	Sim	Normal	SP	5,171645	4,804964	394,7474	249,373	3239,719
	Sim	Normal	MP	4,945448	4,945447	399,9843	249,9916	3214,422
	Sim	Normal	U	4,94521	4,945175	400	250	3214,375
	Sim	Taxonómica	SP	8,269165	15,63654	38,17018	231,9017	3941,773
	Sim	Taxonómica	MP	8,425551	15,58038	38,43211	232,3277	3943,562
	Sim	Taxonómica	U	8,256133	4,882816	399,9975	20,08783	3400,516
Tapezoidal	Não	Normal	SP	5,712572	4,640891	396,8366	220,6024	3290,75
	Não	Normal	MP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	U	5,712572	4,640891	396,8366	220,6024	3290,75
	Sim	Normal	SP	5,041201	4,826195	399,9961	250	3223,008
	Sim	Normal	MP	5,887726	3,805482	400	245,2704	3288,461
	Sim	Normal	U	5,151318	5,149482	389,9966	240,4199	3247,031
	Sim	Taxonómica	SP	6,825421	3,797644	389,0195	187,13	3365,875
	Sim	Taxonómica	MP	19,99871	20	20,00015	192,1841	4243,656
	Sim	Taxonómica	U	6,825378	3,797682	389,0232	187,1245	3365,867
Quadrático	Não	Normal	SP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	MP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	U	5,152382	5,446071	383,2993	240,9044	3286,891
	Sim	Normal	SP	4,945211	4,945176	400	250	3214,375
	Sim	Normal	MP	4,9452	4,945194	400	250	3214,375
	Sim	Normal	U	4,945205	4,945194	400	249,9997	3214,375
	Sim	Taxonómica	SP	6,046756	18,36087	21,5173	212,6024	4033,672
	Sim	Taxonómica	MP	6,125863	18,39721	21,67298	211,8929	4031
	Sim	Taxonómica	U	5,075033	4,914197	399,7774	244,148	3228,672
Sinusoidal	Não	Normal	SP	5,712572	4,640891	396,8366	220,6024	3290,75
	Não	Normal	MP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	U	5,152382	5,446071	383,2993	240,9044	3286,891
	Sim	Normal	SP	4,945197	4,945206	399,9997	250	3214,375
	Sim	Normal	MP	4,945196	4,945203	400	250	3214,375
	Sim	Normal	U	4,945617	4,94502	399,9963	249,9883	3214,422
	Sim	Taxonómica	SP	16,0392	15,62296	49,14142	214,5551	4140,172
	Sim	Taxonómica	MP	15,58453	18,7653	47,00548	190,8364	4313,656
	Sim	Taxonómica	U	6,30927	16,52544	20	231,6167	3953,75

Mais uma vez, vários resultados atingiram o valor de referência. A inclusão do operador de crossover permitiu, também, que várias soluções melhorassem quando comparadas com as soluções previamente obtidas, para os testes sem crossover. Em termos de operadores,

destacam-se os operadores U e MP, que obtiveram alguns dos melhores resultados. O modelo quadrático continuou a ser um dos melhores e a mutação normal, continuou a apresentar os melhores resultados. Os resultados para o algoritmo com o otimizador local e o operador de crossover, utilizando a segunda estratégia (secção 4.2.1), estão apresentados na Tabela 5.15.

Tabela 5.15: Melhores soluções, para o exemplo da viga Gusset, para o algoritmo com operador de otimização local e crossover, seguindo a segunda estratégia (secção 4.2.1).

Modelo	Blending	Mutação	Crossover	x_1	x_2	x_3	x_4	Objetivo
Linear	Não	Normal	SP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	MP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	U	5,152382	5,446071	383,2993	240,9044	3286,891
	Sim	Normal	SP	4,945808	4,945708	400	249,9414	3214,461
	Sim	Normal	MP	4,945212	4,945174	400	250	3214,375
	Sim	Normal	U	4,945366	4,945443	399,9985	249,9819	3214,406
	Sim	Taxonômica	SP	9,897711	7,558258	340,7944	20,79257	3530,242
	Sim	Taxonômica	MP	9,228525	3	370,8145	20	3482,07
	Sim	Taxonômica	U	10,46858	7,768183	325,7272	20,55373	3569,562
Tapezoidal	Não	Normal	SP	5,318369	5,436796	371,9403	242,2179	3295,008
	Não	Normal	MP	5,712572	4,640891	396,8366	220,6024	3290,75
	Não	Normal	U	5,152382	5,446071	383,2993	240,9044	3286,891
	Sim	Normal	SP	5,215884	4,606459	399,9929	249,9953	3237,906
	Sim	Normal	MP	5,041201	4,826195	399,9961	250	3223,008
	Sim	Normal	U	4,950841	4,95081	399,998	249,4187	3215,148
	Sim	Taxonômica	SP	14,31369	14,32662	20,00192	249,1122	3855,234
	Sim	Taxonômica	MP	19,99871	20	20,00015	192,1841	4243,656
	Sim	Taxonômica	U	8,16676	6,667372	399,9705	20,00056	3399,812
Quadrático	Não	Normal	SP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	MP	5,152382	5,446071	383,2993	240,9044	3286,891
	Não	Normal	U	5,133349	5,6146	371,4709	250	3310,539
	Sim	Normal	SP	4,945198	4,945194	400	250	3214,375
	Sim	Normal	MP	4,945211	4,945176	400	250	3214,375
	Sim	Normal	U	4,945289	4,945095	399,9994	250	3214,383
	Sim	Taxonômica	SP	6,139846	18,38985	21,67834	211,945	4030,734
	Sim	Taxonômica	MP	6,046756	18,36087	21,5173	212,6024	4033,672
	Sim	Taxonômica	U	6,073485	18,39942	21,63714	211,9176	4030,57
Sinusoidal	Não	Normal	SP	5,712572	4,640891	396,8366	220,6024	3290,75
	Não	Normal	MP	5,712572	4,640891	396,8366	220,6024	3290,75
	Não	Normal	U	5,152382	5,446071	383,2993	240,9044	3286,891
	Sim	Normal	SP	4,945284	4,945091	400	250	3214,383
	Sim	Normal	MP	4,945197	4,945206	399,9997	250	3214,375
	Sim	Normal	U	4,945432	4,94498	400	249,9964	3214,398
	Sim	Taxonômica	SP	16,04728	15,63091	48,84182	214,6497	4138,945
	Sim	Taxonômica	MP	6,280905	16,49226	20,01035	230,2388	3922,844
	Sim	Taxonômica	U	5,627141	4,860899	381,0619	236,7588	3295,148

A segunda estratégia (secção 4.2.1) conseguiu obter resultados semelhantes à primeira,

desta vez destacando-se mais o operador de crossover MP (Multi-Point). Os modelos quadrático, linear e sinusoidal obtiveram os melhores resultados, destacando-se mais o modelo quadrático que conseguiu obter bons resultados para dois dos operadores de crossover. Para facilitar a comparação entre as estratégias e os modelos, em termos de velocidade. A tabela 5.16 apresenta os dados considerados mais significativos relativamente a este estudo, ao apresentar as melhores soluções obtidas e em que gerações estas foram obtidas.

Tabela 5.16: Comparação da velocidade entre as duas estratégias, para o exemplo da viga de Gusset.

Estratégia (secção 4.2.1)	Modelo	Crossover	Solução	Gerações
1	Linear	U	3214,375	719
	Quadrático	SP	3214,375	1495
	Quadrático	MP	3214,375	1479
	Quadrático	U	3214,375	421
	Sinusoidal	SP	3214,375	4159
	Sinusoidal	MP	3214,375	476
2	Linear	MP	3214,375	512
	Quadrático	SP	3214,375	1146
	Quadrático	MP	3214,375	1495
	Sinusoidal	MP	3214,375	4159

Como é possível notar, o crossover U apresenta algumas das soluções mais rápidas, mas o operador MP foi o que obteve mais soluções. Também possível testemunhar, que a primeira estratégia se provou mais dominante, neste exemplo, apresentando as soluções mais rápidas. Esta tabela também reforça, a excelente sinergia entre a segunda estratégia e o operador de crossover MP.

Focando agora na mutação taxonómica e nas suas dificuldades em obter resultados neste exemplo. A Figura 5.19, apresenta um exemplo, onde foi usado codificação binária para a utilização dos operadores de crossover, que criou maus resultados. Como é possível ver, a mutação taxonómica quando é utilizada com a codificação binária de soluções, inicialmente, gera valores extremamente elevados. Isto é, provavelmente, resultado do número de dígitos para cada variável do vetor binário, o que pode fazer com que os valores gerados violem as restrições, e sejam punidos fortemente. Isto em adição ao facto da mutação taxonómica usar informação da população corrente na mutação das soluções, faz com que o algoritmo se torne extremamente mais lento, tendo, assim, enormes dificuldades em encontrar o ponto óptimo global, no limite de gerações. Na verdade, o facto da mutação taxonómica conseguir obter soluções aceitáveis, repetidamente, quando estava sujeita a este tipo de contratempos, só serve para provar o seu poder de procura exata.

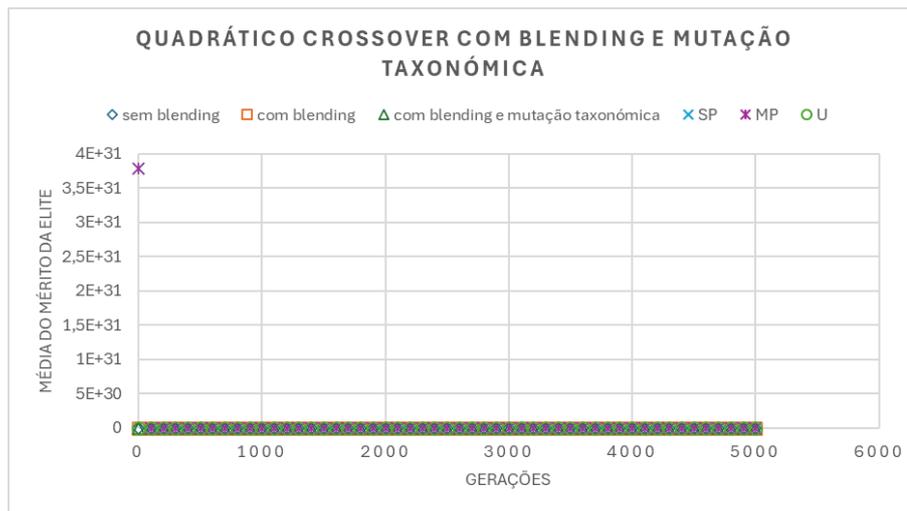


Figura 5.19: Exemplo da evolução da média do mérito da elite, para um teste com codificador binário e mutação taxonômica.

5.5 Otimização de um sistema de molas e pesos

A Figura 5.20 apresenta um sistema de molas, que suporta um conjunto de pesos, nos seus nós entre as molas. O objetivo será determinar a posição de equilíbrio do sistema, minimizando a energia potencial. O sistema terá 6 molas e 5 pesos, tal como apresentado na Figura 5.20, para a posição inicial e posição deformada.

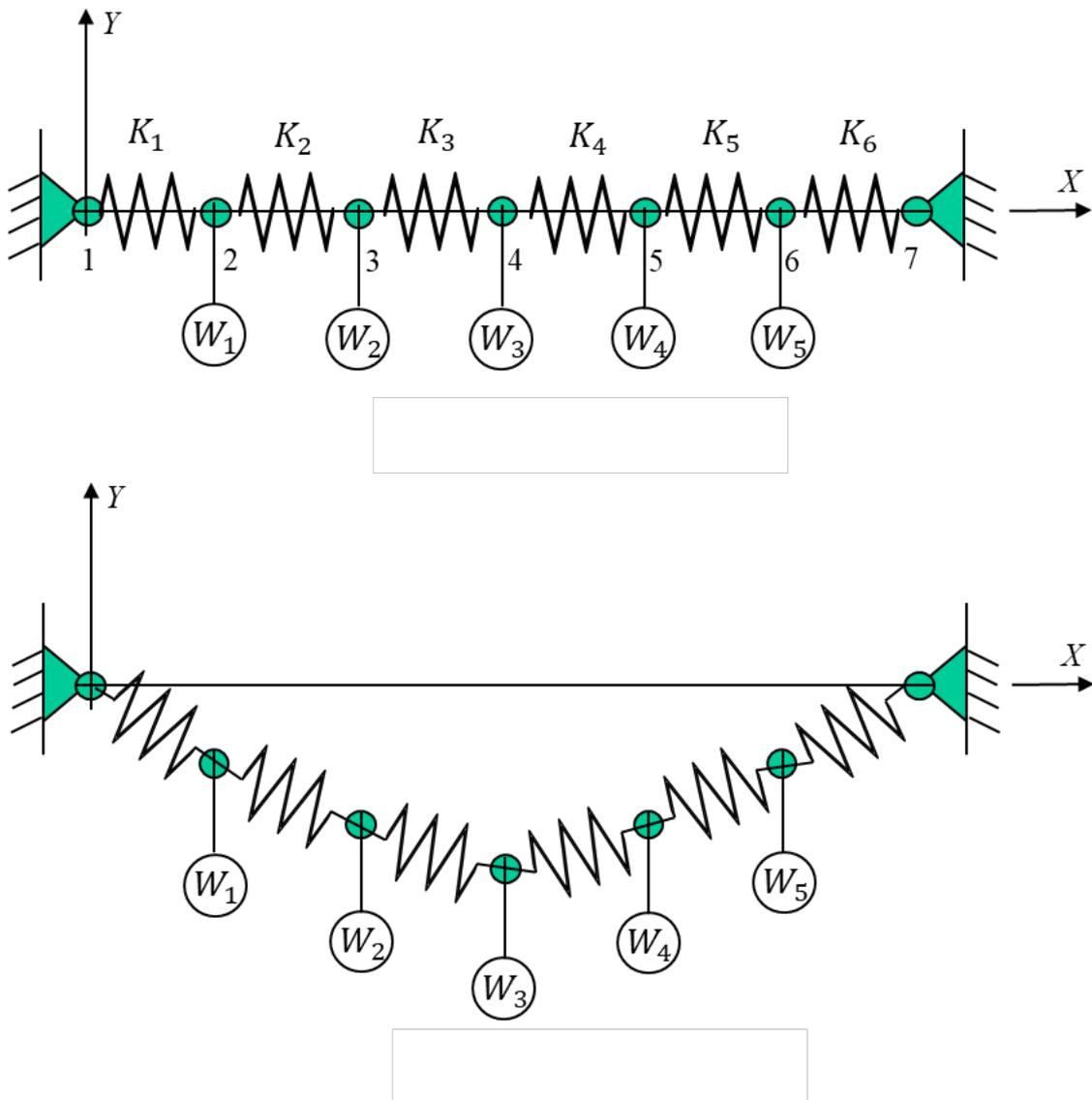


Figura 5.20: Sistema de molas na posição inicial e na posição deformada.

