

Igor Lucas de Souza Russo

**Otimização Multiobjetivo e Programação Genética para Descoberta de
Conhecimento em Engenharia**

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional, da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre em Modelagem Computacional.

Orientador: Prof. D.Sc. Heder Soares Bernardino

Coorientador: Prof. D.Sc. Helio José Corrêa Barbosa

Juiz de Fora

2017

Ficha catalográfica elaborada através do programa de geração automática da Biblioteca Universitária da UFJF, com os dados fornecidos pelo(a) autor(a)

Souza Russo, Igor Lucas de.

Otimização Multiobjetivo e Programação Genética para Descoberta de Conhecimento em Engenharia / Igor Lucas de Souza Russo. -- 2017.

102 f. : il.

Orientador: Heder Soares Bernardino

Coorientador: Helio José Corrêa Barbosa

Dissertação (mestrado acadêmico) - Universidade Federal de Juiz de Fora, ICE/Engenharia. Programa de Pós-Graduação em Modelagem Computacional, 2017.

1. Innovization. 2. Otimização Multiobjetivo. 3. Descoberta de conhecimento. 4. Programação Genética. I. Soares Bernardino, Heder, orient. II. Corrêa Barbosa, Helio José, coorient. III. Título.

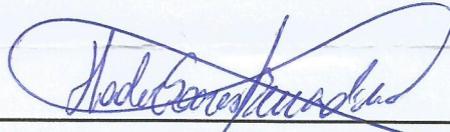
Igor Lucas de Souza Russo

Otimização Multiobjetivo e Programação Genética para Descoberta de
Conhecimento em Engenharia

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora, na área de concentração Modelagem Computacional, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional.

Aprovada em: 26 de Janeiro de 2017

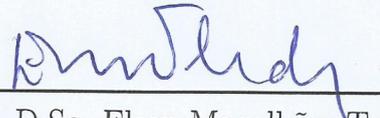
BANCA EXAMINADORA



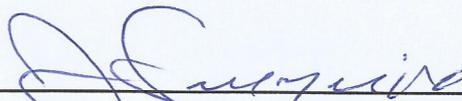
Prof. D.Sc. Heder Soares Bernardino - Orientador
Universidade Federal de Juiz de Fora



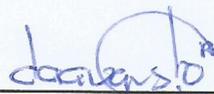
Prof. D.Sc. Helio José Corrêa Barbosa - Coorientador
Universidade Federal de Juiz de Fora



Prof. D.Sc. Elson Magalhães Toledo
Universidade Federal de Juiz de Fora



Prof. D.Sc. João Nisan Correia Guerreiro
Laboratório Nacional de Computação Científica



Prof. D.Sc. Douglas Adriano Augusto
Fiocruz

*Dedico este trabalho aos meus
pais, Aristeia e Elcyr.*

AGRADECIMENTOS

A Deus por ter me dado forças para enfrentar os novos desafios.

À minha família, pelo encorajamento e apoio constantes.

À namorada Graziela, pelo carinho, compreensão e também por me incentivar a sempre perseguir meus sonhos.

Aos meus orientadores, Professores Heder Bernardino e Helio Barbosa, pela oportunidade, pelos ensinamentos valiosos e sobretudo pela paciência.

Aos caros amigos Thiago, Bruno, Marcelo, Victor Hugo, Daniel e Geissy por tornarem a caminhada mais suave, mesmo nos trechos mais íngremes.

Agradeço ainda à tia Helena, à Diná e à Renata pelo suporte e ensinamentos.

Aos coordenadores do Programa de Pós Graduação em Modelagem Computacional (PPGMC), Professores Rafael Bonfim e Leonardo Goliatt, pela diligência em auxiliar sempre que necessário. Aos demais Professores do PPGMC pelos seus ensinamentos e a todos aqueles que, durante esses dois anos, contribuíram de algum modo para a elaboração deste trabalho.

Aos Professores membros da banca avaliadora, que contribuíram imensamente para a qualidade e consistência deste trabalho.

Finalmente, gostaria de agradecer à CAPES pela bolsa de estudos e à UFJF pelo auxílio financeiro obtido para participação em conferência internacional.

*“A tarefa não é tanto ver aquilo
que ninguém viu, mas pensar o
que ninguém ainda pensou sobre
aquilo que todo mundo vê.”*

Arthur Schopenhauer.

RESUMO

A área de Otimização envolve o estudo e emprego de métodos para determinação dos parâmetros que levam à obtenção de soluções ótimas, de acordo com critérios denominados objetivos. Um problema é classificado como multiobjetivo quando apresenta objetivos múltiplos e conflitantes, que devem ser otimizados simultaneamente. Recentemente tem crescido o interesse dos pesquisadores pela análise de pós-otimalidade, que consiste na busca por propriedades intrínsecas às soluções ótimas de problemas de otimização e que podem lançar uma nova luz à compreensão dos mesmos. *Innovization* (inovação através de otimização, do inglês *innovation through optimization*) é um processo de descoberta de conhecimento a partir de problemas de otimização na forma de relações matemáticas entre variáveis, objetivos, restrições e parâmetros. Dentre as técnicas de busca que podem ser utilizadas neste processo está a Programação Genética (PG), uma meta-heurística bioinspirada capaz de evoluir programas de forma automatizada. Além de numericamente válidos, os modelos encontrados devem utilizar corretamente as variáveis de decisão em relação às unidades envolvidas, de forma a apresentar significado físico coerente. Neste trabalho é proposta uma alternativa para tratamento das unidades através de operações protegidas que ignoram os termos inválidos. Além disso, propõe-se aqui uma estratégia para evitar a obtenção de soluções triviais que não agregam conhecimento sobre o problema. Visando aumentar a diversidade dos modelos obtidos, propõe-se também a utilização de um arquivo externo para armazenar as soluções de interesse ao longo da busca. Experimentos computacionais são apresentados utilizando cinco estudos de caso em engenharia para verificar a influência das ideias propostas. Os problemas tratados aqui envolvem os projetos de: uma treliça de 2 barras, uma viga soldada, do corte de uma peça metálica, de engrenagens compostas e de uma treliça de 10 barras, sendo este último ainda não explorado na literatura de descoberta de conhecimento. Finalmente, o conhecimento inferido no estudo de caso da estrutura de 10 barras é utilizado para reduzir a dimensionalidade do problema.

Palavras-chave: Innovization. Otimização Multiobjetivo. Descoberta de conhecimento. Programação Genética.

ABSTRACT

The area of optimization involves the study and the use of methods to determine the parameters that lead to optimal solutions, according to criteria called objectives. A problem is classified as multiobjective when it presents multiple and conflicting objectives which must be simultaneously optimized. Recently, the interest of the researchers has grown in the analysis of post-optimality, which consists in the search for intrinsic properties of the optimal solutions of optimization problems. This can shed a new light on the understanding of the optimization problems. Innovization (from innovation through optimization) is a process of knowledge discovery from optimization problems in the form of mathematical relationships between variables, objectives, constraints, and parameters. Genetic Programming (GP), a search technique that can be used in this process, is a bio-inspired metaheuristic capable of evolving programs automatically. In addition to be numerically valid, the models found must correctly use the decision variables with respect to the units involved, in order to present coherent physical meaning. In this work, a method is proposed to handle the units through protected operations which ignore invalid terms. Also, a strategy is proposed here to avoid trivial solutions that do not add knowledge about the problem. In order to increase the diversity of the models obtained, it is also proposed the use of an external file to store the solutions of interest found during the search. Computational experiments are presented using five case studies in engineering to verify the influence of the proposed ideas. The problems dealt with here are the designs of: a 2-bar truss, a welded beam, the cutting of a metal part, composite gears, and a 10-bar truss. The latter was not previously explored in the knowledge discovery literature. Finally, the inferred knowledge in the case study of the 10-bar truss structure is used to reduce the dimensionality of that problem.

Keywords: Innovization. Multiobjective Optimization. Knowledge discovery. Genetic Programming.

SUMÁRIO

1	INTRODUÇÃO.....	15
1.1	Organização da dissertação	21
2	OTIMIZAÇÃO MULTIOBJETIVO	22
2.1	INTRODUÇÃO	22
2.2	CONCEITOS BÁSICOS	22
2.3	ALGORITMOS EVOLUCIONISTAS PARA PROBLEMAS MULTIOBJETIVO	27
2.3.1	<i>NSGA-II</i>	29
2.4	MEDIDAS DE DESEMPENHO	34
2.4.1	<i>Hipervolume</i>	34
3	<i>INNOVIZATION</i>	35
3.1	INTRODUÇÃO	35
3.1.1	<i>Processo automatizado de Innovization</i>	36
3.1.2	<i>Innovization baseado em Programação Genética</i>	40
3.1.2.1	<i>Introdução</i>	40
3.1.2.2	<i>Programação Genética</i>	42
3.1.2.3	<i>Processo de Innovization baseado em Programação Genética</i>	44
4	MÉTODOS PROPOSTOS	48
4.1	ALTERNATIVA PARA TRATAMENTO DE UNIDADES	48
4.2	ESTRATÉGIA PARA EVITAR A OBTENÇÃO DE SOLUÇÕES TRIVIAIS	50
4.3	UTILIZAÇÃO DE ARQUIVO EXTERNO	52
5	ESTUDOS DE CASOS.....	55
5.1	INTRODUÇÃO	55
5.2	ESTUDO DE CASO 1: PROJETO DE UMA TRELIÇA DE 2 BARRAS	55
5.3	ESTUDO DE CASO 2: PROJETO DE UMA VIGA SOLDADA ...	56

5.4	ESTUDO DE CASO 3: PROCESSO DE CORTE DE UMA BARRA DE METAL	60
5.5	ESTUDO DE CASO 4: PROJETO DE ENGRENAGENS COMPOSTAS	62
5.6	ESTUDO DE CASO 5: PROJETO DE UMA TRELIÇA DE 10 BARRAS	63
6	EXPERIMENTOS COMPUTACIONAIS	65
6.1	INTRODUÇÃO	65
6.2	ESTUDO DE CASO 1: PROJETO DE UMA TRELIÇA DE 2 BARRAS	66
6.2.1	<i>Problema Multiobjetivo</i>	66
6.2.2	<i>Descoberta de Conhecimento via Innovization</i>	68
6.3	ESTUDO DE CASO 2: PROJETO DE UMA VIGA SOLDADA ...	72
6.3.1	<i>Problema Multiobjetivo</i>	72
6.3.2	<i>Descoberta de Conhecimento via Innovization</i>	73
6.4	ESTUDO DE CASO 3: PROCESSO DE CORTE DE UMA BARRA DE METAL	76
6.4.1	<i>Problema Multiobjetivo</i>	76
6.4.2	<i>Descoberta de Conhecimento via Innovization</i>	77
6.5	ESTUDO DE CASO 4: PROJETO DE ENGRENAGENS COMPOSTAS	81
6.5.1	<i>Problema Multiobjetivo</i>	81
6.5.2	<i>Descoberta de Conhecimento via Innovization</i>	83
6.6	ESTUDO DE CASO 5: PROJETO DE UMA TRELIÇA DE 10 BARRAS	85
6.6.1	<i>Problema Multiobjetivo</i>	85
6.6.2	<i>Descoberta de Conhecimento via Innovization</i>	86
7	CONCLUSÕES E TRABALHOS FUTUROS	94
	REFERÊNCIAS	96

LISTA DE ILUSTRAÇÕES

2.1	Mapeamento entre os espaços de decisão e objetivos.	23
2.2	Convergência e diversidade ilustradas num problema com dois objetivos.	25
2.3	Exemplo de Ordenação de Pareto, em que 13 soluções são classificadas em 3 postos.	27
2.4	Exemplo do cômputo do CD para uma solução i de uma Frente de Pareto, dado neste caso por $a + b$. Figura adaptada de Deb et al. (2002).	31
2.5	Representação da geração da próxima população de uma iteração do algoritmo NSGA-II, a partir da população atual (<i>Filhos</i>) e da população anterior (<i>Pais</i>). Figura adaptada de Deb et al. (2002)	33
2.6	Exemplo de cálculo do hipervolume H para um conjunto de 5 soluções não- dominadas em um problema com dois objetivos a serem minimizados. W é o ponto de referência. H corresponde à área da região sombreada. Figura baseada em Deb (2001).	34
3.1	Exemplo de agrupamento de 15 c -valores, com $d = 7$ e critério de formação de sub-grupos igual a 2. Adaptado de Bandaru et al. (2015)	39
3.2	Árvore para um programa que calcula o valor de $X \times X + \frac{Y}{2}$	42
3.3	Exemplo de recombinação na PG com dois pais gerando dois filhos.	44
3.4	Exemplo de mutação de ponto na PG, em que dois nós (um de função e outro terminal) são modificados.	45
3.5	Exemplo de duas árvores que pertencem ao mesmo nicho.	46
4.1	Exemplo de avaliação de uma árvore utilizando as operações protegidas definidas neste trabalho. São utilizadas duas variáveis (X_1 e X_2) e 3 unidades.	50
4.2	Exemplo de árvores que representam o mesmo princípio de projeto, $X_1 \times X_3 +$ $X_2 = c$	53
5.1	Ilustração de uma treliça de 2 barras (Deb e Srinivasan 2006).	55
5.2	Ilustração de uma viga soldada (Deb e Srinivasan 2006).	57
5.3	Esquema de corte de uma peça metálica em um torno mecânico.	60

5.4	Representação do projeto de 4 engrenagens compostas. Adaptado de Deb e Srinivasan (2006).	63
5.5	Ilustração de uma treliça de 10 barras (Vargas 2015).	64
6.1	Frente de Pareto encontrada em uma execução do NSGA-II para o problema da treliça de duas barras.	67
6.2	Frente de Pareto com as soluções não-dominadas obtidas por 30 execuções do NSGA-II para o problema da treliça de duas barras.	67
6.3	Frente de Pareto encontrada em uma execução do NSGA-II para o problema da viga soldada.	72
6.4	Frente de Pareto com as soluções não-dominadas obtidas por 30 execuções do NSGA-II para o problema da viga soldada.	73
6.5	Frente de Pareto encontrada em uma execução do NSGA-II para o problema de otimização do processo de corte de uma peça metálica.	77
6.6	Frente de Pareto com as soluções não-dominadas obtidas por 30 execuções do NSGA-II para o problema de otimização do processo de corte de uma peça metálica.	77
6.7	Frente de Pareto encontrada em uma execução do NSGA-II para o problema das engrenagens compostas.	82
6.8	Frente de Pareto com as soluções não-dominadas obtidas por 30 execuções do NSGA-II para o problema das engrenagens compostas.	83
6.9	Frente de Pareto encontrada em uma execução do NSGA-II para o problema da treliça de 10 barras.	86
6.10	Frente de Pareto com as soluções não-dominadas obtidas por 30 execuções do NSGA-II para o problema da treliça de 10 barras.	86
6.11	Ilustração dos grupos obtidos para o princípio DP1, juntamente com os valores obtidos para cada solução.	88
6.12	Ilustração da Frente de Pareto obtida para o problema das 10 barras, em que são destacadas as regiões correspondentes aos grupos mostrados na Figura 6.11.	88
6.13	Frentes de Pareto obtidas ao resolver os problemas da treliça de 10 barras original e simplificado (utilizando 6 variáveis ao invés das 10 originais). . .	91

LISTA DE TABELAS

3.1	Cálculo do expoente E em operações entre dois terminais T_1 e T_2	47
4.1	Exemplos de soluções triviais para <i>Innovization</i> que podem ser obtidas por uma PG.	51
6.1	Parâmetros adotados para o NSGA-II no problema da treliça de 2 barras. . . .	67
6.2	Parâmetros adotados para o <i>Innovization</i> no problema da treliça de 2 barras. .	68
6.3	Unidades das variáveis utilizadas no problema da treliça de 2 barras.	68
6.4	Resultados quantitativos obtidos no problema da treliça de 2 barras, considerando a utilização de arquivo externo.	69
6.5	Resultados obtidos para o estudo de caso da estrutura de 2 barras. As unidades básicas são massa (kg), comprimento (m) e comprimento(s).	70
6.6	Resultados obtidos em Bandaru e Deb (2013a) para o estudo de caso da estrutura de 2 barras. As unidades básicas são massa (kg), comprimento (m) e comprimento(s).	71
6.7	Parâmetros adotados para o <i>Innovization</i> no problema da viga soldada.	73
6.8	Unidades das variáveis utilizadas no estudo de caso da viga soldada.	74
6.9	Resultados quantitativos obtidos para o estudo de caso da viga soldada, considerando a utilização de arquivo externo.	74
6.10	Princípios obtidos para o estudo de caso da viga soldada. As unidades básicas são massa (lb), comprimento (in), tempo (s) e custo.	75
6.11	Resultados obtidos por Bandaru e Deb (2013a) para o estudo de caso da viga soldada.	76
6.12	Parâmetros adotados para o <i>Innovization</i> no estudo de caso do corte da barra de metal.	78
6.13	Unidades das variáveis utilizadas no estudo de caso do corte da barra metálica.	78
6.14	Resultados quantitativos obtidos para o estudo de caso do corte de uma barra metálica, considerando a utilização de arquivo externo.	79
6.15	Princípios obtidos para o estudo de caso do corte de uma barra metálica. . . .	80

6.16	Resultados obtidos por Bandaru e Deb (2013a) para o estudo de caso do corte de uma barra de metal.	81
6.17	Parâmetros adotados para o NSGA-II no problema das engrenagens compostas.	82
6.18	Resultados quantitativos obtidos para o estudo de caso das engrenagens compostas, considerando a utilização de arquivo externo.	83
6.19	Princípios obtidos para o estudo de caso das engrenagens compostas.	84
6.20	Parâmetros adotados para o NSGA-II no problema da treliça de 10 barras. . .	85
6.21	Resultados quantitativos obtidos para o estudo de caso da estrutura de 10 barras, considerando a utilização de arquivo externo.	87
6.22	Primeira parte dos princípios de projeto obtidos para o estudo de caso da estrutura de 10 barras. As dimensões básicas utilizadas são massa (lb) e comprimento(in).	89
6.23	Segunda parte dos princípios de projeto obtidos para o estudo de caso da estrutura de 10 barras. As dimensões básicas utilizadas são massa (lb) e comprimento(in).	90

1 INTRODUÇÃO

Nos últimos anos tem crescido o interesse de pesquisadores pela análise de pós-otimalidade, que consiste em descobrir características intrínsecas às soluções obtidas em problemas de otimização. De acordo com Deb e Srinivasan (2006), em problemas de engenharia pode-se esperar que existam similaridades entre as soluções de um problema de otimização, relacionadas às suas condições de otimalidade. O termo *Innovization* (inovação através de otimização, do inglês *innovation through optimization*) foi cunhado por Deb e Srinivasan (2006) para introduzir um processo de descoberta de conhecimento a partir das soluções de um problema de otimização, na forma de relações matemáticas entre variáveis, objetivos, restrições e parâmetros do problema. Tais relações, denominadas **princípios de projeto**, possibilitam (i) compreender novos aspectos do problema; (ii) criar novas (boas) soluções sem a necessidade de um novo processo de otimização; e (iii) informar um algoritmo de otimização, indicando regiões promissoras do espaço de busca.