A deformação das molas é dada, pela seguinte fórmula,

$$\Delta L_i = [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{\frac{1}{2}} - L_i^0 \quad (5.46)$$

onde $L_i^0 = 10\text{m}$. Sendo que existem $N + 1$ molas, onde $N = 5$ é o número de pesos, a rigidez de cada mola é,

$$K_i = 500 + 200 \cdot \left(\frac{N}{3} - i\right)^2 \quad (5.47)$$

O peso aplicado a cada j nó é,

$$W_j = 50j \quad (5.48)$$

Finalmente, a energia potencial (PE) do sistema é,

$$PE = \sum_{i=1}^{N+1} \frac{1}{2} K_i \cdot (\Delta L_i)^2 + \sum_{j=1}^N W_j \cdot y_{j+1} \quad (5.49)$$

As variáveis do problema serão as posições dos nós no estado deformado, assim existirão 10 variáveis e são, x_i e y_i , com $i = 2, \dots, 6$. A função de mérito utilizada será a função objetivo (5.49) e as únicas restrições serão as restrições de dimensão, que tentarão garantir que os nós não interfiram um com o outro de uma forma fisicamente impossível. Este exemplo foi retirado de um livro [71], que apresenta várias soluções ótimas. Por isso, será utilizado como valor de referência ótimo o seguinte valor.

$$f(\mathbf{x}^*) = -4416 \quad (5.50)$$

5.5.1 Resultados

Os testes apresentados nesta seção, tentarão ilustrar a variação do comportamento do algoritmo com os diferentes operadores e modelos. Tal como, foi apresentado anteriormente, estes testes foram todos realizados com o mesmo critério de paragem, 5000 gerações.

Testes com Migração e *Blending*

As melhores soluções, obtidas pelo algoritmo, estão apresentadas nas Tabelas 5.17-5.20. Nota-se, que o algoritmo não atingiu a solução ótima de referência, nestes testes. As soluções mais próximas foram geradas pelos modelos de migração quando era utilizado a estratégia de *blending* e a mutação normal. De forma geral, este foi o modelo que obteve melhores resultados. O modelo trapezoidal, pelo contrário, apresentou os piores resultados. Os outros dois modelos demonstraram um comportamento semelhante, com o modelo quadrático a obter os melhores resultados entre os dois.

Tabela 5.17: Melhores soluções usando o modelo linear, para o exemplo das molas e pesos.

Modelo	Linear			
<i>Blending</i> :	Não	Não	Sim	Sim
Mutação:	Normal	Taxonómica	Normal	Taxonómica
x_1	10,661	9,693	10,687	10,942
x_2	21,662	20,007	21,898	21,509
x_3	32,472	32,468	32,099	31,957
x_4	42,576	41,354	42,324	41,636
x_5	52,798	50,033	51,633	51,858
x_6	-1,074	-5,157	-4,513	-4,924
x_7	-6,640	-5,452	-8,426	-8,514
x_8	-8,149	-8,277	-11,010	-9,782
x_9	-8,245	-4,353	-9,361	-9,055
x_{10}	-6,338	-1,409	-5,726	-6,257
Objetivo	-2672,011	1365,146	-4092,193	-3584,690

Tabela 5.18: Melhores soluções usando o modelo trapezoidal, para o exemplo das molas e pesos.

Modelo	Trapezoidal			
<i>Blending</i> :	Não	Não	Sim	Sim
Mutação:	Normal	Taxonómica	Normal	Taxonómica
x_1	10,661	9,693	10,210	11,140
x_2	21,662	20,007	21,019	23,525
x_3	32,472	32,468	30,616	32,636
x_4	42,576	41,354	41,503	42,185
x_5	52,798	50,033	50,794	51,121
x_6	-1,074	-5,157	-4,930	-4,312
x_7	-6,640	-5,452	-7,047	-7,116
x_8	-8,149	-8,277	-8,249	-5,870
x_9	-8,245	-4,353	-9,860	-8,093
x_{10}	-6,338	-1,409	-5,130	-4,198
Objetivo	-2672,011	1365,146	-2933,314	-998,124

Tabela 5.19: Melhores soluções usando o modelo quadrático, para o exemplo das molas e pesos.

Modelo	Quadrático			
<i>Blending</i> :	Não	Não	Sim	Sim
Mutação:	Normal	Taxonómica	Normal	Taxonómica
x_1	10,154	9,693	11,114	9,623
x_2	21,185	20,007	21,928	21,124
x_3	31,923	32,468	32,299	31,305
x_4	42,058	41,354	42,538	41,743
x_5	52,167	50,033	52,181	51,953
x_6	-2,759	-5,157	-5,259	-6,019
x_7	-3,516	-5,452	-8,705	-9,242
x_8	-3,314	-8,277	-9,434	-10,971
x_9	-6,436	-4,353	-8,325	-9,279
x_{10}	-4,824	-1,409	-6,790	-6,374
Objetivo	-1142,608	1365,146	-3401,534	-3823,344

Tabela 5.20: Melhores soluções usando o modelo sinusoidal, para o exemplo das molas e pesos.

Modelo	Sinusoidal			
<i>Blending</i> :	Não	Não	Sim	Sim
Mutação:	Normal	Taxonómica	Normal	Taxonómica
x_1	10,661	9,693	10,302	11,166
x_2	21,662	20,007	20,590	22,290
x_3	32,472	32,468	31,840	32,088
x_4	42,576	41,354	42,206	41,771
x_5	52,798	50,033	51,044	51,926
x_6	-1,074	-5,157	-4,984	-4,698
x_7	-6,640	-5,452	-10,580	-8,648
x_8	-8,149	-8,277	-10,518	-11,007
x_9	-8,245	-4,353	-10,631	-8,776
x_{10}	-6,338	-1,409	-4,953	-5,549
Objetivo	-2672,011	1365,146	-3631,474	-3050,272

Em termos da mutação, a mutação taxonómica, continuou a apresentar os piores resultados quando o *blending* não é aplicado, mas apresentou resultados excelentes quando é aplicado. Isto já foi experienciado nos exemplos anteriores, e como havia sido dito, a utilização da mutação taxonómica tende a tornar o algoritmo um pouco mais lento, o que se sente bastante sem a utilização do *blending*. As Figuras 5.21-5.24 representam a evolu-

ção do mérito médio da elite com o número das gerações. Mais uma vez, é possível notar a tendência do algoritmo de ficar preso num mesmo ponto, que não o ótimo, por várias gerações e a rápida evolução do algoritmo.

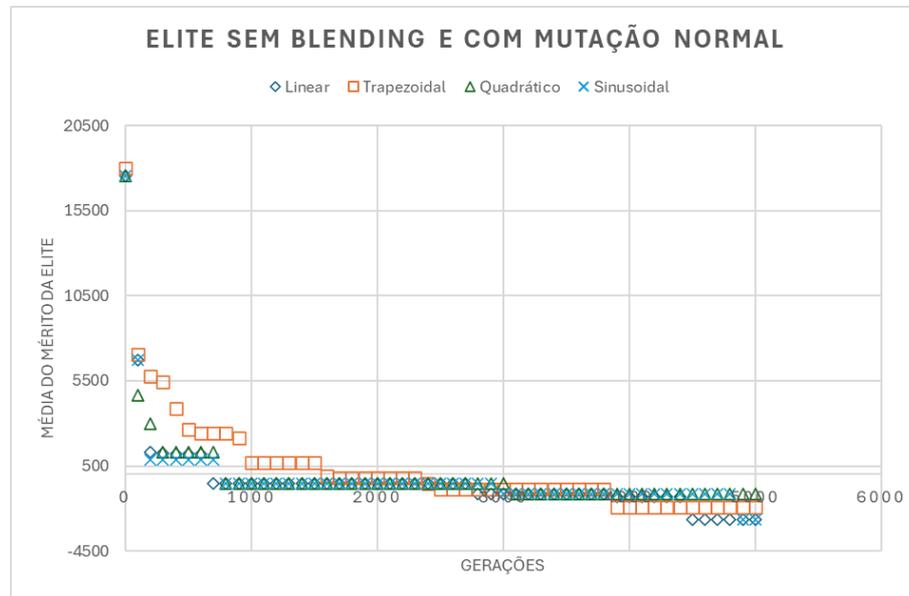


Figura 5.21: Evolução da média do mérito da elite ao longo das gerações, para o exemplo das molas e pesos, usando a mutação normal sem *blending*.

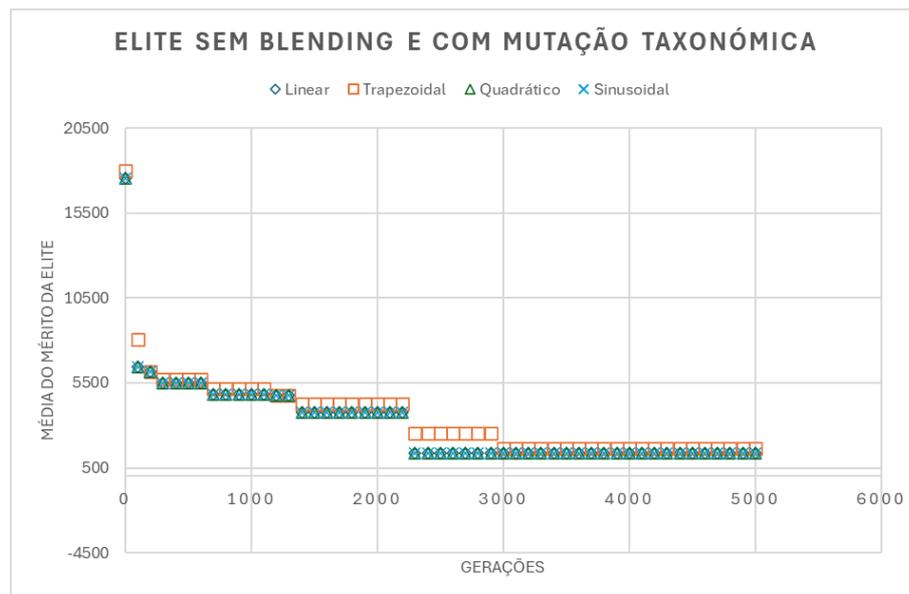


Figura 5.22: Evolução da média do mérito da elite ao longo das gerações, para o exemplo das molas e pesos, usando a mutação taxonômica sem *blending*.

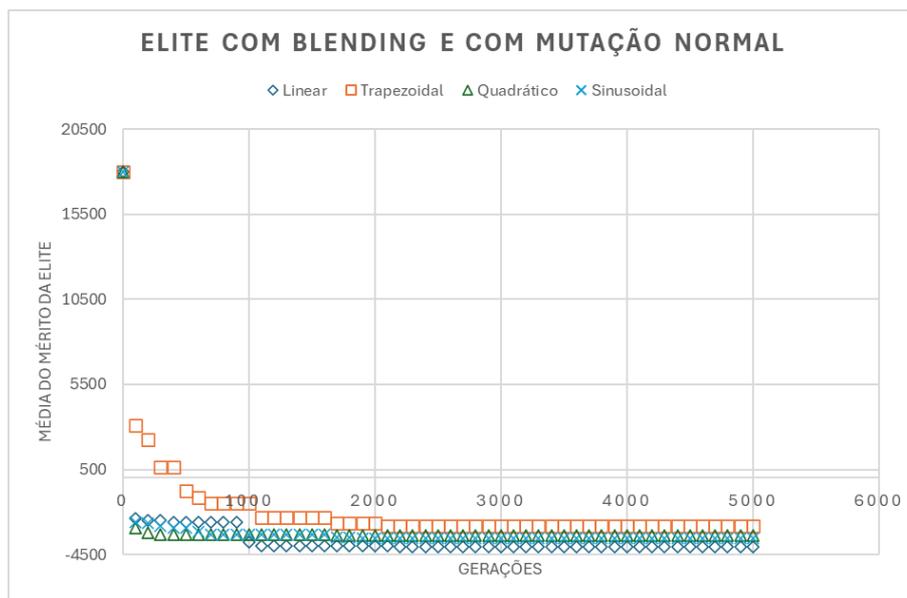


Figura 5.23: Evolução da média do mérito da elite ao longo das gerações, para o exemplo das molas e pesos, usando *blending* e a mutação normal.

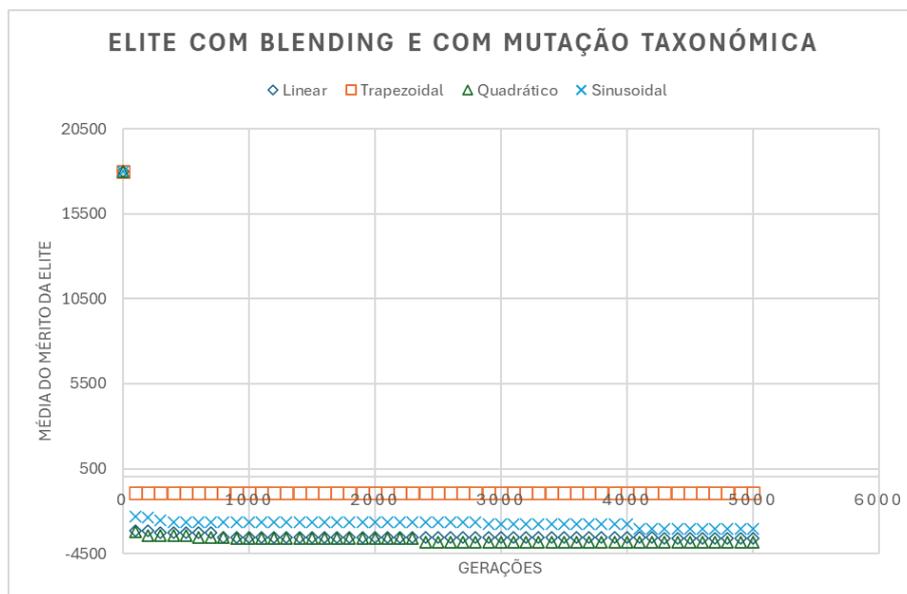


Figura 5.24: Evolução da média do mérito da elite ao longo das gerações, para o exemplo das molas e pesos, usando *blending* e a mutação taxonômica.

Testes com Operadores de Crossover

Mantiveram-se os parâmetros de testes dos testes anteriores, como o critério de paragem, colocado a 5000 gerações, acrescentado apenas os operadores de crossover e a codificação binária de soluções. Os resultados obtidos, relativamente às melhores soluções da primeira estratégia (secção 4.2.1), estão apresentados nas Tabelas 5.21-5.24. Devido aos maus resultados obtidos usando a mutação taxonômica, estes não serão apresentados.

Tabela 5.21: Melhores soluções usando o modelo linear com operadores de crossover, com a primeira estratégia (secção 4.2.1), para o exemplo das molas e pesos.