Apesar da ideia de descoberta de conhecimento sobre as soluções do problema se estender ao caso mono-objetivo, a pluralidade de soluções usualmente obtida ao resolver-se problemas multiobjetivo (denominadas Soluções Pareto-Ótimas) permite a pesquisa por propriedades que possam revelar características do problema até então desconhecidas (Deb e Srinivasan 2006). No caso mono-objetivo, tal análise se reduz a verificações de sensibilidade. Quando são considerados problemas multiobjetivo, tem-se conteúdo suficiente para se descobrir relações entre as diversas características do problema (Deb e Srinivasan 2006).

Na proposta inicial de *Innovization*, a busca por princípios de projeto foi realizada manualmente, a partir da observação de regiões promissoras do espaço de busca, seguida pela identificação de correlações entre variáveis, objetivos ou restrições. Apesar disso, os autores apresentaram a utilidade da técnica em vários problemas de engenharia, descobrindo relações relevantes e não triviais. Por exemplo, considerando um problema multiobjetivo que consiste na otimização do projeto de uma treliça de duas barras, com os objetivos de minimizar o volume ocupado pela estrutura e a tensão máxima, os autores observaram que todas as soluções Pareto-ótimas possuem a mesma tensão nas duas barras. Além disso, a técnica foi capaz de encontrar neste problema todas as relações

analiticamente previstas. Este problema será apresentado com detalhes na Seção 5.2. Outros problemas de engenharia explorados por Deb e Srinivasan (2006) foram o projeto de um sistema de embreagem de múltiplos discos, o projeto de uma mola, o projeto de engrenagens acopladas e o de uma viga soldada.

A busca por regras em função dos parâmetros do problema foi também proposta por Deb e Srinivasan (2006), recebendo o nome de *Innovization* de alto nível. Neste caso, o problema é resolvido diversas vezes variando-se o valor de um dado parâmetro, buscando-se então padrões presentes em todos os conjuntos de soluções obtidos. Além da descoberta de regras de alto nível, os autores propuseram também o *Innovization* de baixo nível, que consiste em focar a busca em regiões promissoras do espaço de objetivos (Bandaru e Deb 2013b), obtendo regras que sejam relacionadas a um conjunto de soluções não-dominadas, de acordo com o interesse do tomador de decisão (TD).

O processo de *Innovization* foi estendido por Bandaru e Deb (2010), Bandaru e Deb (2011a), Bandaru e Deb (2011b), Bandaru e Deb (2013a) e Bandaru et al. (2015). No processo automatizado de *Innovization* (Bandaru e Deb 2010; Bandaru e Deb 2011a; Bandaru e Deb 2011b) utiliza-se uma meta-heurística para buscar princípios de projeto que revelem características das soluções Pareto-Ótimas. Um dos desafios da tarefa é a obtenção de diversas regras de projeto. Esse desafio foi solucionado pelos autores através do uso de um Algoritmo Genético com nichos, permitindo a existência de diferentes espécies na mesma população e, assim, soluções com características diferentes são obtidas na mesma execução da técnica. O espaço de busca consiste em um conjunto formado pelas variáveis de decisão, objetivos, restrições, além de funções adicionais que podem ser indicadas pelo usuário.

Um passo rumo à obtenção de regras de projeto mais gerais foi a adoção da Programação Genética (PG) no trabalho de Bandaru e Deb (2013a), técnica largamente utilizada para evolução de estruturas complexas, como expressões aritméticas e classificadores. Uma modificação foi necessária para utilizar a PG como mecanismo de busca: além de encontrar regras que reflitam invariâncias, os modelos encontrados devem relacionar corretamente as variáveis de decisão no que tange às unidades envolvidas. Assim, Bandaru e Deb (2013a) propõem o uso de uma PG em que a consistência das unidades é verificada durante o processo de busca, de forma a incentivar a geração de soluções candidatas válidas. A abordagem escolhida pelos autores foi a penalização, que

atribui um valor arbitrariamente alto às restrições da solução candidata que apresentar inconsistência no emprego das unidades.

Bandaru et al. (2015) propuseram uma generalização do processo de *Innovization* utilizando PG como mecanismo de busca, capaz de identificar princípios de projeto de baixo e alto níveis. Foram realizados experimentos utilizando o problema da treliça de duas barras, um problema de corte de metal e um modelo de gerenciamento de estoque e produção, que consiste na produção baseada em previsões, ao invés de demanda real. Assim, deseja-se manter um estoque de produtos suficiente para atender as demandas recebidas, de forma a maximizar as demandas atendidas, minimizar a produção por unidade de tempo e minimizar a quantidade de produtos armazenados em estoque, que implica em custos operacionais. A técnica foi capaz de identificar diversos princípios de projeto relevantes para o problema, por exemplo revelando quais variáveis de decisão possuem maior influência sobre cada objetivo.

Apesar do foco dos estudos na área ser a análise de soluções não-dominadas, a inclusão de dados de outras regiões do espaço de busca pode ser promissora, levando a descoberta de princípios de projeto que de outra forma seriam dificilmente obtidas (Chichakly e Eppstein 2013).

Além das extensões e aplicações da técnica de *Innovization* citados até aqui, diversos outros estudos independentes foram realizados com o objetivo de descobrir informações em problemas multiobjetivo baseando-se em técnicas de aprendizado de máquina (Obayashi e Sasaki 2003; Obayashi, Jeong e Chiba 2005; Chiba et al. 2006; Packham e Parmee 2000; Pryke, Mostaghim e Nazemi 2007; Ulrich, Brockhoff e Zitzler 2008; Ulrich 2013; Doncieux e Hamdaoui 2011; Taboada e Coit 2006; Sugimura et al. 2009; Oyama, Nonomura e Fujii 2010; Brownlee e Wright 2012; Kudo e Yoshikawa 2012) ou mesmo em métodos analíticos (Papalambros e Wilde 2000; Askar e Tiwari 2011).

Mapas auto-organizáveis de Kohonen (SOMs) (Kohonen 1990) foram utilizados por Obayashi e Sasaki (2003), Chiba et al. (2006) e Doncieux e Hamdaoui (2011) para extrair informações de Soluções Pareto-Ótimas. SOMs são um tipo de rede neural recorrente não supervisionada capaz de separar espacialmente dados multidimensionais em grupos com características semelhantes, mantendo os grupos mais correlacionados próximos entre si. Obayashi e Sasaki (2003) utilizaram SOMs para buscar por padrões em projetos de asa e fuselagem de aeronaves supersônicas, Além disso, a técnica foi empregada

para analisar o impacto de 131 variáveis de decisão nos objetivos do problema. Chiba et al. (2006) aplicaram a técnica para verificar o impacto das variáveis de decisão nos objetivos de um problema de otimização de ruído produzido por motores de aeronaves. SOMs também foram utilizadas por Doncieux e Hamdaoui (2011) para identificar padrões que afetam a velocidade final num projeto de asa de ornitóptero, que compõe uma aeronave que voa através do movimento de suas asas, de forma semelhante ao que ocorre com aves e morcegos.

Uma outra abordagem baseada em técnicas de visualização foi proposta por Pryke, Mostaghim e Nazemi (2007), onde *heatmaps* foram utilizados para visualizar simultaneamente os espaços de decisão e objetivos, possibilitando identificar correlações entre os mesmo. Uma vantagem do método proposto é que pode ser aplicado a problemas com mais de três objetivos.

Ulrich, Brockhoff e Zitzler (2008) propuseram o uso de dendrogramas para agrupar soluções não-dominadas para descoberta de princípios de projeto. Dendrogramas são árvore binárias, nas quais a distribuição dos nós separa os dados de forma hierárquica e ascendente, de tal forma que as soluções se encontram nas folhas da árvore e nós próximos tendem a apresentar informações relacionadas. A técnica foi aplicada com sucesso ao problema da mochila e ao projeto de um processador embarcado voltado para processamento de pacotes de redes de computadores.

Ulrich (2013) propôs uma formulação bi-objetivo ao problema de encontrar correlações entre variáveis de decisão e o espaço de objetivos, além de um algoritmo capaz de resolvê-lo denominado *Pareto-front Analyser* (PAN). A principal contribuição deste trabalho foi propor uma nova estratégia para relacionar as variáveis de decisão a regiões de interesse do espaço de objetivos, atuando como um algoritmo de agrupamento. Foram realizados testes comparativos com os algoritmos de agrupamento comumente utilizados na literatura. A técnica foi validada utilizando o problema da mochila e aplicada a um problema real de projeto de uma ponte em estrutura de treliça, de forma a suportar uma dada carga externa. Os objetivos considerados para o projeto da ponte foram a minimização da soma dos pesos das barras e a minimização da maior barra utilizada. De acordo com os resultados obtidos, a técnica foi superior às abordagens tradicionais, mostrando-se capaz de separar soluções com características semelhantes em grupos compactos. Apesar de realizar um agrupamento satisfatório nos problemas considerados, a técnica não oferece

uma forma automatizada de verificar as relações entre variáveis de decisão e objetivos, sendo necessária a inspeção visual dos projetos obtidos para descoberta de padrões.

Brownlee e Wright (2012) utilizaram planilhas eletrônicas para destacar e identificar manualmente padrões em conjuntos de soluções não-dominadas de um problema bi-objetivo. Os autores relataram dificuldade na tarefa devido ao grande número de soluções analisadas.

Para extração de conhecimento de soluções não-dominadas, Kudo e Yoshikawa (2012) propuseram a utilização do método de visualização *isomap*, originalmente criado para analisar problemas de classificação múltipla. A abordagem consiste em representar a distância das soluções no espaço de variáveis considerando sua distância no espaço dos objetivos. A técnica foi aplicada ao projeto de um motor foguete híbrido, que opera com uma mistura de propelentes líquidos e sólidos. De acordo com os autores do trabalho, este tipo de motor foguete é menos suscetível a explosões e possui menor impacto ambiental. Foram considerados 5 objetivos: minimização do peso e comprimento do foguete e da aceleração máxima; e maximização da altitude atingível e também da carga útil que pode ser transportada. No problema tratado foram encontradas algumas relações entre os valores das variáveis de decisão e os dos objetivos.

Apesar de possibilitar a descoberta de algumas inovações nos problemas em estudo, as técnicas citadas carecem de um método automatizado para obtenção de padrões inteligíveis, que é um dos objetivos primordiais do processo de *Innovization* (Bandaru et al. 2011).

Um método de busca denominado *multiobjective robust design exploration* foi proposto por Sugimura et al. (2009), que consiste em uma otimização multiobjetivo seguida de análise das soluções utilizando regras de associação, buscando-se correlações entre as variáveis de decisão e os objetivos. Regras de associação são expressões da forma “se-então” que podem ser utilizadas para inferir causalidade em dados categóricos. Aplicada ao problema de projeto de um ventilador centrífugo para máquinas de lavar e secar, com os objetivos de maximizar eficiência e minimizar ruídos, a técnica mostrou-se capaz de reproduzir os melhores resultados da literatura. Diversos princípios de projeto foram inferidos na forma de regras de associação, fornecendo informações relevantes ao projeto em estudo. Os autores destacam ainda a importância de se utilizar uma técnica automatizada para extração de padrões simbólicos, sobretudo para problemas com alta

dimensionalidade, onde a aplicação de técnicas de visualização torna-se difícil.

Dentre os métodos analíticos há a análise de monotonicidade, também capaz de obter princípios de projeto diretamente a partir da definição dos problemas (Papalambros e Wilde 2000). Entretanto, como o nome sugere, assume-se que objetivos e restrições sejam funções monotônicas em relação às variáveis de decisão, ou ainda que as funções objetivos sejam livres de ao menos uma das variáveis de decisão, o que limita seu escopo de utilização.

Askar e Tiwari (2011) apresentaram um outro método analítico capaz de obter diretamente os princípios de projeto a partir da formulação do problema multiobjetivo. A técnica foi aplicada com sucesso a dois problemas de engenharia, o projeto de uma viga soldada e o projeto de um sistema de drenagem, neste último revelando princípios até então desconhecidos. O método possui, entretanto, uma grande limitação: os princípios de projeto devem ser combinações lineares das variáveis de decisão. Além disso, é necessário conhecer a definição matemática do problema, o que em alguns casos não é possível (por exemplo quando as soluções candidatas são avaliadas por um simulador externo).

Propõe-se aqui algumas melhorias ao processo de *Innovization* baseado em PG, visando a obtenção de soluções mais diversas, possibilitando assim maior esclarecimento ao especialista de domínio sobre o problema abordado. Inicialmente, observa-se que há uma perda de princípios promissores ao longo do processo de busca. Neste sentido, propõe-se aqui a incorporação de um arquivo externo para armazenamento de todos os princípios de projeto potencialmente de interesse.

Além disso, dada a capacidade da PG em gerar modelos sem uma estrutura pré-definida, torna-se necessário introduzir um tratamento para que as operações aritméticas sejam adequadamente executadas, considerando as unidades dos termos envolvidos. Nota-se que as unidades das variáveis são conhecidas *a priori*. Neste trabalho é proposta uma solução alternativa para tratamento da consistência entre unidades, que envolve realizar operações protegidas, ignorando os termos inválidos das expressões. A utilização de operações protegidas é comumente adotada na PG para tratar a execução de operações aritméticas envolvendo valores inválidos. Por exemplo, é comum utilizar uma operação protegida de divisão, retornando um valor fixo quando o denominador é igual a zero.

Verifica-se também que são gerados princípios de projeto, que dada sua simplicidade não agregam conhecimento do domínio do problema, sendo portanto irrelevantes. Tais

soluções são chamadas aqui de soluções triviais. Propõe-se aqui um procedimento para dificultar a manutenção deste tipo de princípio na população.

As técnicas propostas aqui são aplicadas a cinco estudos de caso em engenharia, sendo quatro deles já abordados na literatura de *Innovization*, envolvendo os projetos de: uma treliça de duas barras, uma viga soldada, o corte de uma barra de metal e engrenagens compostas. Além destes, também será proposta aqui a análise de uma estrutura de treliça de 10 barras, comumente utilizada em problemas de otimização mono e multiobjetivo. Apesar de largamente utilizada como um problema teste, não há na literatura trabalhos envolvendo a descoberta de conhecimento desta estrutura.

1.1 Organização da dissertação

Este trabalho está organizado da seguinte forma: o Capítulo 2 apresenta conceitos básicos de otimização multiobjetivo, imprescindíveis à apreciação das técnicas utilizadas aqui. No Capítulo 3 a técnica de *Innovization* é introduzida, havendo também uma breve introdução à Programação Genética. No Capítulo 4 são delineados os métodos propostos aqui para *Innovization* baseado em Programação Genética. Os estudos de caso tratados são apresentados no Capítulo 5. Os resultados dos experimentos computacionais desses problemas são apresentados e analisados no Capítulo 6. Concluindo este trabalho, o Capítulo 7 possui as considerações finais e trabalhos futuros.

2 OTIMIZAÇÃO MULTIOBJETIVO

2.1 INTRODUÇÃO

A otimização tem por objetivo a obtenção de soluções que sejam ótimas em relação a determinados objetivos, como redução de custo ou tempo de fabricação de um produto, satisfação de requisitos de segurança, etc. Quando se deseja modelar e resolver problemas reais, não é rara a existência de múltiplos objetivos, os quais muitas vezes são conflitantes e devem ser otimizados simultaneamente. Neste capítulo serão apresentados conceitos e definições relacionados à tais problemas, denominados problemas de otimização multiobjetivo.

2.2 CONCEITOS BÁSICOS

Um Problema de Otimização (PO) pode ser definido de forma geral como

$$\begin{aligned}
 \text{Min/Max} \quad & f_m(\mathbf{x}), & m = 1, 2, \dots, M; \\
 \text{sujeito a} \quad & g_j(\mathbf{x}) \geq 0, & j = 1, 2, \dots, J; \\
 & h_k(\mathbf{x}) = 0, & k = 1, 2, \dots, K; \\
 & x_i^I \leq x_i \leq x_i^S, & i = 1, 2, \dots, N;
 \end{aligned} \tag{2.1}$$

onde o vetor \mathbf{x} corresponde às variáveis de decisão, $f_m : \mathbb{R}^N \rightarrow \mathbb{R}$ são as funções objetivo, $h_k : \mathbb{R}^N \rightarrow \mathbb{R}$ as restrições de igualdade e $g_j : \mathbb{R}^N \rightarrow \mathbb{R}$ as restrições de desigualdade que devem ser satisfeitas. As soluções são vetores da forma $\mathbf{x} = (x_1, x_2, \dots, x_N)^T$, sendo cada dimensão denominada variável de decisão. O último conjunto de restrições em (2.1) são os limites das variáveis de decisão, de forma que cada variável x_i pode assumir valores entre o limite inferior x_i^I e o superior x_i^S . Tais limites definem o espaço das variáveis de decisão, $D \subseteq \mathbb{R}^N$. Sem perda de generalidade, aqui considera-se que todos os POs são do tipo minimização, de acordo com o princípio da dualidade, que possibilita transformar

POs de maximização em POs de minimização multiplicando-se as funções objetivo por -1 .