Modelo	Linear					
<i>Blending</i>	Não	Não	Não	Sim	Sim	Sim
Mutação	Normal	Normal	Normal	Normal	Normal	Normal
Crossover	SP	MP	U	SP	MP	U
x_1	10,661	10,661	10,661	10,138	10,139	10,135
x_2	21,662	21,662	21,662	21,365	21,419	21,306
x_3	32,472	32,472	32,472	31,552	31,503	31,457
x_4	42,576	42,576	42,576	41,634	41,642	41,671
x_5	52,798	52,798	52,798	51,231	51,367	51,271
x_6	-1,074	-1,074	-1,074	-4,565	-4,853	-4,202
x_7	-6,640	-6,640	-6,640	-6,713	-7,092	-6,783
x_8	-8,149	-8,149	-8,149	-8,980	-9,102	-9,186
x_9	-8,245	-8,245	-8,245	-8,843	-8,542	-8,542
x_{10}	-6,338	-6,338	-6,338	-5,249	-5,283	-5,255
Objetivo	-2672,01	-2672,01	-2672,01	-4151,46	-4132,08	-4221,53

Tabela 5.22: Melhores soluções usando o modelo trapezoidal com operadores de crossover, com a primeira estratégia (secção 4.2.1), para o exemplo das molas e pesos.

Modelo	Trapezoidal					
<i>Blending</i>	Não	Não	Não	Sim	Sim	Sim
Mutação	Normal	Normal	Normal	Normal	Normal	Normal
Crossover	SP	MP	U	SP	MP	U
x_1	10,154	10,661	10,661	9,175	9,371	9,215
x_2	21,185	21,662	21,662	20,519	20,907	21,025
x_3	31,923	32,472	32,472	30,992	31,174	31,511
x_4	42,058	42,576	42,576	40,771	41,536	42,021
x_5	52,167	52,798	52,798	50,707	51,110	51,924
x_6	-2,759	-1,074	-1,074	-5,392	-5,462	-5,769
x_7	-3,516	-6,640	-6,640	-6,997	-5,511	-7,357
x_8	-3,314	-8,149	-8,149	-9,103	-8,452	-8,913
x_9	-6,436	-8,245	-8,245	-6,488	-7,439	-6,924
x_{10}	-4,824	-6,338	-6,338	-4,038	-5,227	-6,041
Objetivo	-1142,61	-2672,01	-2672,01	-3643,45	-3476,41	-3513,4

Tabela 5.23: Melhores soluções usando o modelo quadrático com operadores de crossover, com a primeira estratégia (secção 4.2.1), para o exemplo das molas e pesos.

Modelo	Quadrático					
	Não	Não	Não	Sim	Sim	Sim
Mutação	Normal	Normal	Normal	Normal	Normal	Normal
Crossover	SP	MP	U	SP	MP	U
x_1	10,154	10,154	10,154	10,114	10,117	10,119
x_2	21,185	21,185	21,185	20,925	20,924	20,925
x_3	31,923	31,923	31,923	31,145	31,146	31,149
x_4	42,058	42,058	42,058	41,277	41,275	41,276
x_5	52,167	52,167	52,167	50,749	50,748	50,754
x_6	-2,759	-2,759	-2,759	-5,626	-5,630	-5,638
x_7	-3,516	-3,516	-3,516	-9,752	-9,749	-9,753
x_8	-3,314	-3,314	-3,314	-11,695	-11,689	-11,683
x_9	-6,436	-6,436	-6,436	-9,288	-9,286	-9,287
x_{10}	-4,824	-4,824	-4,824	-4,806	-4,798	-4,813
Objetivo	-1142,61	-1142,61	-1142,61	-3794,05	-3794,22	-3795,69

Tabela 5.24: Melhores soluções usando o modelo sinusoidal com operadores de crossover, com a primeira estratégia (secção 4.2.1), para o exemplo das molas e pesos.

Modelo	Sinusoidal					
	Não	Não	Não	Sim	Sim	Sim
Mutação	Normal	Normal	Normal	Normal	Normal	Normal
Crossover	SP	MP	U	SP	MP	U
x_1	10,661	10,661	10,661	10,286	10,215	10,199
x_2	21,662	21,662	21,662	21,075	21,083	21,085
x_3	32,472	32,472	32,472	31,191	31,193	31,195
x_4	42,576	42,576	42,576	41,179	41,176	41,177
x_5	52,798	52,798	52,798	51,088	51,081	51,079
x_6	-1,074	-1,074	52,798	-5,085	-5,093	-5,090
x_7	-6,640	-6,640	-1,074	-8,319	-8,328	-8,319
x_8	-8,149	-8,149	-6,640	-10,341	-10,352	-10,351
x_9	-8,245	-8,245	-8,149	-8,224	-8,215	-8,220
x_{10}	-6,338	-6,338	-6,338	-5,192	-5,183	-5,191
Objetivo	-2672,01	-2672,01	-2672,01	-4057,12	-4058,82	-4059,01

De imediato, repara-se que as soluções melhoraram, com a inclusão do operador de crossover, mas continuam longe do valor de referência. Nesta iteração, destaca-se o modelo sinusoidal, que conseguiu melhorar a sua solução significativamente com o operador, e o

modelo linear, que continuou a apresentar as melhores soluções. Em termos de operador de crossover, o operador Uniform (U) apresenta os melhores resultados, na maioria dos modelos. Em relação à segunda estratégia (secção 4.2.1), os resultados, obtidos estão expostos nas Tabelas 5.25-5.28.

Tabela 5.25: Melhores soluções usando o modelo linear com operadores de crossover, com a segunda estratégia (secção 4.2.1), para o exemplo das molas e pesos.

Modelo	Linear					
	Não	Não	Não	Sim	Sim	Sim
Blending	Normal	Normal	Normal	Normal	Normal	Normal
Mutação	Normal	Normal	Normal	Normal	Normal	Normal
Crossover	SP	MP	U	SP	MP	U
x_1	9,652	10,661	10,661	10,135	10,138	9,896
x_2	20,573	21,662	21,662	21,499	21,365	21,289
x_3	32,017	32,472	32,472	31,475	31,552	31,571
x_4	41,389	42,576	42,576	41,402	41,634	41,321
x_5	52,089	52,798	52,798	51,269	51,231	51,221
x_6	-5,497	-1,074	-1,074	-5,333	-4,565	-5,951
x_7	-9,322	-6,640	-6,640	-7,634	-6,713	-9,003
x_8	-12,772	-8,149	-8,149	-9,362	-8,980	-10,882
x_9	-9,698	-8,245	-8,245	-8,442	-8,843	-9,668
x_{10}	-5,889	-6,338	-6,338	-5,449	-5,249	-5,674
Objetivo	-1335,97	-2672,01	-2672,01	-3879,82	-4151,46	-3467,18

Tabela 5.26: Melhores soluções usando o modelo trapezoidal com operadores de crossover, com a segunda estratégia (secção 4.2.1), para o exemplo das molas e pesos.

Modelo	Trapezoidal					
<i>Blending</i>	Não	Não	Não	Sim	Sim	Sim
Mutação	Normal	Normal	Normal	Normal	Normal	Normal
Crossover	SP	MP	U	SP	MP	U
x_1	10,661	10,154	10,154	9,218	9,175	9,008
x_2	21,662	21,185	21,185	20,928	20,519	21,049
x_3	32,472	31,923	31,923	31,426	30,992	31,502
x_4	42,576	42,058	42,058	41,537	40,771	41,213
x_5	52,798	52,167	52,167	51,703	50,707	51,134
x_6	-1,074	-2,759	-2,759	-5,330	-5,392	-5,756
x_7	-6,640	-3,516	-3,516	-7,306	-6,997	-7,656
x_8	-8,149	-3,314	-3,314	-8,091	-9,103	-8,603
x_9	-8,245	-6,436	-6,436	-8,068	-6,488	-7,314
x_{10}	-6,338	-4,824	-4,824	-6,171	-4,038	-5,132
Objetivo	-2672,01	-1142,61	-1142,61	-3791,27	-3643,45	-3397,73

Tabela 5.27: Melhores soluções usando o modelo quadrático com operadores de crossover, com a segunda estratégia (secção 4.2.1), para o exemplo das molas e pesos.

Modelo	Quadrático					
<i>Blending</i>	Não	Não	Não	Sim	Sim	Sim
Mutação	Normal	Normal	Normal	Normal	Normal	Normal
Crossover	SP	MP	U	SP	MP	U
x_1	10,154	10,154	10,154	10,118	10,114	10,115
x_2	21,185	21,185	21,185	20,925	20,925	20,925
x_3	31,923	31,923	31,923	31,146	31,145	31,143
x_4	42,058	42,058	42,058	41,276	41,277	41,276
x_5	52,167	52,167	52,167	50,745	50,749	50,756
x_6	-2,759	-2,759	-2,759	-5,648	-5,626	-5,638
x_7	-3,516	-3,516	-3,516	-9,747	-9,752	-9,750
x_8	-3,314	-3,314	-3,314	-11,685	-11,695	-11,692
x_9	-6,436	-6,436	-6,436	-9,284	-9,288	-9,286
x_{10}	-4,824	-4,824	-4,824	-4,808	-4,806	-4,815
Objetivo	-1142,61	-1142,61	-1142,61	-3791,24	-3794,05	-3794,97

Tabela 5.28: Melhores soluções usando o modelo sinusoidal com operadores de crossover, com a segunda estratégia (secção 4.2.1), para o exemplo das molas e pesos.

Modelo	Sinusoidal					
	Não	Não	Não	Sim	Sim	Sim
Mutação	Normal	Normal	Normal	Normal	Normal	Normal
Crossover	SP	MP	U	SP	MP	U
x_1	10,661	10,661	10,661	10,220	10,286	10,238
x_2	21,662	21,662	21,662	21,084	21,075	21,085
x_3	32,472	32,472	32,472	31,188	31,191	31,191
x_4	42,576	42,576	42,576	41,176	41,179	41,179
x_5	52,798	52,798	52,798	51,085	51,088	51,089
x_6	-1,074	-1,074	-1,074	-5,099	-5,085	-5,092
x_7	-6,640	-6,640	-6,640	-8,325	-8,319	-8,321
x_8	-8,149	-8,149	-8,149	-10,361	-10,341	-10,348
x_9	-8,245	-8,245	-8,245	-8,208	-8,224	-8,196
x_{10}	-6,338	-6,338	-6,338	-5,172	-5,192	-5,183
Objetivo	-2672,01	-2672,01	-2672,01	-4059,73	-4057,12	-4059,74

Desta vez, os resultados pioraram. de forma geral, quando comparados com os da primeira estratégia. Em termos de operador de crossover, o operador MP, apresenta os melhores resultados para esta estratégia, algo que se tem repetido entre exemplos. O modelo sinusoidal, conseguiu manter as suas soluções na gama de valores obtidos anteriormente, sendo assim este o modelo de destaque. Realça-se também o modelo linear, que teve um desempenho significativamente pior, chegando a algumas soluções piores que as soluções obtidas nos testes sem o operador de crossover.

Continuando a focar na mutação taxonómica e na sua má sinergia com a codificação binária de soluções. Os resultados obtidos, mesmo quando aplicado a migração com *blending*, algo que tem sido útil a tentar evitar este problema, ficaram bastante aquém dos outros resultados, para este teste. Como foi mencionado no exemplo anterior, a mutação taxonómica apresenta uma enorme dificuldade em coexistir com a codificação binária de soluções, quando esta tem de trabalhar entre gamas de valores muito extremas, que resultam num elevado número de dígitos. Por isso, como as soluções iniciais se dispersam imenso, a mutação taxonómica torna o algoritmo extremamente mais lento, não conseguindo chegar a boas soluções dentro dos limites estabelecidos.

Testes com otimizador Local (Hooke-Jeeves)

Continuando com os mesmos parâmetros iniciais e adicionando o otimizador local, é esperado que os resultados melhorem em todos os setores. Os resultados apresentados nas tabelas 5.29-5.30, apresentam as melhores soluções obtidas, para o teste do otimizador local sem a utilização dos operadores de crossover.

Tabela 5.29: Melhores soluções usando os modelos linear e trapezoidal com otimizador local, para o exemplo das molas e pesos.

Modelo	Linear			Trapezoidal		
	Não	Sim	Sim	Não	Sim	Sim
Mutação	Normal	Normal	Taxonómica	Normal	Normal	Taxonómica
x_1	10,661	10,930	10,363	10,154	10,331	10,260
x_2	21,662	21,547	20,989	21,185	21,037	20,769
x_3	32,472	31,661	31,741	31,923	31,642	31,299
x_4	42,576	41,782	41,606	42,058	42,043	41,583
x_5	52,798	51,442	52,014	52,167	51,727	51,383
x_6	-1,074	-3,490	-4,154	-2,759	-4,289	-4,159
x_7	-6,640	-7,311	-9,098	-3,516	-7,907	-7,885
x_8	-8,149	-10,505	-10,760	-3,314	-9,817	-9,418
x_9	-8,245	-8,945	-8,395	-6,436	-9,330	-8,652
x_{10}	-6,338	-5,561	-6,235	-4,824	-5,954	-5,501
Objetivo	-2672,01	-4221,75	-3873,36	-1142,61	-4415,8	-4354,79

Tabela 5.30: Melhores soluções usando os modelos quadrático e sinusoidal com otimizador local, para o exemplo das molas e pesos.

Modelo	Quadrático			Sinusoidal		
	Não	Sim	Sim	Não	Sim	Sim
Mutação	Normal	Normal	Taxonómica	Normal	Normal	Taxonómica
x_1	10,661	10,667	10,150	10,661	10,460	11,207
x_2	21,662	21,738	21,176	21,662	22,095	21,368
x_3	32,472	31,239	31,797	32,472	31,603	32,085
x_4	42,576	41,669	42,104	42,576	41,640	42,114
x_5	52,798	51,403	52,068	52,798	50,962	51,852
x_6	-1,074	-4,145	-4,669	-1,074	-5,584	-3,648
x_7	-6,640	-6,214	-7,761	-6,640	-6,962	-7,362
x_8	-8,149	-10,090	-9,252	-8,149	-10,473	-7,424
x_9	-8,245	-9,239	-7,960	-8,245	-8,966	-8,470
x_{10}	-6,338	-5,813	-5,894	-6,338	-4,787	-5,840
Objetivo	-2672,01	-3969,49	-4091,68	-2672,01	-3552,7	-3834,83

Apenas com o otimizador local, os resultados aparentam ser mais promissores. Existindo até resultados muito próximos do valor óptimo de referência. Destaca-se, pela positiva, o modelo trapezoidal, que obteve a melhoria mais significativa e os melhores resultados deste teste, e, pela negativa, o modelo sinusoidal, que conseguiu piorar os seus resultados. Mais uma vez, a mutação taxonómica sem *blending*, apresentou maus resultados, daí não

estarem aqui apresentados. As Figuras 5.25-5.27 representam, graficamente, a evolução da média da elite pelas gerações do algoritmo.