Quando o PO possui apenas um objetivo ($M = 1$), este é chamado de problema de otimização mono-objetivo. Caso contrário, quando há mais de uma função objetivo ($M > 1$), o PO é denominado problema de otimização multiobjetivo.

Quando trabalhando com meta-heurísticas e o PO possui apenas o conjunto de restrições que define o espaço de busca, este é denominado PO sem restrições. Isto se deve ao fato de que tais restrições são comumente tratadas de forma trivial quando esse tipo de técnica é utilizada. Entretanto, grande parte dos POs possui outras restrições, sobretudo quando se considera a modelagem de problemas reais. Neste caso, as soluções que satisfazem todas as restrições são chamadas soluções viáveis (ou factíveis), enquanto aquelas que violam ao menos uma restrição são denominadas inviáveis (ou infactíveis). O conjunto de todas soluções viáveis define o espaço viável $S \subseteq D$. Para cada solução $\mathbf{x} \in D$ há um ponto z no espaço de objetivos \mathbb{Z} , obtido pelo mapeamento $z = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x}))$. A Figura 2.1 mostra o mapeamento entre os espaços de decisão e objetivos de um problema contendo três variáveis de decisão e duas funções objetivo.

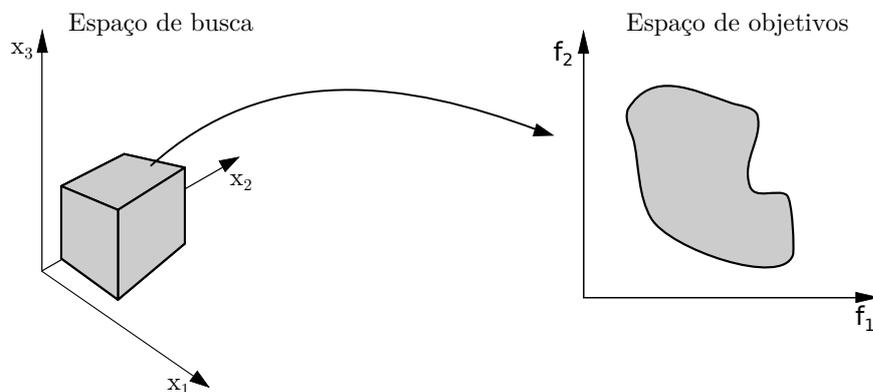


Figura 2.1: Mapeamento entre os espaços de decisão e objetivos.

De acordo com Deb (2001), a característica que mais diferencia problemas de otimização mono-objetivo e multiobjetivo é a definição de otimalidade. Ao resolver um problema mono-objetivo, deseja-se geralmente obter uma solução que apresente o melhor valor possível (ótimo) para a função objetivo, respeitando-se as restrições¹. Contudo,

¹Uma exceção são os problemas multimodais, nos quais pode-se desejar encontrar múltiplas soluções aceitáveis, ao invés de uma única solução ótima.

quando se trata de problemas com vários objetivos, o conceito de otimalidade não é trivial, uma vez que estes são conflitantes. Assim, é necessário introduzir um critério que permita comparar as soluções de um problema multiobjetivo. O critério de comparação mais comumente adotado na literatura é a dominância de Pareto, definida a seguir (Deb 2001).

Definição 1. Considerando um problema de minimização contendo M objetivos, uma solução \mathbf{X}_1 domina outra \mathbf{X}_2 se as seguintes condições são satisfeitas:

1. \mathbf{X}_1 não é pior que \mathbf{X}_2 em nenhum objetivo, ou seja, $f_i(\mathbf{X}_1) \leq f_i(\mathbf{X}_2)$, para $i = 1, 2, \dots, M$;
2. \mathbf{X}_1 é estritamente melhor que \mathbf{X}_2 em ao menos um objetivo, isto é, $f_i(\mathbf{X}_1) < f_i(\mathbf{X}_2)$ para algum i entre 1 e M .

Utilizando tal definição, três situações podem ocorrer ao analisar duas soluções \mathbf{X}_1 e \mathbf{X}_2 (Deb 2001):

1. \mathbf{X}_1 domina \mathbf{X}_2 ;
2. \mathbf{X}_2 domina \mathbf{X}_1 ;
3. \mathbf{X}_1 e \mathbf{X}_2 são soluções não-dominadas entre si.

As soluções viáveis não-dominadas por nenhuma outra no espaço de busca viável são chamadas de Soluções Pareto-Ótimas, também conhecidas como Conjunto Pareto-Ótimo (Coello, Van Veldhuizen e Lamont 2002). Frente de Pareto Ótima é o nome dado à imagem do Conjunto Pareto-Ótimo no espaço de objetivos do problema (Coello, Van Veldhuizen e Lamont 2002).

Deb (2001) conjectura duas metas para a otimização multiobjetivo:

- **Convergência:** Encontrar um conjunto de soluções não-dominadas tão próximas quanto possível da Frente de Pareto Ótima;
- **Diversidade:** Encontrar um conjunto de soluções não-dominadas tão diversas quanto possível.

A primeira meta, presente em qualquer processo de otimização, consiste em encontrar soluções próximas à solução ótima, que no caso multiobjetivo corresponde à Frente de

Pareto Ótima. A segunda se deve à impossibilidade de se escolher uma dentre as soluções Pareto-ótimas, pois, sem considerar informações adicionais sobre o problema, todos os objetivos são igualmente importantes. Logo, é desejável que o algoritmo de busca adotado seja capaz de encontrar um conjunto de soluções distintas, que serão analisadas *a posteriori* pelo tomador de decisão (TD). As duas metas são ilustradas na Figura 2.2.

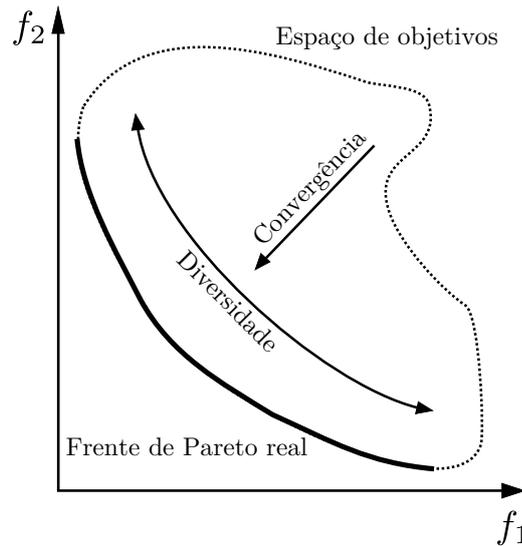


Figura 2.2: Convergência e diversidade ilustradas num problema com dois objetivos.

Dentre as soluções de um problema multiobjetivo há algumas que são comumente utilizadas por algoritmos de otimização, definidas a seguir.

Definição 2. Um vetor em que cada componente possui o valor ótimo para o objetivo correspondente em todo o espaço de busca é chamado de **solução ideal**, consistindo num limite inferior para qualquer solução (Deb 2001).

Definição 3. Um vetor em que cada componente possui o pior valor do objetivo correspondente dentre as soluções da Frente de Pareto é denominado **solução nadir** (Deb 2001). É importante notar que, diferentemente do vetor ideal, a solução nadir é definida sobre as soluções presentes na Frente de Pareto e não sobre todo o espaço de busca.

Diversos métodos foram propostos na literatura para a resolução de problemas multiobjetivo (Zadeh 1963; Haimes, Lasdon e Wismer 1971; Schaffer 1985; Srinivas e Deb 1994; Palli et al. 1998; Deb et al. 2002). Os chamados métodos clássicos consistem em transformar os objetivos do problema em somente um, que é então resolvido utilizando uma técnica de otimização mono-objetivo (Deb 2001). No método da soma

ponderada (Zadeh 1963), a função objetivo é a soma dos objetivos, cada um multiplicado por um peso indicado pelo usuário. Outra abordagem encontrada na literatura é manter um dos objetivos do problema original, tratando os demais como restrições do problema de otimização (Haimes, Lasdon e Wismer 1971).

De acordo com Deb (2001), os métodos clássicos apresentam algumas desvantagens. Inicialmente, a diversidade não é incentivada, pois uma única solução é obtida como solução do problema mono-objetivo, sendo necessário executar os métodos diversas vezes para obtenção de soluções (potencialmente) distintas. Uma dificuldade adicional é que nem todas as soluções do espaço de busca original podem ser obtidas quando o problema não é convexo. Finalmente, os métodos requerem informação adicional sobre o problema, como preferência entre os objetivos ou valores alvo para orientar a busca.