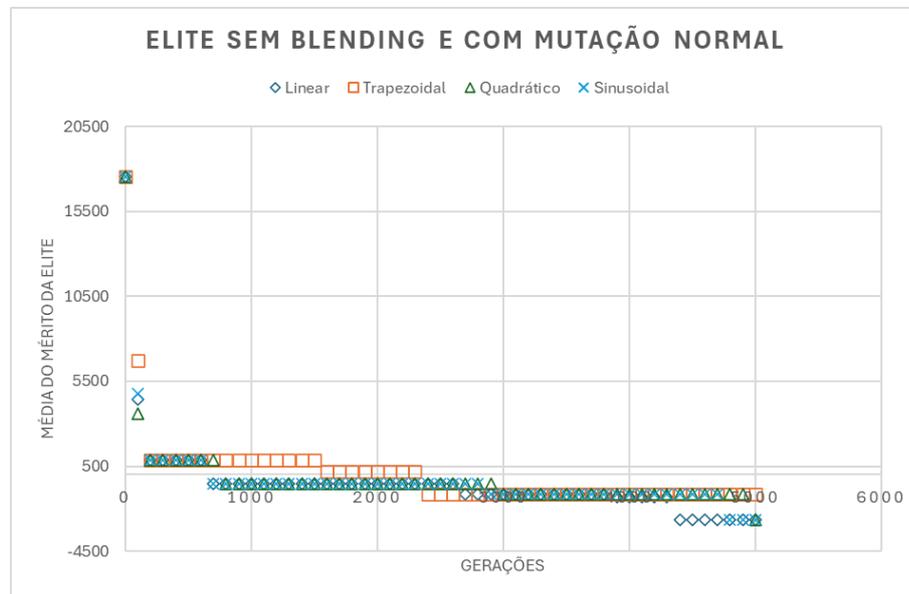


Figura 5.25: Evolução da média do mérito da elite pelas gerações, para o exemplo das molas e pesos, usando o otimizador local, com a mutação normal e sem *blending*.

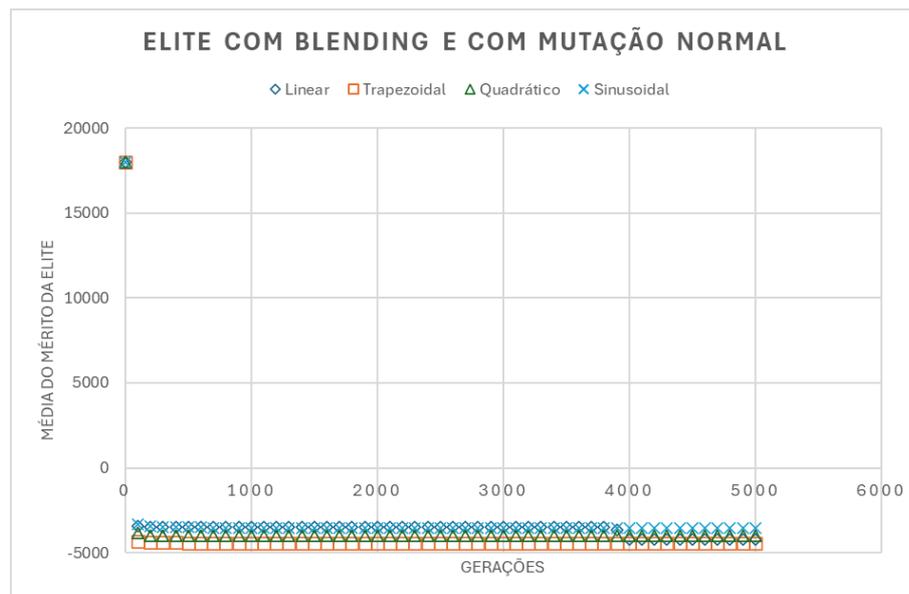


Figura 5.26: Evolução da média do mérito da elite pelas gerações, para o exemplo das molas e pesos, usando o otimizador local e *blending* com a mutação normal.

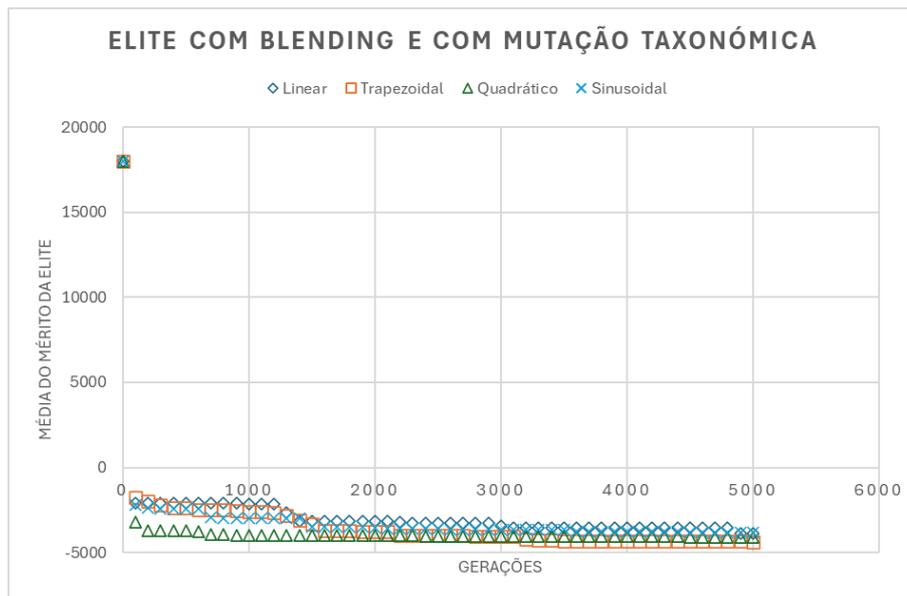


Figura 5.27: Evolução da média do mérito da elite pelas gerações, para o exemplo das molas e pesos, usando o otimizador local, mutação taxonômica e *blending*.

A Tabela 5.31, apresenta as melhores soluções obtidas para o teste com o crossover, usando a primeira estratégia (secção 4.2.1), e o otimizador local.

Tabela 5.31: Melhores soluções usando o otimizador local com operadores de crossover, com a primeira estratégia (secção 4.2.1), para o exemplo das molas e pesos.

Modelo	Blending	Mutação	Crossover	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	Objetivo
Linear	Não	Normal	SP	10,661	21,662	32,472	42,576	52,798	-1,074	-6,640	-8,149	-8,245	-6,338	-2672,01
	Não	Normal	MP	10,661	21,662	32,472	42,576	52,798	-1,074	-6,640	-8,149	-8,245	-6,338	-2672,01
	Não	Normal	U	10,661	21,662	32,472	42,576	52,798	-1,074	-6,640	-8,149	-8,245	-6,338	-2672,01
	Sim	Normal	SP	10,331	21,041	31,633	42,026	51,699	-4,285	-7,898	-9,825	-9,304	-5,906	-4415,18
	Sim	Normal	MP	10,303	20,985	31,611	42,019	51,706	-4,319	-7,970	-9,840	-9,305	-5,925	-4414,51
	Sim	Normal	U	10,315	21,012	31,634	42,042	51,732	-4,315	-7,956	-9,849	-9,335	-5,963	-4415,42
	Sim	Taxonómica	SP	10,415	21,274	31,222	41,081	51,172	-0,593	-0,212	-4,148	-6,981	-5,096	-2655,75
	Sim	Taxonómica	MP	10,645	21,310	30,486	40,678	50,679	-0,897	-0,997	-6,018	-6,595	-4,247	-2823,18
	Sim	Taxonómica	U	9,888	20,404	30,863	41,065	50,302	-4,098	-6,662	-6,739	-7,565	-3,317	-3596,18
Trapezoidal	Não	Normal	SP	11,768	22,538	33,738	43,868	53,219	-2,226	-3,393	-6,936	-4,160	-8,210	-591,992
	Não	Normal	MP	11,768	22,538	33,738	43,868	53,219	-2,226	-3,393	-6,936	-4,160	-8,210	-591,992
	Não	Normal	U	10,154	21,185	31,923	42,058	52,167	-2,759	-3,516	-3,314	-6,436	-4,824	-1142,61
	Sim	Normal	SP	10,306	21,032	31,653	42,059	51,747	-4,356	-7,942	-9,819	-9,349	-5,980	-4415,67
	Sim	Normal	MP	10,140	20,690	31,398	41,828	51,581	-4,402	-8,035	-9,552	-8,986	-5,761	-4387,48
	Sim	Normal	U	10,227	20,981	31,604	42,008	51,729	-4,317	-7,776	-9,512	-9,206	-5,950	-4408,81
	Sim	Taxonómica	SP	9,466	20,044	30,766	41,020	51,111	-3,973	-7,708	-9,126	-7,741	-5,185	-3985,7
	Sim	Taxonómica	MP	9,802	20,056	30,523	40,046	50,297	-4,188	-7,959	-9,255	-5,551	-3,481	-3368,31
	Sim	Taxonómica	U	10,806	22,068	32,375	42,744	52,527	-3,949	-7,178	-9,944	-9,835	-6,814	-4235,29
Quadrático	Não	Normal	SP	10,154	21,185	31,923	42,058	52,167	-2,759	-3,516	-3,314	-6,436	-4,824	-1142,61
	Não	Normal	MP	10,154	21,185	31,923	42,058	52,167	-2,759	-3,516	-3,314	-6,436	-4,824	-1142,61
	Não	Normal	U	10,154	21,185	31,923	42,058	52,167	-2,759	-3,516	-3,314	-6,436	-4,824	-1142,61
	Sim	Normal	SP	10,347	21,076	31,679	42,082	51,764	-4,292	-7,914	-9,860	-9,390	-6,003	-4416,36
	Sim	Normal	MP	10,318	21,039	31,639	42,037	51,741	-4,306	-7,906	-9,815	-9,283	-5,969	-4415,55
	Sim	Normal	U	10,148	20,972	31,623	42,021	51,715	-4,912	-8,274	-9,983	-9,307	-5,941	-4394,94
	Sim	Taxonómica	SP	9,939	19,855	30,038	40,173	49,684	-3,061	-6,516	-5,159	-5,009	-1,828	-2393,95
	Sim	Taxonómica	MP	10,353	21,227	30,899	40,862	50,243	-2,968	-6,119	-8,874	-6,263	-2,582	-3260,09
	Sim	Taxonómica	U	10,154	20,064	30,173	40,382	50,267	-3,277	-6,079	-7,265	-5,678	-2,896	-3315,33
Simusoidal	Não	Normal	SP	10,661	21,662	32,472	42,576	52,798	-1,074	-6,640	-8,149	-8,245	-6,338	-2672,01
	Não	Normal	MP	10,661	21,662	32,472	42,576	52,798	-1,074	-6,640	-8,149	-8,245	-6,338	-2672,01
	Não	Normal	U	10,661	21,662	32,472	42,576	52,798	-1,074	-6,640	-8,149	-8,245	-6,338	-2672,01
	Sim	Normal	SP	10,349	21,074	31,680	42,082	51,762	-4,302	-7,944	-9,874	-9,390	-6,000	-4416,29
	Sim	Normal	MP	10,317	21,064	31,669	42,073	51,756	-4,326	-7,936	-9,903	-9,380	-5,994	-4415,95
	Sim	Normal	U	10,326	21,032	31,623	42,021	51,680	-4,295	-7,910	-9,840	-9,320	-5,878	-4414,65
	Sim	Taxonómica	SP	10,257	20,862	31,337	40,485	50,499	-3,321	-5,749	-6,406	-1,911	-3,466	-2382,48
	Sim	Taxonómica	MP	10,813	21,227	30,698	41,064	50,806	1,907	-2,052	-6,492	-7,273	-4,459	-2687,21
	Sim	Taxonómica	U	11,257	21,895	31,066	40,698	50,884	2,241	4,982	-0,474	-4,227	-4,402	-147,1

Os resultados melhoraram significativamente, com a introdução do crossover, havendo algumas soluções que chegaram ao valor óptimo de referência. Realça-se o crossover SP, que obteve algumas das melhores soluções e o crossover MP, que conseguiu também obter vários resultados positivos. Em relação aos modelos, todos conseguiram obter bons resultados, quando utilizados com o *blending*. A mutação taxonómica, conseguiu produzir alguns bons resultados, mas a mutação normal continuou a ser superior, em termos de velocidade. Os valores obtidos para a segunda estratégia (secção 4.2.1) estão apresentados na Tabela 5.32.

Tabela 5.32: Melhores soluções usando o otimizador local com operadores de crossover, com a segunda estratégia (secção 4.2.1), para o exemplo das molas e pesos.

Modelo	Blending	Mutação	Crossover	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	Objetivo
Linear	Não	Normal	SP	10,661	21,662	32,472	42,576	52,798	-1,074	-6,640	-8,149	-8,245	-6,338	-2672,01
	Não	Normal	MP	10,661	21,662	32,472	42,576	52,798	-1,074	-6,640	-8,149	-8,245	-6,338	-2672,01
	Não	Normal	U	10,661	21,662	32,472	42,576	52,798	-1,074	-6,640	-8,149	-8,245	-6,338	-2672,01
	Sim	Normal	SP	10,337	21,065	31,665	42,068	51,758	-4,294	-7,889	-9,832	-9,356	-5,995	-4416,24
	Sim	Normal	MP	10,331	21,041	31,633	42,026	51,699	-4,285	-7,898	-9,825	-9,304	-5,906	-4415,18
	Sim	Normal	U	10,330	21,063	31,667	42,067	51,756	-4,311	-7,883	-9,780	-9,350	-5,991	-4415,99
	Sim	Taxonômica	SP	10,388	20,843	30,666	40,790	50,330	-2,042	-3,527	-5,819	-6,667	-3,362	-3161,83
	Sim	Taxonômica	MP	10,415	21,274	31,222	41,081	51,172	-0,593	-0,212	-4,148	-6,981	-5,096	-2655,75
	Sim	Taxonômica	U	10,870	21,905	32,398	41,941	52,043	1,373	0,204	-1,126	-5,064	-6,531	-1673,48
Trapezoidal	Não	Normal	SP	10,154	21,185	31,923	42,058	52,167	-2,759	-3,516	-3,314	-6,436	-4,824	-1142,61
	Não	Normal	MP	11,768	22,538	33,738	43,868	53,219	-2,226	-3,393	-6,936	-4,160	-8,210	-591,992
	Não	Normal	U	10,154	21,185	31,923	42,058	52,167	-2,759	-3,516	-3,314	-6,436	-4,824	-1142,61
	Sim	Normal	SP	10,339	21,063	31,672	42,075	51,757	-4,317	-7,964	-9,877	-9,377	-5,994	-4416,17
	Sim	Normal	MP	10,306	21,032	31,653	42,059	51,747	-4,356	-7,942	-9,819	-9,349	-5,981	-4415,67
	Sim	Normal	U	10,262	20,907	31,546	41,950	51,695	-4,275	-7,835	-9,588	-9,100	-5,904	-4408,09
	Sim	Taxonômica	SP	10,887	21,619	31,176	40,847	50,720	0,601	1,713	-3,083	-6,399	-4,083	-1845,5
	Sim	Taxonômica	MP	9,466	20,044	30,766	41,020	51,111	-3,973	-7,708	-9,126	-7,741	-5,185	-3985,7
	Sim	Taxonômica	U	9,762	20,334	31,183	41,555	51,101	-5,433	-9,440	-10,187	-8,809	-4,931	-4201,11
Quadrático	Não	Normal	SP	10,154	21,185	31,923	42,058	52,167	-2,759	-3,516	-3,314	-6,436	-4,824	-1142,61
	Não	Normal	MP	10,154	21,185	31,923	42,058	52,167	-2,759	-3,516	-3,314	-6,436	-4,824	-1142,61
	Não	Normal	U	10,154	21,185	31,923	42,058	52,167	-2,759	-3,516	-3,314	-6,436	-4,824	-1142,61
	Sim	Normal	SP	10,339	21,062	31,661	42,060	51,731	-4,291	-7,906	-9,848	-9,362	-5,953	-4416,05
	Sim	Normal	MP	10,347	21,076	31,679	42,082	51,764	-4,292	-7,914	-9,860	-9,390	-6,003	-4416,36
	Sim	Normal	U	10,305	20,995	31,597	41,990	51,651	-4,337	-7,960	-9,849	-9,291	-5,845	-4413,31
	Sim	Taxonômica	SP	10,083	20,598	31,143	40,659	49,928	-3,901	-6,768	-7,254	-3,214	1,048	-1623,48
	Sim	Taxonômica	MP	9,939	19,855	30,038	40,173	49,684	-3,061	-6,516	-5,159	-5,009	-1,828	-2393,95
	Sim	Taxonômica	U	9,608	19,535	29,999	39,658	49,749	-4,512	-8,190	-7,177	-3,539	-0,831	-2195,72
Sinusoidal	Não	Normal	SP	10,661	21,662	32,472	42,576	52,798	-1,074	-6,640	-8,149	-8,245	-6,338	-2672,01
	Não	Normal	MP	10,661	21,662	32,472	42,576	52,798	-1,074	-6,640	-8,149	-8,245	-6,338	-2672,01
	Não	Normal	U	10,661	21,662	32,472	42,576	52,798	-1,074	-6,640	-8,149	-8,245	-6,338	-2672,01
	Sim	Normal	SP	10,346	21,074	31,675	42,078	51,759	-4,290	-7,910	-9,852	-9,380	-5,996	-4416,34
	Sim	Normal	MP	10,349	21,074	31,680	42,082	51,762	-4,302	-7,944	-9,874	-9,390	-6,000	-4416,29
	Sim	Normal	U	10,349	21,078	31,679	42,080	51,766	-4,286	-7,892	-9,840	-9,377	-6,005	-4416,36
	Sim	Taxonômica	SP	10,758	21,925	31,203	41,450	50,404	-2,186	-3,309	-7,845	-8,014	-3,318	-3215,07
	Sim	Taxonômica	MP	10,257	20,862	31,337	40,485	50,499	-3,321	-5,749	-6,406	-1,911	-3,466	-2382,48
	Sim	Taxonômica	U	9,456	19,602	29,777	40,365	49,874	-4,957	-8,265	-6,802	-6,177	-2,357	-2923,45

Valores próximos do ótimo de referência também foram obtidos para esta estratégia, tendo esta conseguido atingir resultados ainda melhores que a estratégia anterior, a primeira. Todos os modelos conseguiram gerar boas soluções, destacando-se o modelo sinusoidal. Em termos de operadores de crossover, todos conseguiram um bom desempenho, sendo o melhor deles o operador MP, que apresenta várias soluções positivas. A mutação taxonômica voltou a obter os piores resultados, confirmando, mais uma vez, a sua má sinergia com a codificação binária de soluções.

Este exemplo, conseguiu demonstrar não só o poder do otimizador local, mas também a sua incrível sinergia com os operadores de crossover. Assim, poderá ser dito que a adição destes dois operadores, provou-se decisiva e eficaz, tendo assistido na resolução de alguns dos problemas e complementando as qualidades do BBO.

Capítulo 6

Conclusões e Desenvolvimentos Futuros

Foi implementado um algoritmo evolucionário baseado em biogeografia (BBO), para ser aplicado em vários problemas da área da otimização de sistemas mecânicos, e estudar o seu comportamento. Este algoritmo, foi mais tarde hibridizado com outros conceitos e métodos, tal como, a codificação binária de soluções e o operador de crossover dos algoritmos genéticos, e a aprendizagem e pesquisa local dos algoritmos meméticos. Foram implementados três operadores de crossover, Single-Point (SP), Multiple-Point (MP) e Uniform (U), sendo sujeitos a duas estratégias de seleção dos progenitores para recombinação genética e seleção da descendência sobrevivente, e um otimizador local Hooke-Jeeves. Estes operadores e conceitos foram acrescentados ao algoritmo que já apresentava quatro modelos de migração, Linear, Trapezoidal, Quadrático e Sinusoidal, e dois operadores de mutação, normal e taxonómica.

O BBO base provou ser um modelo com uma enorme facilidade em termos de pesquisa global, nas gerações iniciais, mas uma enorme carência, na pesquisa local, apresentando imensas dificuldades em encontrar ou até mesmo aproximar-se do ponto ótimo da função a ser estudada. A adição da estratégia de *blending* (mistura de valores), aplicado durante o processo de migração, já proposto pelo autor original do BBO, Dan Simon, provou ser extremamente benéfico, aumentando a velocidade do algoritmo, a sua precisão e a sua sinergia com os restantes operadores. Mesmo assim, alguns dos problemas iniciais do BBO, mantinham-se. Foram adicionados outros processos, já conhecidos no ramo da otimização, como a injeção de novas soluções, antes da implementação da hibridização, e os resultados provaram-se positivos, sendo sentido uma melhoria nos resultados.

Em termos de migração, os modelos que mais se destacaram, durante os testes, foram os modelos sinusoidal e linear, que conseguiram apresentar vários resultados ótimos. O modelo quadrático, embora menos consistente, também conseguiu apresentar bons resultados, tendo superado os outros modelos em alguns casos. O pior, o modelo trapezoidal, revelou bons resultados, mas apresentou ser o mais lento, quando comparado aos outros

modelos, e apresentou uma má sinergia com os outros operadores implementados. Em termos de mutação, ambas as mutações conseguiram apresentar bons resultados, quando era utilizado o *blending*. A mutação normal priorizava a velocidade do algoritmo, gerando boas soluções rapidamente, enquanto a mutação taxonômica, apresentou uma precisão maior, ao custo de um pouco de velocidade. A maior disparidade verificou-se nos casos onde se utilizou codificação binária de soluções com um número de dígitos elevado, onde a mutação taxonômica teve grandes dificuldades em chegar a boas soluções, repetidamente, mesmo com a adoção do otimizador local.

A hibridização, começou com a implementação dos operadores de crossover, o que necessitou da implementação de codificação binária, que por sua vez, obrigou à implementação de um sistema de decodificação e codificação binária (*decoding* e *encoding*). Os três operadores implementados, foram sujeitos a duas estratégias de seleção diferente. A inclusão destes operadores, provou ser positiva, havendo uma melhoria de resultados, de forma geral. A segunda estratégia, aparentou ser um pouco melhor, principalmente quando usada em conjunto com o operador MP, mas a primeira estratégia também foi bastante positiva. Os três operadores conseguiram bons resultados, bastantes similares, e apresentaram uma boa sinergia com os restantes operadores e conceitos, especialmente com o *blending*.

De seguida, foi implementado o otimizador local Hooke-Jeeves, que apresentou resultados excelentes, havendo uma enorme melhoria de resultados, mesmo sem a inclusão do crossover. No entanto, os melhores resultados surgiram quando o algoritmo, totalmente hibridizado, foi utilizado. Assim, o algoritmo conseguiu, não só chegar aos pontos ótimos rapidamente, como também, em alguns casos, ultrapassá-los.

Concluindo, o comportamento inicial do método, sugeria que era necessário algumas melhorias, para este chegar ao seu verdadeiro potencial. Com estas novas inovações, o método apresentou excelentes resultados, conseguindo atingir o seu objetivo, com relativa facilidade. Poderá ser dito, que estas inovações, complementam o método de uma forma extremamente positiva e que a sua inclusão deveria ser ainda mais reforçada.

Ainda assim, ainda existem algumas lacunas, que deverão ser trabalhadas, para, realmente, atingir o potencial máximo do algoritmo. Como foi mencionado anteriormente, a mutação taxonômica teve uma má interação com a codificação binária de soluções com um número de dígitos elevado. Uma solução inicial, para este problema, seria utilizar a mutação normal, nas primeiras gerações, para aumentar a velocidade do algoritmo e melhorar a população geral, e, mais tarde, utilizar a mutação taxonômica. Outra solução, passará por analisar o método de codificação binária e alterá-lo de forma que não seja necessário um número de dígitos tão elevado. Finalmente, outra solução relativamente a este problema, seria a realização de uma análise ao problema antes de utilizar o algoritmo, aplicando-se uma triagem de soluções, de forma a descobrir as zonas de maior penalização e evitar que o método evolua para essas regiões mais problemáticas.

Poderia ser interessante, continuar pelo caminho da hibridização e implementar outros

processos de aprendizagem e otimização local, talvez até alterar otimizador local Hooke-Jeeves, para que este melhore a sua sinergia com os outros operadores utilizados por este método. Também seria interessante, estudar o comportamento deste algoritmo para casos mais complexos, tal como, exemplos estruturais, com vários elementos e nodos, ou até em exemplos multiobjectivo.

Além disso, foram apresentados aqui vários operadores dos algoritmos genéticos e modificações do BBO, que não foram implementadas no decorrer deste trabalho. Seria de se esperar que tais inovações, continuem a melhorar o desempenho deste método, podendo até corrigir alguns dos erros e problemas que foram apresentados e discutidos aqui.

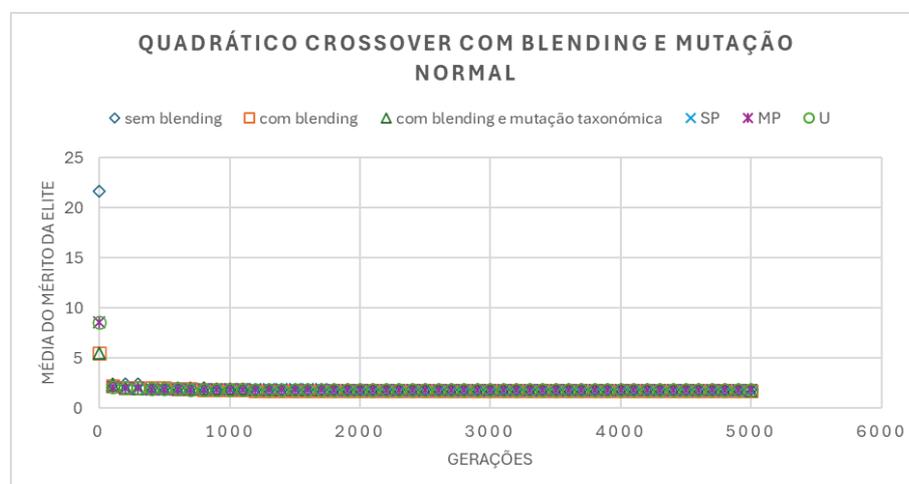
Apêndice A

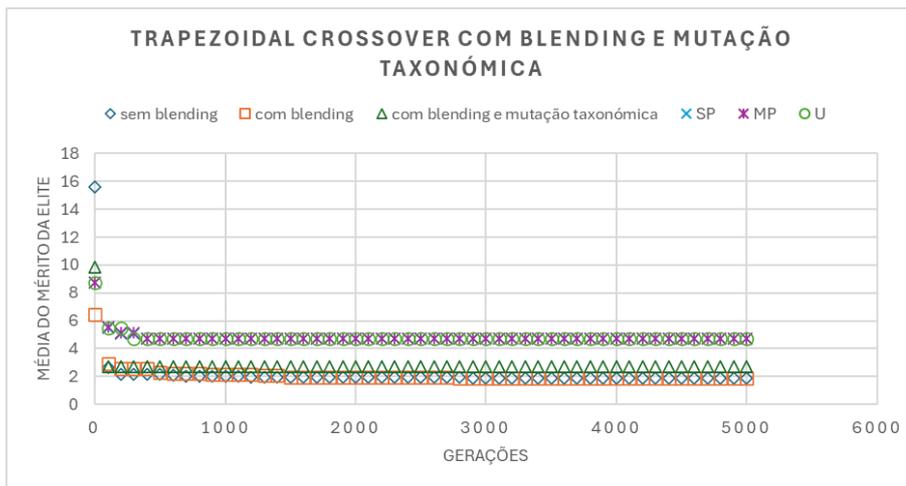
Gráficos da média do mérito da elite

Aqui serão apresentados alguns dos gráficos, mais significativos, que representam a evolução do mérito da elite, ao longo das gerações, para os testes onde foram utilizados os operadores de crossover. As informações relativas às estratégias de seleção e sobre o operador de crossover, estão apresentados nas secções [4.2.1](#) e [3.1.3](#), respetivamente.