Desta forma, tornou-se mais usual resolver problemas multiobjetivo de forma heurística (Coello, Van Veldhuizen e Lamont 2002). Os algoritmos evolucionistas (AE) são técnicas atrativas para este fim, uma vez que o processo de busca utiliza um conjunto de soluções candidatas (Deb 2001). Assim, um AE tradicional pode ser adaptado para preservar a diversidade de soluções e também para convergir em direção à Frente de Pareto real. Os AEs adaptados para a resolução de problemas multiobjetivo são conhecidos como MOEAs (algoritmos evolucionistas multiobjetivo).

Para adaptar um AE para problemas multiobjetivo, é necessário introduzir uma forma de classificar as soluções candidatas. A maioria dos MOEAs da literatura utiliza o método da Ordenação de Pareto (Deb 2001), inicialmente sugerido por Golberg (1989), para esta finalidade. O método consiste em classificar a população em postos, de acordo com o nível de não-dominância das soluções. As soluções não-dominadas constituem o posto 0; aquelas dominadas apenas pelas soluções do posto 0 formam o posto 1, e assim sucessivamente. Durante o processo de busca, as soluções de posto mais baixo são escolhidas, uma vez que estas dominam as outras de posto mais alto. Na Figura 2.3 pode-se observar a Ordenação de Pareto para um conjunto de soluções de um problema com dois objetivos.

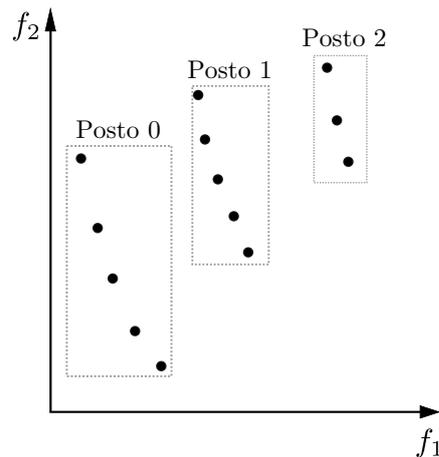


Figura 2.3: Exemplo de Ordenação de Pareto, em que 13 soluções são classificadas em 3 postos.

2.3 ALGORITMOS EVOLUCIONISTAS PARA PROBLEMAS MULTIOBJETIVO

Meta-heurísticas são métodos que combinam procedimentos de busca local com estratégias de mais alto nível, de modo a criar um processo capaz de escapar de ótimos locais e realizar uma busca robusta no espaço de soluções de um problema (Glover e Kochenberger 2003). Com o passar do tempo, esta definição passou a englobar todas as técnicas que utilizam estratégias para escapar de ótimos locais em espaços de busca complexos (Glover e Kochenberger 2003).

Algumas meta-heurísticas possuem inspiração natural, ou seja, seu procedimento é baseado em algum comportamento observado na natureza. Os algoritmos evolucionistas (AEs) pertencem a esta categoria e consistem técnicas baseadas na Teoria da Evolução das Espécies de Charles Darwin (Darwin 1859). Na literatura pode-se encontrar outros exemplos de meta-heurísticas bioinspiradas, como Otimização por Colônias de Formigas (Colorni, Dorigo e Maniezzo 1991), Otimização por Enxame de Partículas (Kennedy e Eberhart 1995), Sistemas Imunológicos Artificiais (Farmer, Packard e Perelson 1986), entre outros.

Algoritmos Genéticos (AGs) (Holland 1975) são AEs em que uma população de indivíduos (também chamados de soluções candidatas) passa por processos análogos àqueles observados na seleção natural. Através de um processo iterativo, os melhores indivíduos da população (denominados mais aptos) são selecionados e passam por um

processo de recombinação e mutação, dando origem a uma nova população de soluções. O Algoritmo 1 ilustra este processo.

Algoritmo 1: Pseudo-código de um AG clássico.

```

1 início
2    $pop_{corrente} \leftarrow \text{InicializaPopulação}();$ 
3   Avaliação( $pop_{corrente}$ );
4   enquanto Critério de parada não for atingido faça
5      $indivíduos_{selecionados} \leftarrow \text{Seleção}(pop_{corrente});$ 
6      $pop_{filhos} \leftarrow \text{Recombinação}(indivíduos_{selecionados});$ 
7     Mutação( $pop_{filhos}$ );
8     Avaliação( $pop_{filhos}$ );
9      $pop_{corrente} \leftarrow \text{Substituição}(pop_{corrente}, pop_{filhos});$ 
10  fim enquanto
11 fim

```

Cada indivíduo é formado por um genótipo, que é a codificação de uma solução candidata para o problema. Na proposta original, a solução é codificada em um vetor binário. Contudo, outras codificações foram propostas, como a codificação real, recomendada para problemas com parâmetros contínuos (Michalewicz 1996).

Além da codificação/decodificação das soluções numa dada representação, outra etapa dependente do problema é a função de avaliação, na qual a aptidão de um indivíduo é calculada.

Um processo de seleção é empregado para determinar os indivíduos que irão passar por recombinação, em que os indivíduos mais aptos têm maiores chances de serem escolhidos. Uma das abordagens de seleção comumente adotadas na literatura é o torneio, no qual uma competição é realizada entre um número determinado de indivíduos tomados aleatoriamente, sendo escolhido aquele com a maior aptidão. Aos indivíduos selecionados são aplicados os operadores genéticos de recombinação e mutação.

O operador de recombinação realiza o cruzamento entre os genótipos dos indivíduos selecionados. Geralmente a recombinação é aplicada a cada par de indivíduos, dando origem a 2 novos indivíduos. A recombinação é realizada de forma probabilística, de acordo com um parâmetro do AG (probabilidade de recombinação). Caso não seja aplicado, os 2 filhos gerados são clones de cada pai. São realizados N torneios, sendo N o tamanho da população.

O operador de mutação é aplicado a todos os indivíduos gerados na etapa de recombinação, consistindo em uma perturbação do genótipo.

Após a criação de uma nova população, esta tem sua aptidão calculada e ocorre a substituição da população anterior. Na política de substituição geracional, a nova população substitui a geração anterior, passando a ser a população corrente. Pode-se também fazer uso de elitismo, que consiste em manter na população as melhores soluções candidatas presentes na população anterior.

Diferentes critérios de parada podem ser utilizados, como um número máximo de gerações ou um valor de aptidão a ser atingido. Enquanto este critério não for satisfeito, novas gerações são operadas a partir da seleção de indivíduos mais aptos, recombinação e mutação.

2.3.1 NSGA-II

Um dos MOEAs mais conhecidos e utilizados na literatura de otimização multiobjetivo é o *Nondominated Sorting Genetic Algorithm II* (NSGA-II), proposto por Deb et al. (2002), que consiste na adaptação de um AG para resolver problemas multiobjetivo. Desenvolvido como uma extensão do método desenvolvido e apresentado em Srinivas e Deb (1994), o NSGA-II tornou-se popular por introduzir um algoritmo de baixo custo computacional para a Ordenação de Pareto, além de uma estratégia sem parâmetros para promoção da diversidade da população.

Um pseudo-código da Ordenação de Pareto é mostrado no Algoritmo 2. Inicialmente, para cada solução p dois itens são computados: (i) o número de soluções que dominam p , utilizando o critério de dominância de Pareto ($Dominam_p$); e (ii) um conjunto das soluções dominadas por p ($Dominados_p$). A Frente de Pareto F_1 é então formada pelas soluções não-dominadas, que são as soluções p tais que $Dominam_p = 0$. Para classificar as outras Frentes de Pareto, uma ideia é remover as soluções de F_1 do conjunto de soluções candidatas e repetir o processo anterior. Entretanto, tal procedimento seria computacionalmente custoso. Ao invés disso, a Frente de Pareto subsequente (F_{i+1}) é formada da seguinte forma: examina-se cada solução p presente na Frente de Pareto atual (linha 20); e para cada solução q dominada por p , decrementa-se em uma unidade o número de soluções que dominam q . As soluções q não-dominadas formam então a segunda Frente de Pareto. É importante notar que tal procedimento possibilita classificar as soluções sucessivamente, sem a necessidade de verificar as relações de dominância entre todas as soluções novamente.

Algoritmo 2: Pseudo-código do procedimento de Ordenação de Pareto. Adaptado de Deb et al. (2002).