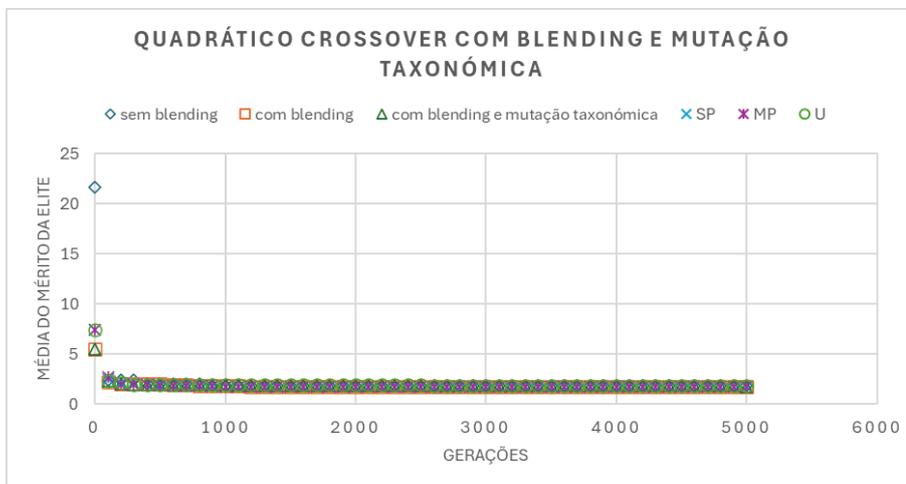
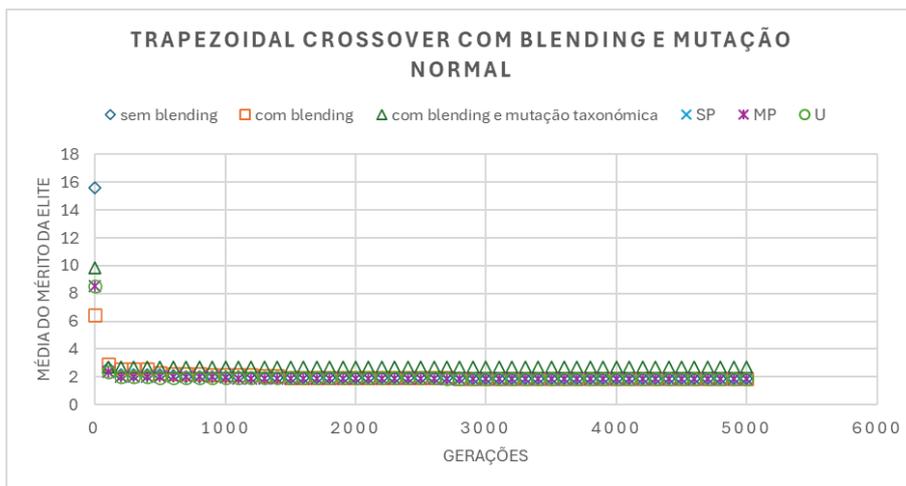
Dimensionamento de uma Viga Soldada

Primeira Estratégia

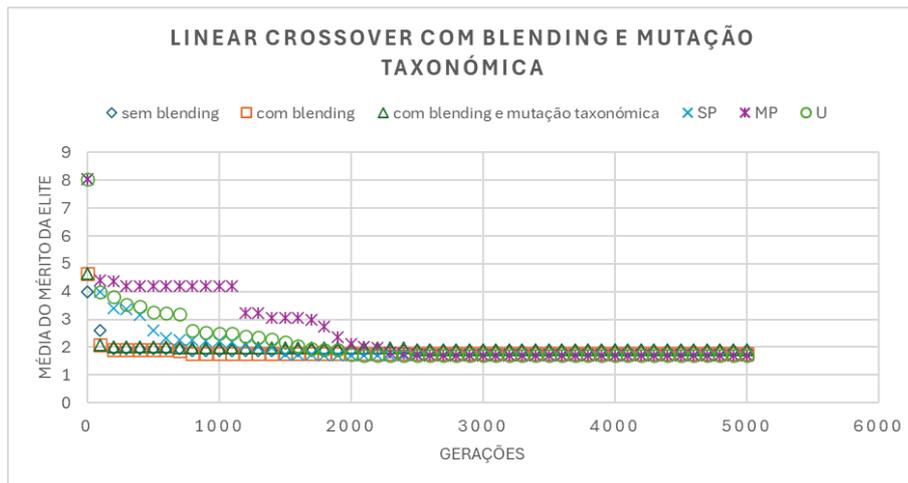
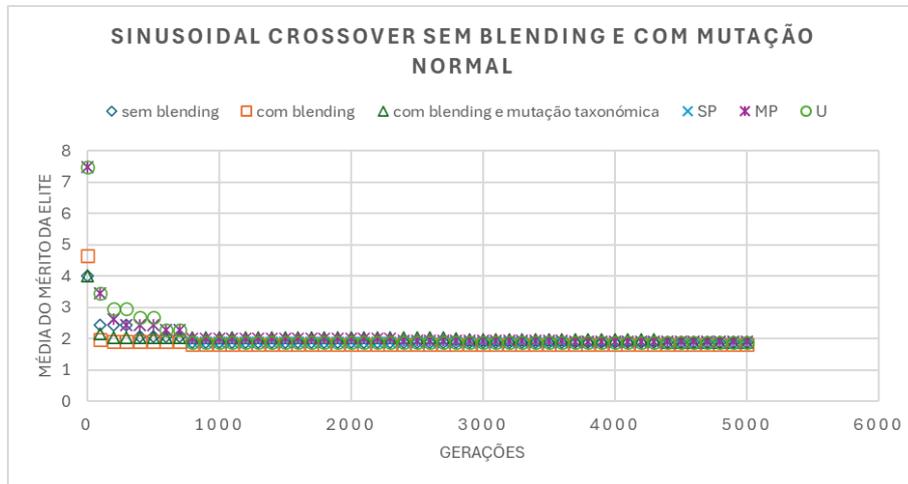




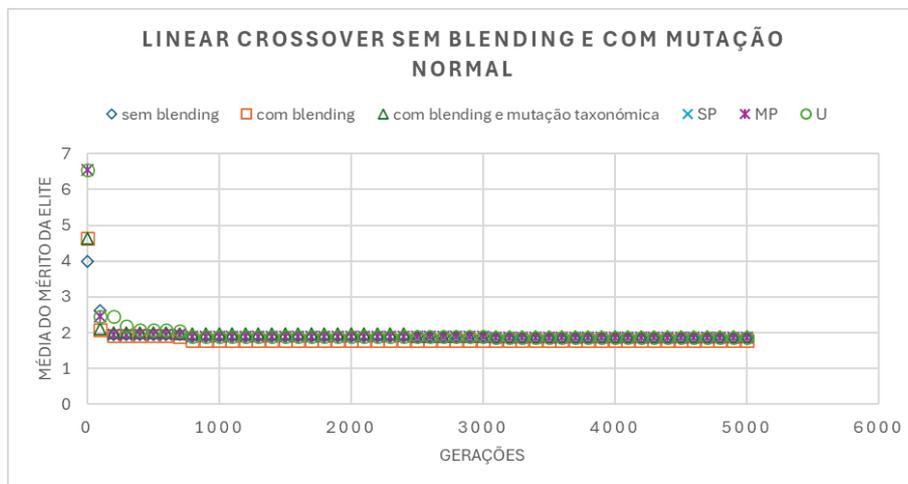
Segunda Estratégia

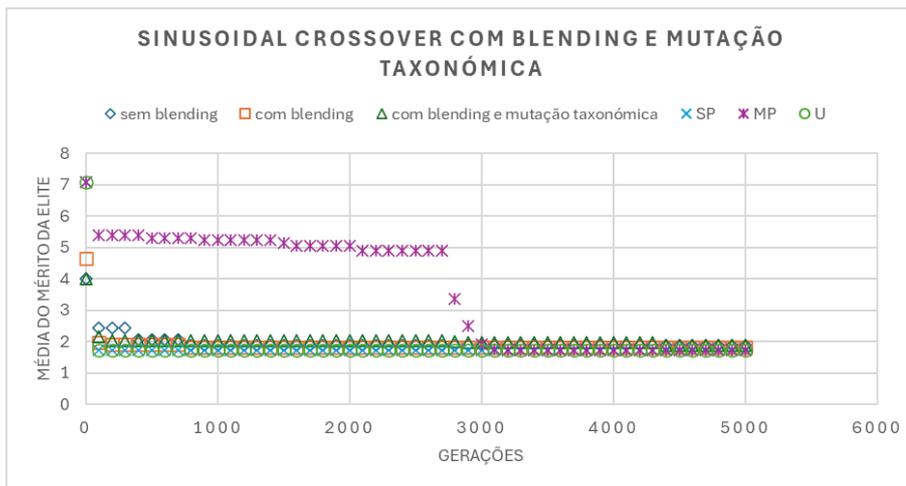


Primeira Estratégia com otimização local



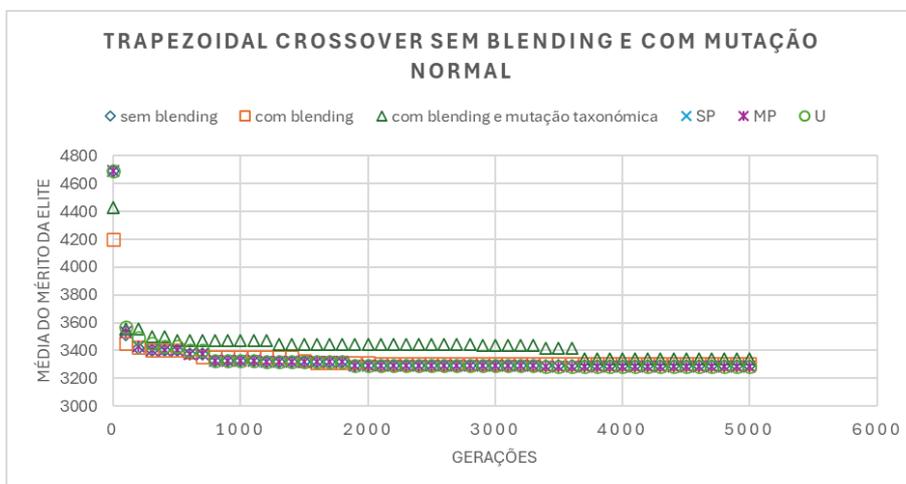
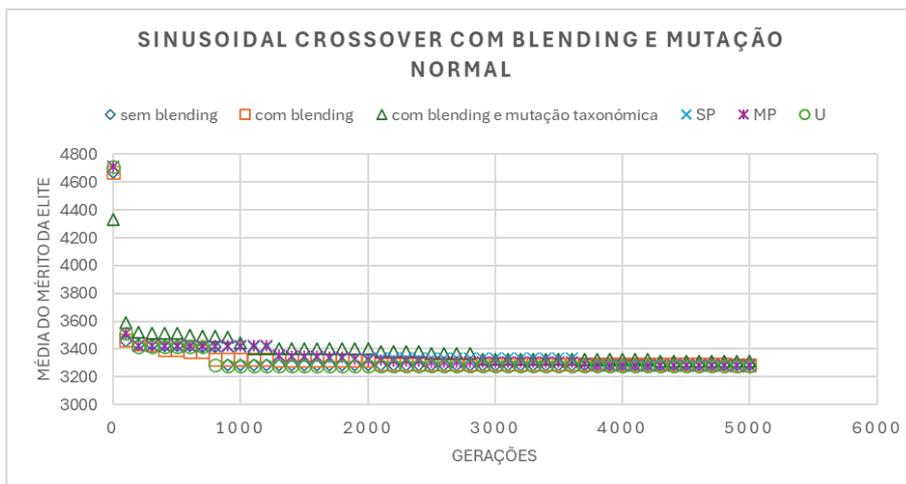
Segunda Estratégia com otimização local



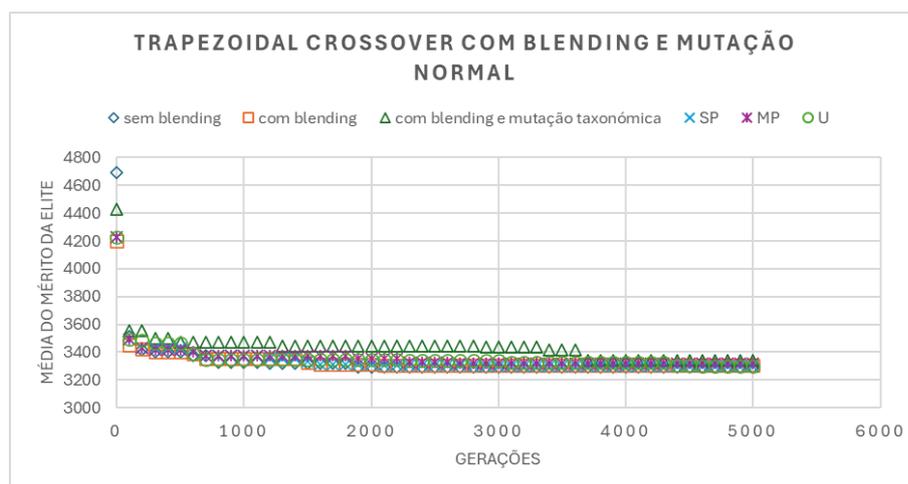
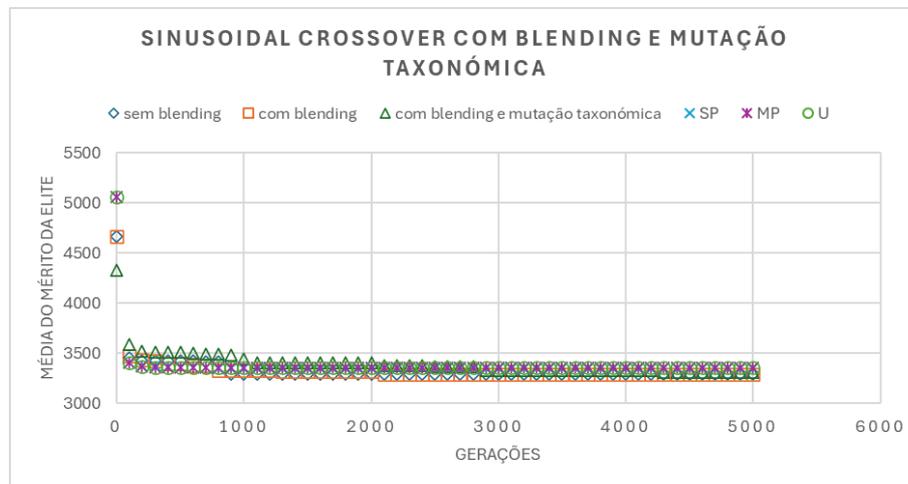


Dimensionamento de um Reforço Soldado Gusset

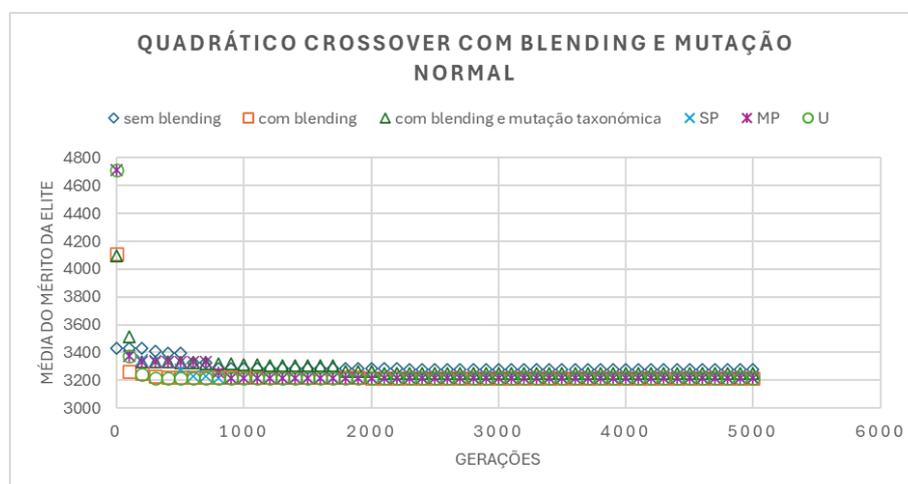
Primeira Estratégia

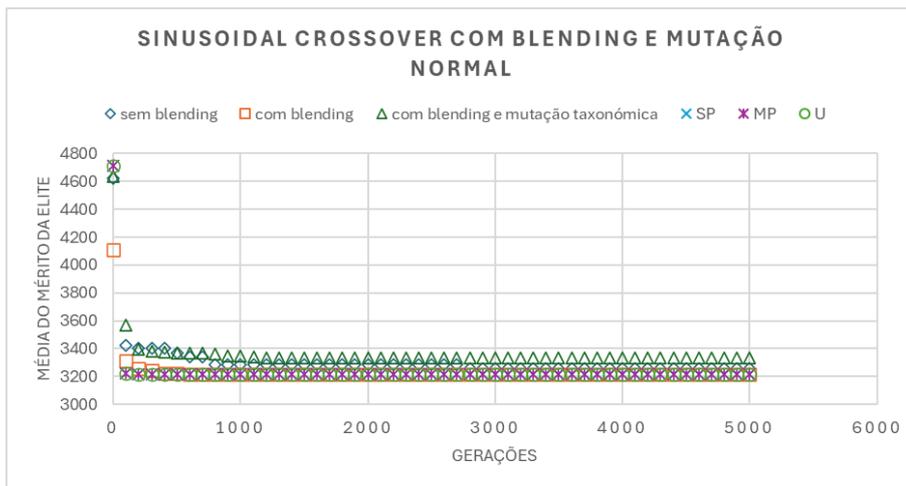


Segunda Estratégia

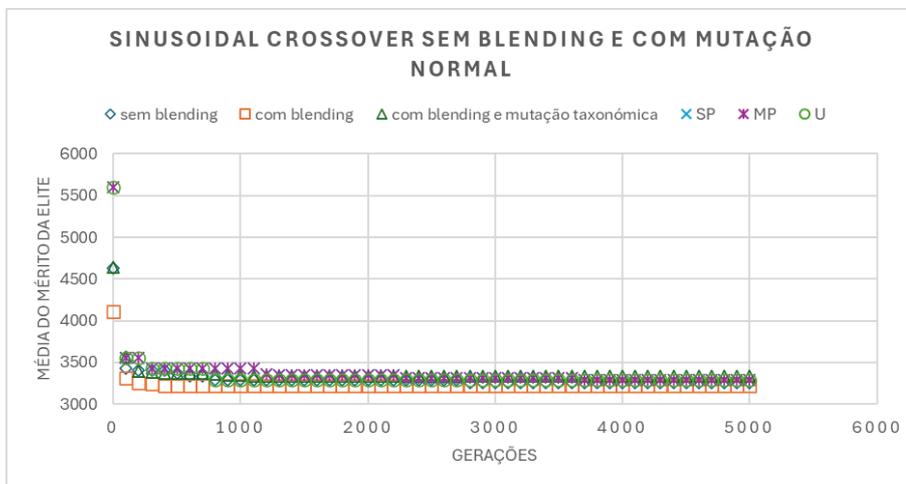
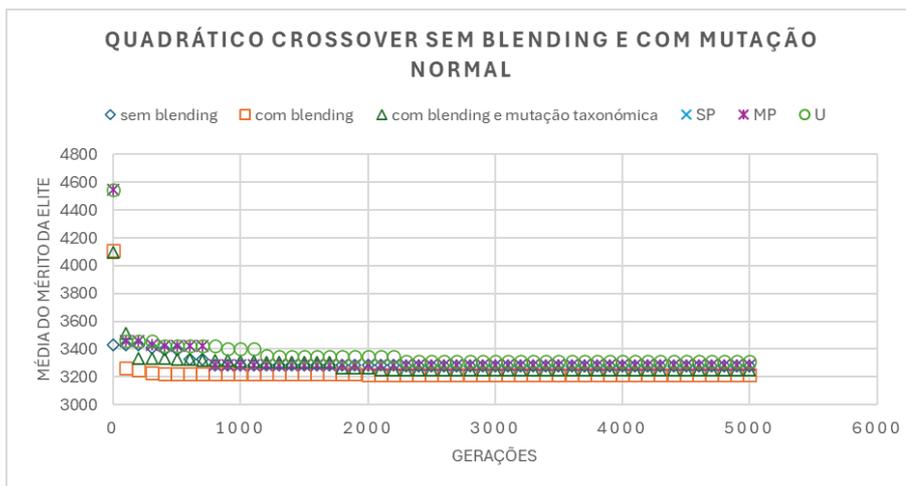


Primeira Estratégia com otimização local



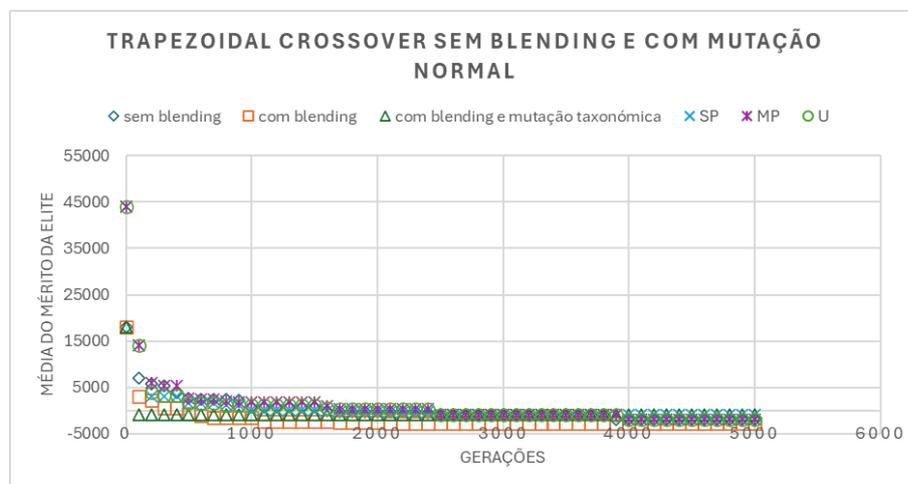
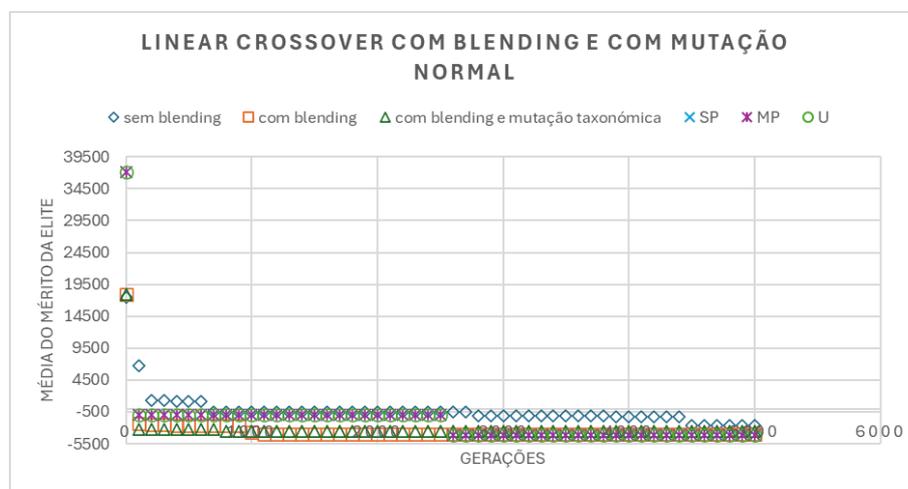


Segunda Estratégia com otimização local

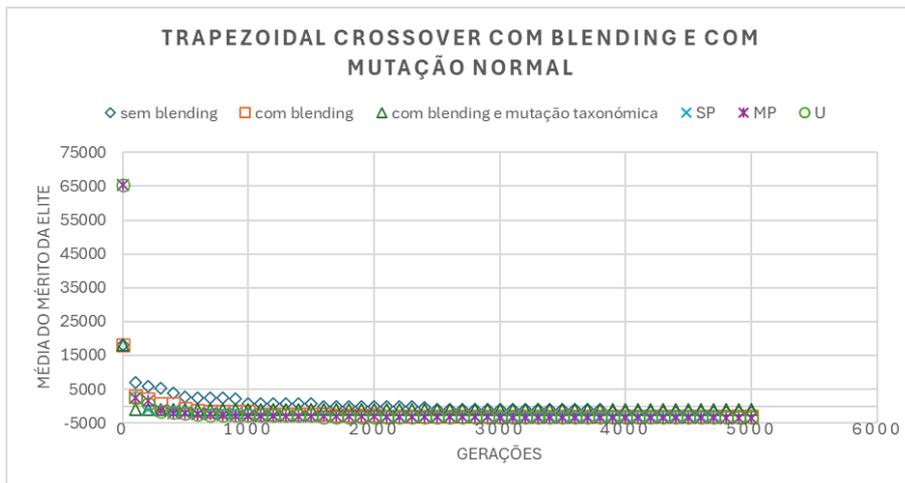
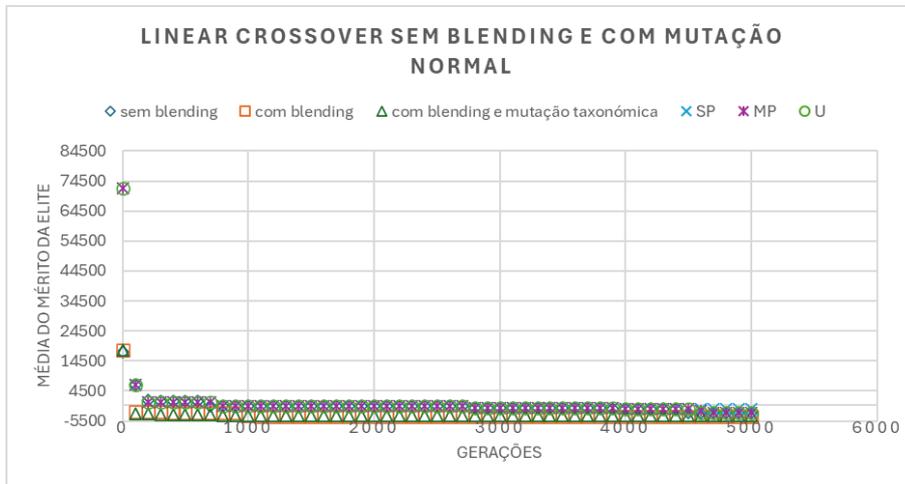


Otimização de um sistema de molas e pesos

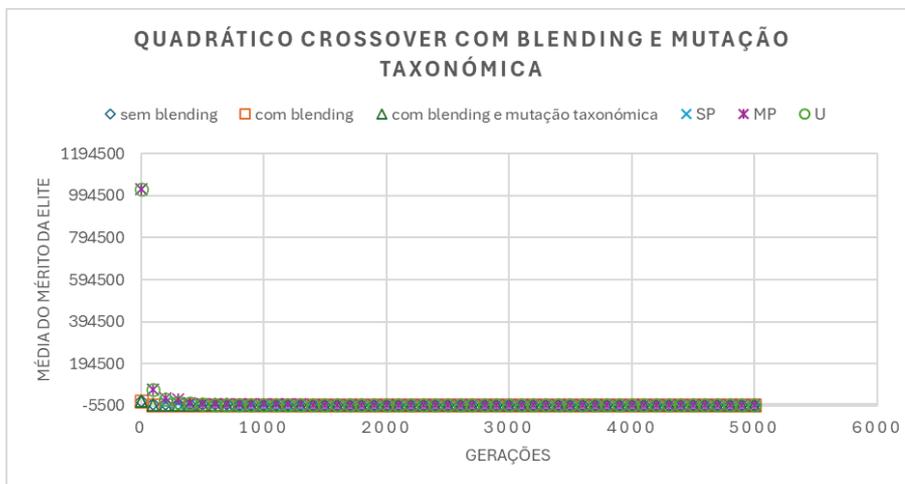
Primeira Estratégia

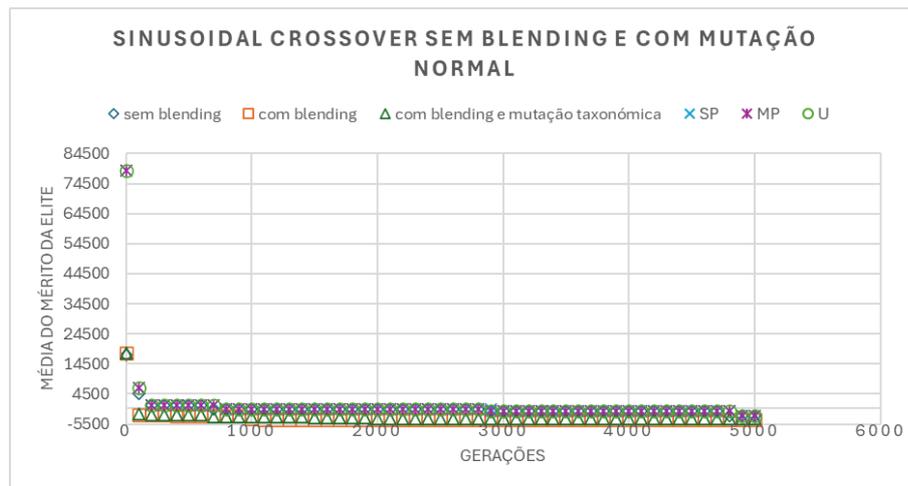


Segunda Estratégia

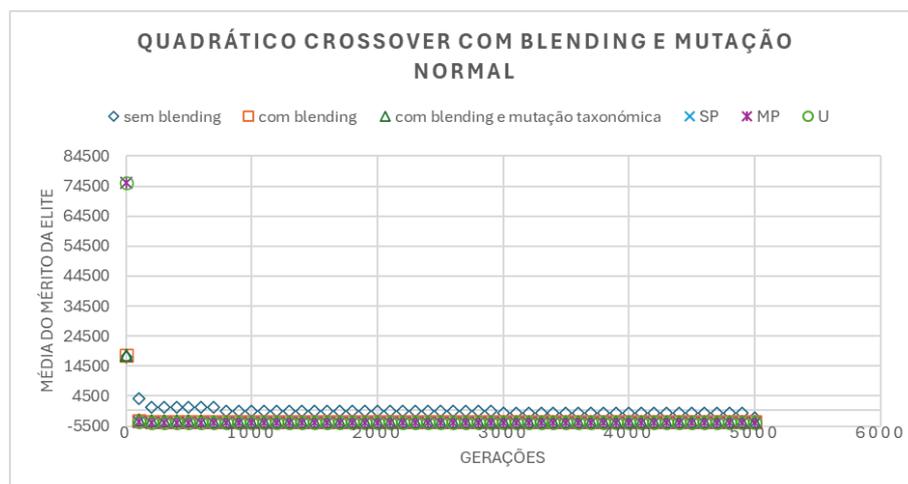
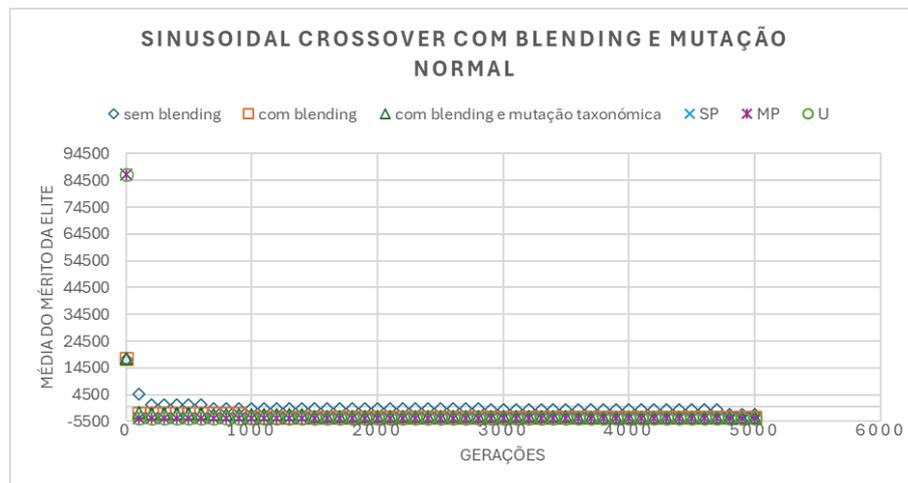