Input: Conjunto de soluções P
Output: Frentes de Pareto $F = (F_1, F_2, \dots, F_l)$,
 $postos = (posto_1, posto_2, \dots, posto_n)$

```

1 início
2   para cada Solução  $p \in P$  faça
3      $Dominam_p \leftarrow 0$ ;
4      $Dominados_p \leftarrow \emptyset$ ;
5     para cada Solução  $q \in P$  faça
6       se  $q$  domina  $p$  então
7          $Dominam_p \leftarrow Dominam_p + 1$ ;
8       fim se
9       senão se  $p$  domina  $q$  então
10         $Dominados_p \leftarrow Dominados_p \cup q$ ;
11      fim se
12    fim para cada
13    se  $Dominam_p = 0$  então
14       $F_1 \leftarrow F_1 \cup p$ ;
15       $posto_p \leftarrow 1$ ;
16    fim se
17     $i \leftarrow 1$ ;
18    enquanto  $F_i \neq \emptyset$  faça
19       $F_{i+1} \leftarrow \emptyset$ ;
20      para cada Solução  $p \in F_i$  faça
21        para cada Solução  $q \in Dominados_p$  faça
22           $Dominam_q \leftarrow Dominam_q - 1$ ;
23          se  $Dominam_q = 0$  então
24             $posto_q \leftarrow i + 1$ ;
25             $F_{i+1} \leftarrow F_{i+1} \cup q$ ;
26          fim se
27        fim para cada
28      fim para cada
29       $i \leftarrow i + 1$ ;
30    fim enquanto
31  fim para cada
32 fim

```

Conforme delineado na seção anterior, muitos MOEAs utilizam o posto para ordenar as soluções durante o processo de busca. Tal estratégia favorece a convergência das técnicas. Entretanto, além da convergência, espera-se que os MOEAs apresentem diversidade de soluções. Uma questão que surge neste contexto é como selecionar soluções mais adequadas da mesma Frente de Pareto durante a busca. A estratégia adotada no NSGA foi utilizar um valor real para estimar a densidade de soluções em uma região circular

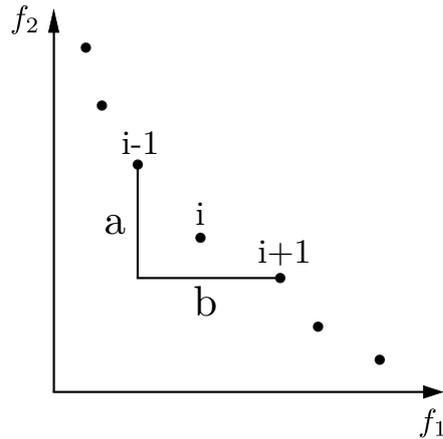


Figura 2.4: Exemplo do cômputo do CD para uma solução i de uma Frente de Pareto, dado neste caso por $a + b$. Figura adaptada de Deb et al. (2002).

do espaço de objetivos, centrada em determinada solução (Srinivas e Deb 1994). A desvantagem desta técnica é a necessidade de se especificar o raio dessa vizinhança.

No NSGA-II este problema foi resolvido com uma nova métrica para estimação da densidade de soluções, denominada *Crowding Distance* (CD), que consiste na distância entre as duas soluções mais próximas em relação a cada objetivo. Logo, quanto maior o CD de uma solução, menos densa é a região em que a mesma se encontra na Frente de Pareto. As soluções localizadas nos extremos da Frente, isto é, aquelas que apresentam os melhores valores para cada objetivo individualmente, possuem CD igual a “infinito”, indicando que são prioritárias em relação às soluções intermediárias, no sentido de preservar a diversidade. Na Figura 2.4 é apresentado um exemplo do cálculo do CD para uma solução i de um problema com dois objetivos, em que o valor o CD é dado por $a + b$.

Além de não necessitar de parâmetros, o cálculo do CD apresenta baixo custo computacional. Um pseudo-código para o cálculo do CD é mostrado no Algoritmo 3, que recebe como entradas as soluções de uma Frente de Pareto (S) e seu número de elementos. Inicialmente o valor de CD é inicializado, atribuindo-se o valor “infinito” para as soluções localizadas nos extremos da Frente Pareto e 0 para as demais. Depois ordena-se o conjunto em relação a cada objetivo, incrementando o CD de cada solução com a distância entre seus vizinhos mais próximos. As distâncias são calculadas utilizando os valores normalizados dos objetivos, de forma que objetivos com diferentes ordens de magnitude possuam a mesma influência no valor total do CD.

Deb et al. (2002) definiram um operador de comparação a partir do posto e do CD das

Algoritmo 3: Pseudo-código para o cálculo do CD. Adaptado de Deb et al. (2002).

Input: S : Soluções de uma Frente de Pareto, N : Número de soluções de S ,
 M : Número de objetivos
Output: CD : *Crowding distance* de cada solução

```

1 início
2   para  $i \leftarrow 1$  até  $N$  faça
3     |  $CD[i] \leftarrow 0$ ;
4   fim para
5   para  $i \leftarrow 1$  até  $M$  faça
6     | Ordena( $S, i$ );
7     |  $CD[1] \leftarrow CD[N] \leftarrow \infty$ ;
8     |  $f_{min} \leftarrow S[1].objetivos[i]$ ;
9     |  $f_{max} \leftarrow S[N].objetivos[i]$ ;
10    | para  $j \leftarrow 2$  até  $N - 1$  faça
11      |  $CD[j] \leftarrow$ 
12      |  $CD[j] + (S[j + 1].objetivos[i] - S[j - 1].objetivos[i]) / (f_{max} - f_{min})$ ;
13    fim para
14 fim

```

soluções candidatas, que é utilizado nas etapas de seleção e substituição da população. Entre duas soluções contidas em postos distintos, seleciona-se a de menor posto. Caso contrário, é escolhida aquela com maior CD.

Agora que suas principais características foram comentadas, pode-se apresentar completamente o algoritmo NSGA-II. Inicialmente, uma população de tamanho N é criada de forma aleatória. Uma ordenação de Pareto é então realizada sobre a população inicial, gerando um conjunto de Frentes de Pareto. Os operadores de seleção, recombinação e mutação clássicos dos AGs são então aplicados para gerar a próxima população. A seleção utiliza o operador de comparação supracitado. A população total, formada pela união da anterior com a atual, é ordenada de acordo com a não-dominância. A seleção dos indivíduos que irão formar a próxima população se dá selecionando-se as solução de postos mais baixos, até completar N indivíduos. Caso seja necessário escolher dentre indivíduos do mesmo posto, são escolhidos aqueles de maior CD. Este procedimento é ilustrado na Figura 2.5.

Quando se utiliza codificação real, os operadores genéticos recomendados para o NSGA-II são a recombinação binária simulada (SBX) e a mutação polinomial (Deb et al. 2002), que foram portanto utilizados neste trabalho. Dadas duas soluções selecionadas como pais (denominadas aqui P_1 e P_2), a recombinação utilizando o operador SBX se

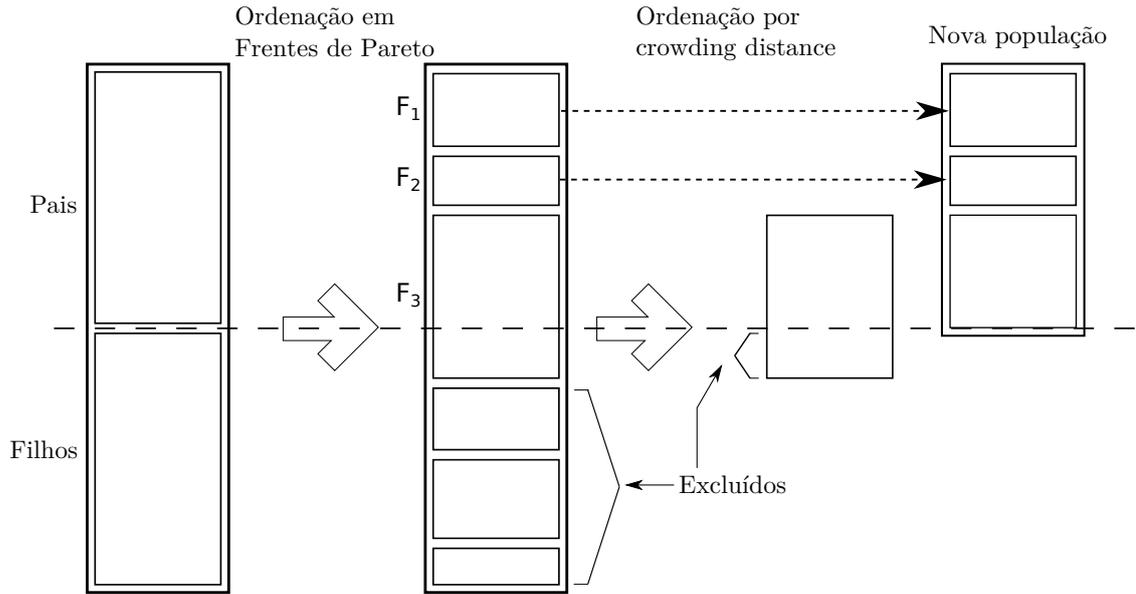


Figura 2.5: Representação da geração da próxima população de uma iteração do algoritmo NSGA-II, a partir da população atual (*Filhos*) e da população anterior (*Pais*). Figura adaptada de Deb et al. (2002)

dá da seguinte forma. Para cada variável i um número aleatório $u \in [0,1]$ é gerado. Os valores de i das novas soluções (denominadas F_1 e F_2) são gerados a partir das seguintes expressões (Deb e Goyal 1996):

$$F_1[i] = 0,5\{(1 + \beta)P_1[i] + (1 - \beta)P_2[i]\}$$

$$F_2[i] = 0,5\{(1 - \beta)P_1[i] + (1 + \beta)P_2[i]\}$$

O valor de β é calculado a partir de uma função de distribuição de probabilidade e é dado por:

$$\beta = \begin{cases} (2u)^{\frac{1}{n_c+1}}, & \text{se } u \leq 0,5, \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{n_c+1}}, & \text{caso contrário} \end{cases},$$

onde n_c é um parâmetro real informado pelo usuário, denominado índice de distribuição de recombinação.