Primeira Estratégia com otimização local





Segunda Estratégia com otimização local



Bibliografia

- [1] John H. Holland. “Genetic Algorithms”. Em: *Scientific American* (1992).
- [2] Richard Dawkins. “The Selfish Gene”. Em: *Oxford University Press* (1976).
- [3] J. B. Lamarck. “Zoological Philosophy”. Em: *s.n.* (1914).
- [4] J. Mark Bladwin. “A New Factor in Evolution”. Em: *The American Nationalist* (1869). DOI: [10.1086/276408](https://doi.org/10.1086/276408).
- [5] D. Simon. “Biogeography-based optimization”. Em: *IEEE Transactions on Evolutionary Computation* (2008). DOI: [10.1109/TEVC.2008.919004](https://doi.org/10.1109/TEVC.2008.919004).
- [6] C. C. António. “Optimization of Mechanical Systems”. Em: *Optimization of Mechanical Systems* (2021).
- [7] *Wikipédia - Otimização*. Acessado: 05/06/2024. URL: <https://pt.wikipedia.org/wiki/Otimiza%C3%A7%C3%A3o>.
- [8] Edward Osborne Wilson e Robert MacArthur. *The Theory of Island Biogeography*. Princeton University Press, 1967.
- [9] V. Garg e K. Deep. “A State-of-the-Art Review of Biogeography-Based Optimization”. Em: *Proceedings of Fourth International Conference on Soft Computing for Problem Solving, Advances in Intelligent Systems and Computing 336* (2015). DOI: [10.1007/978-81-322-2220-0_44](https://doi.org/10.1007/978-81-322-2220-0_44).
- [10] Haiping Ma. “An analysis of the equilibrium of migration models for biogeography-based optimization”. Em: *Elsevier* (2010). DOI: [10.1016/j.ins.2010.05.035](https://doi.org/10.1016/j.ins.2010.05.035).
- [11] F. Albrase, A. Alroomi e J. Talaq. “Optimal coordination of directional overcurrent relays using biogeography-based optimization algorithm”. Em: *IEEE Trans. Power Del.* (2015).
- [12] A. Al-Roomi e M. El-Hawary. “Metropolis biogeography-based optimization”. Em: *Inf. Sci.* (2016).
- [13] Q. Fen et al. “Modified biogeography-based optimization with local search mechanism”. Em: *J. Appl. Math.* (2013).
- [14] W. Gong, Z. Cai e C. King. “DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization”. Em: *Soft Comput.* (2010).

- [15] A. Bhattacharya e P. Chattopadhyay. “Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch”. Em: *IEEE Trans. Power Syst.* (2010).
- [16] P. Chattopadhyay. “Hybrid differential evolution with biogeography-based optimization algorithm for solution of economic emission load dispatch problems”. Em: *Expert Syst. Appl.* (2011).
- [17] I. Boussard, A. Chatterjee e P. Siarry. “Hybridizing biogeography-based optimization with differential evolution for optimal power allocation in wireless sensor networks”. Em: *IEEE Trans. Veh. Technol.* (2011).
- [18] H. Kundra, A. Kaur e V. Panchal. “An integrated approach to biogeography-based optimization with case based reasoning for exploring groundwater responsibility”. Em: *Delving: J. Technol. Eng. Sci* (2009).
- [19] H. Kundra e M. Sood. “Cross-country path finding using hybrid approach of PSO and BBO”. Em: *Int. J. Comput. Appl.* (2010).
- [20] L. Wang e Y. Xu. “An effective hybrid biogeography-based optimization algorithm for parameter estimation of chaotic systems.” Em: *Expert Syst. Appl.* (2012).
- [21] P. Arora, H. Kundra e V. Panchal. “Fusion of biogeography-based optimization and artificial bee colony for identification of natural terrain features”. Em: *Int. J. Adv. Comput. Sci. Appl.* (2012).
- [22] G. Wang et al. “Hybridizing harmony search with biogeography-based optimization for global numerical optimization.” Em: *J. Comput. Theor. Nanosci.* (2013).
- [23] K. Jayaraman e G. Ravi. “Long-term sector-wise electrical energy forecasting using artificial neural network and biogeography-based optimization”. Em: *Electr. Power Compon. Syst* (2015).
- [24] U. Krishnasamy e D. Najundappan. “Hybrid weighted probabilistic neural network and biogeography based optimization for dynamic economic dispatch of integrated multiple-fuel and wind power plants”. Em: *Int. J. Electr. Power Energy Syst* (2016).
- [25] J. Jin et al. “Biogeography-based optimization with ensemble of migration models for global numerical optimization”. Em: *Proc. 2012 IEEE Congr. Evol. Comput.* (2012).
- [26] Z. Xie et al. “Variations of biogeography-based optimization and Markov analysis”. Em: *Inf. Sci.* (2013).
- [27] X. Chen et al. “Biogeography-based optimization with covariance matrix based migration”. Em: *Appl. Soft Comput.* (2016).
- [28] Q. Feng et al. “Biogeography-based optimization with orthogonal crossover”. Em: *Math. Probl. Eng.* (2013).

- [29] G. Vanita e K. Deep. “Performance of Laplacian biogeography-based optimization algorithm on CEC 2014 continuous optimization benchmarks and camera calibration problem”. Em: *Swarm Evol. Comput.* (2016).
- [30] G. Xiong, D. Shi e X. Duan. “Enhancing the performance of biogeography-based optimization using polyphyletic migration operator and orthogonal learning”. Em: *Comput. Oper. Res.* (2014).
- [31] W. Gong et al. “A real-coded biogeography-based optimization with mutation”. Em: *Appl. Math. Comput.* (2010).
- [32] J. Bansal. “Burgeoning nature of biogeography based optimization”. Em: *Proc. 2nd Int. Conf. Harmonic Search Algorithm* (2015).
- [33] Q. Niu, L. Zhang e K. Li. “A biogeography-based optimization algorithm with mutation strategies for model parameter estimation of solar and fuel cells”. Em: *Energy Convers. Manage.* (2014).
- [34] M. Lohokare, S. Pattnaik e B. Panigrahi. ““Neighborhood search driven accelerated biogeography-based optimization for optimal load dispatch”. Em: *IEEE Trans. Syst. Man Cybern. - Part C* (2012).
- [35] M. Lohokare, B. Panigrahi e S. Pattnaik S.and Das. “Accelerated biogeography-based optimization with neighborhood search for optimization”. Em: *Appl. Soft Comput.* (2013).
- [36] M. Ergezer, D. Simon e D. Du. “Oppositional biogeography-based optimization”. Em: *Proc. IEEE Int. Conf. Syst.* (2009).
- [37] P. Roy e D. Mandal. “Quasi-oppositional biogeography-based optimization for multi-objective optimal power flow”. Em: *Electr. Power Compon. Syst.* (2012).
- [38] W. Guo et al. ““Backtracking biogeography-based optimization for numerical optimization and mechanical design problems”. Em: *Appl. Intell.* (2016).
- [39] S. Kim, J. Byeon e S. Lee. “A grouping biogeography-based optimization for location area planning”. Em: *Neural Comput. Appl.* (2015).
- [40] Y. Zheng et al. “Localized biogeography-based optimization”. Em: *Soft Comput.* (2014).
- [41] Charles Darwin e Alfred Russell. “On the Tendency of Species to form Varieties”. Em: *Journal of the Proceedings of the Linnean Society* (1858).
- [42] Charles Darwin. “On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life”. Em: *The Theory of Development in Nature* (1859).
- [43] J. A. Joines e C. R. Houck. “On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with ga’s.” Em: *IEEE world congress on computational intelligence* (1994).

- [44] A. B. Hadj-Alouane e J. C. Bean. “A genetic algorithm for the multiple-choice integer program”. Em: *Department of Industrial and Operations Engineering, The University of Michigan* (1992).
- [45] J. Holland. “Adaptation in Natural and Artificial Systems”. Em: *The MIT Press* (1975).
- [46] F. Herrera, M. Lozano e A. M. Sánchez. “A taxonomy for the crossover operator for real-coded genetic algorithms”. Em: *Int J Intell Syst* (2003).
- [47] L.J. Eshelman, A. Caruana e J.D. Schaffer. “Biases in the crossover landscape”. Em: *Proceedings of the third international conference on genetic algorithms* (1989).
- [48] R. Moraglio A. anda Poli e R. Seehuus. “Geometric Crossover for Biological Sequences”. Em: *Genetic Programming, 9th European Conference* (2006). DOI: [10.1007/11729976_11](https://doi.org/10.1007/11729976_11).
- [49] Kalyan Deb, Ashish Anand e Dhiraj Joshi. “A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization”. Em: *Evolutionary Computation* (2002). DOI: [10.1162/106365602760972767](https://doi.org/10.1162/106365602760972767).
- [50] Kalyanmoy Deb e Ram Bhushan Agrawal. “Simulated Binary Crossover for Continuous Search Space”. Em: *Complex Systems* (1995).
- [51] C. García-Martínez et al. “Global and local real-coded genetic algorithms based on parent-centric crossover operators”. Em: *European Journal of Operational Research* (2008). DOI: [10.1016/j.ejor.2006.06.043](https://doi.org/10.1016/j.ejor.2006.06.043).
- [52] F. Herrera e M. Lozano. “Gradual distributed real-coded genetic algorithms”. Em: *IEEE Transactions on Evolutionary Computation* (2000).
- [53] Crina Grosan e Ajith Abraham. *Studies in Computational Intelligence, volume 75*. Springer, 2007. Cap. Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews.
- [54] Carlos García-Martínez e Manuel Lozano. “Local Search Based on Genetic Algorithms”. Em: *Advances in Metaheuristics for Hard Optimization (pp.199-221)* (2007). DOI: [10.1007/978-3-540-72960-0_10](https://doi.org/10.1007/978-3-540-72960-0_10).
- [55] Mahamed G. H. Omran. “A novel cultural algorithm for real-parameter optimization”. Em: *International Journal of Computer Mathematics* (2015). DOI: [10.1080/00207160.2015.1067309](https://doi.org/10.1080/00207160.2015.1067309).
- [56] Darrell Whitley, V. Scott Gordon e Keith Mathias. “Lamarckian evolution, the Baldwin effect and function optimization”. Em: *Lecture Notes in Computer Science, volume 866 (pp. 5-15)* (2005).
- [57] Q. H. Nguyen, Y. S. Ong e M. H. Lim. “Non-genetic transmission of memes by diffusion”. Em: *Proceedings of the 8th annual conference on Genetic and evolutionary computation (pp. 1017-1024)* ().

- [58] Fluvio Corno, M. Reorda e Giovanni Sguillero. “The selfish gene algorithm: a new evolutionary optimization strategy”. Em: *ACM Symposium on Applied Computing* (1998).
- [59] Y. Zheng et al. *Biogeography-based optimization: Algorithms and applications*. Science Press Beijing, 2018. DOI: [10.1007/978-981-13-2586-1](https://doi.org/10.1007/978-981-13-2586-1).
- [60] Dan Simon e Haiping Ma. “Blended biogeography-based optimization for constrained optimization”. Em: *Eng Appl Artif Intell* (2011). DOI: [10.1016/j.engappai.2010.08.005](https://doi.org/10.1016/j.engappai.2010.08.005).
- [61] J. Hao. “A Novel Biogeography-Based Optimization Algorithm with Momentum Migration and Taxonomic Mutation”. Em: *Lecture Notes in Computer Science* (2011). DOI: [10.1007/978-3-030-53956-6_8](https://doi.org/10.1007/978-3-030-53956-6_8).
- [62] I. Boussaïd et al. “Biogeography-based optimization for constrained optimization problems”. Em: *Comput Oper Res* (2012). DOI: [10.1016/j.cor.2012.04.012](https://doi.org/10.1016/j.cor.2012.04.012).
- [63] X. Bi e J. Wang. “Constrained optimization based on epsilon constrained biogeography-based optimization”. Em: *Proceedings of the 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics* (2012). DOI: [10.1109/IHMSC.2012.184](https://doi.org/10.1109/IHMSC.2012.184).
- [64] H. P. Ma, X. Y. Ruan e Z. X. Pan. “Handling multiple objectives with biogeography-based optimization”. Em: *International Journal of Automation and Computing* (2012). DOI: [10.1007/s11633-012-0613-9](https://doi.org/10.1007/s11633-012-0613-9).
- [65] Xiao-jun Bi, J. Wang e B. Li. “Multi-objective optimization based on hybrid biogeography-based optimization”. Em: *Xi Tong Gong Cheng Yu Dian Zi Ji Shu/Systems Engineering and Electronics* (2014). DOI: [10.3969/j.issn.1001-506X.2014.01.29..](https://doi.org/10.3969/j.issn.1001-506X.2014.01.29..)
- [66] C. C. António. “A multilevel genetic algorithm for optimization of geometrically nonlinear stiffened composite structures”. Em: *Struct Multidisc Optim* (2002). DOI: [10.1007/s00158-002-0249-4](https://doi.org/10.1007/s00158-002-0249-4).
- [67] C. C. António e G. N. Carneiro. “Optimization of Mechanical Systems: Optimal Design Applications”. Em: *Optimization of Mechanical Systems: Optimal Design Applications* (2022).
- [68] *Welded Beam Design Optimization - Maple Help*. Acessado: Abril 2024. URL: https://www.maplesoft.com/products/maple/app_gallery/pdf/Welded_Beam_Design_Optimization.pdf.
- [69] Regulamento de Estruturas de Aço para Edifícios (REAE). “Decreto-Lei nº 21/86”. Em: (31 de julho de 1986).
- [70] H. F. S. Gonçalves. *Development and implementation of memetic models for the optimal design of mechanical systems*. 2023.
- [71] G. N. Vanderplaats. “Numerical optimization techniques for engineering design”. Em: *Numerical optimization techniques for engineering design* (1999).