Caso ocorra mutação na variável de índice j de um indivíduo $S = (x_1, x_2, \dots, x_n)$, um número aleatório $u \in [0,1]$ é gerado e S_j tem seu valor modificado para $S_j + \sigma_j$, dado por (Deb e Goyal 1996):

$$\delta_j = \begin{cases} [(2u)^{\frac{1}{n_m+1}}] - 1, & \text{se } u < 0,5, \\ 1 - [2(1-u)]^{\frac{1}{n_m+1}}, & \text{caso contrário} \end{cases},$$

onde n_m é um parâmetro real (índice de distribuição de mutação).

2.4 MEDIDAS DE DESEMPENHO

Na literatura pode-se encontrar diversas medidas de desempenho para avaliação da qualidade MOEAs, que consideram tanto a diversidade quanto a convergência do conjunto encontrado de soluções. Neste trabalho o hipervolume foi adotado e é descrito na próxima seção.

2.4.1 Hipervolume

A métrica do hipervolume (H) foi proposta por Zitzler e Thiele (1999) e é muito popular na literatura, sobretudo por não exigir conhecimento sobre a Frente de Pareto real. Seu valor é calculado como o hipervolume dos hipercubos definidos pelas Soluções Pareto-Ótimas e um ponto de referência W . O ponto W pode ser obtido utilizando os piores valores encontrados para cada função objetivo (Deb 2001). Na Figura 2.6 é mostrado um exemplo para um conjunto de soluções não-dominadas de um problema com dois objetivos, no qual o H é calculado como a área da região sombreada.

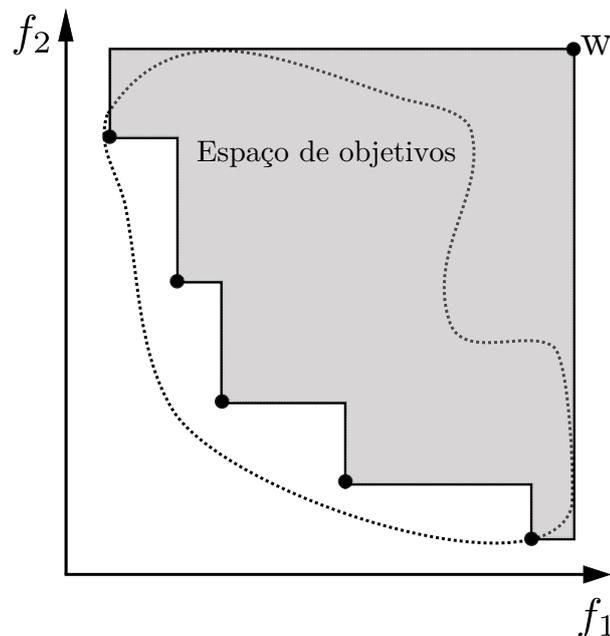


Figura 2.6: Exemplo de cálculo do hipervolume H para um conjunto de 5 soluções não-dominadas em um problema com dois objetivos a serem minimizados. W é o ponto de referência. H corresponde à área da região sombreada. Figura baseada em Deb (2001).

3 *INNOVIZATION*

3.1 INTRODUÇÃO

A solução ótima de um problema multiobjetivo consiste, tipicamente, de um conjunto de soluções não-dominadas entre si, usualmente chamado de Frente de Pareto (Deb 2001). Idealmente, tal conjunto representa as melhores soluções atingíveis, levando-se em consideração o compromisso entre os objetivos, que neste caso são conflitantes. Após o processo de otimização, o TD tem a responsabilidade de escolher a solução que de fato melhor atende aos seus requisitos. Nos últimos anos, tem crescido o interesse de pesquisadores pela análise dos resultados encontrados num processo de otimização, que consiste em descobrir características adicionais intrínsecas às soluções obtidas. Enquanto no caso mono-objetivo tal análise se reduz a verificações de sensibilidade, a pluralidade de soluções observada em problemas multiobjetivo permite a pesquisa por propriedades que possam revelar características do problema até então desconhecidas (Deb e Srinivasan 2006).

O termo *Innovization* foi criado por Deb e Srinivasan (2006) para definir um processo de busca no qual, a partir das soluções de um problema de otimização, deseja-se encontrar princípios de projetos, que são relações matemáticas entre variáveis, objetivos, restrições e parâmetros do problema. Tais relações possibilitam (i) compreender novos aspectos do problema; (ii) criar novas (boas) soluções sem a necessidade de um novo processo de otimização; e (iii) informar um algoritmo de otimização, indicando regiões promissoras do espaço de busca.

Para extrair informações úteis, é desejável que as soluções não-dominadas correspondam à Frente de Pareto ótima, que é desconhecida na quase totalidade dos problemas. Deb e Srinivasan (2006) adotaram métodos para assegurar que a Frente de Pareto obtida é uma boa aproximação para a Frente de Pareto Ótima. As etapas do processo de *Innovization* definidas nesta referência e utilizadas aqui são enumeradas a seguir:

1. Inicialmente são obtidas as melhores soluções considerando cada objetivo separadamente, utilizando um algoritmo de otimização mono-objetivo, como um

- AG ou mesmo um método clássico;
2. Deve-se então encontrar as soluções ideal e nadir;
 3. O algoritmo NSGA-II é utilizado para encontrar o conjunto de soluções não-dominadas;
 4. Todos os objetivos são normalizados utilizando os pontos ideal e nadir;
 5. Um procedimento de busca local deve ser aplicado para a obtenção de uma frente de Pareto otimizada. Deb e Srinivasan (2006) adotaram o método de Benson (Benson 1978);
 6. Utiliza-se o método das restrições normais (NCM) (Messac e Mattson 2004) a partir de alguns pontos para verificar se a frente de Pareto encontrada é uma boa aproximação para a Frente de Pareto real;
 7. Finalmente, a Frente de Pareto é analisada para descoberta de regras de projeto.

3.1.1 *Processo automatizado de Innovization*

Apesar de apresentar resultados promissores, a proposta inicial de *Innovization* carece de um processo automatizado para busca por princípios de projeto, sendo necessária a inspeção visual da relação entre variáveis/objetivos/restrições. Pode-se perceber que isto limita a aplicação da técnica, sobretudo quando são considerados problemas com alta dimensionalidade.

Neste sentido, um processo automatizado de *Innovization* foi desenvolvido por Bandaru e Deb (2010), Bandaru e Deb (2011a) e Bandaru e Deb (2011b) e consiste em utilizar um AG na busca de invariâncias, isto é, de princípios de projeto que sejam válidos para toda a Frente de Pareto e, assim, revelem características inerentes às soluções Pareto-ótimas. O espaço de busca é formado pelas variáveis de decisão, objetivos e restrições de um problema multiobjetivo, além de dados adicionais que podem ser indicadas pelo usuário. Assim, deseja-se encontrar funções da forma $\Psi_i(\mathbf{x}, \mathbf{f}(\mathbf{x}), \mathbf{g}(\mathbf{x})) = c_i$, onde c_i é uma constante associada ao princípio Ψ_i .

De acordo com Bandaru e Deb (2010), os princípios usualmente observados em problemas de engenharia pode ser representada pela estrutura genérica mostrada na

Equação 3.1, na qual os termos ϕ_j são as entradas (variáveis de decisão, objetivos e restrições). Para descobrir o princípio de projeto Ψ_i , deve-se encontrar as potências b_{ij} e calcular c_i .

$$\Psi_i(\mathbf{x}, \mathbf{f}(\mathbf{x}), \mathbf{g}(\mathbf{x})) = \prod_{j=1}^N \phi_j(\mathbf{x})^{b_{ij}} = c_i \quad (3.1)$$

Contudo, duas situações podem dificultar a solução de 3.1. Inicialmente, os princípios podem não ser válidos para todos os pontos, caso a Frente de Pareto utilizada não seja a Frente de Pareto Ótima. Assim, o valor c_i pode não ser constante, sendo necessária uma alternativa estatística para testar a validade dos princípios. Para este fim, os autores da técnica utilizaram o coeficiente de variação dos valores c_i . O coeficiente de variação (c_v) de um conjunto é calculado como na Equação 3.2, onde σ e μ representam, respectivamente, o desvio padrão e a média dos valores do conjunto.

$$c_v = \frac{\sigma}{\mu} \quad (3.2)$$

Além disso, mesmo se uma Frente de Pareto acurada for utilizada, pode haver princípios válidos apenas em algumas regiões contíguas. Para resolver os dois problemas, um algoritmo de agrupamento foi proposto, com o objetivo de encontrar princípios que (i) sejam válidos na maior parte da Frente de Pareto e (ii) possuam variação numérica mínima, isto é, aproximando-se de um valor constante c_i . O objetivo (ii) é reforçado por uma restrição, utilizando o conceito de significância de um princípio de projeto (S), que consiste no percentual da Frente de Pareto em que este é aplicável.

Durante o processo de busca, a avaliação da qualidade dos princípios se dá da seguinte forma: (i) os princípios são executados para cada solução não-dominada i (Equação 3.1), gerando um valor c_i (chamado daqui em diante de c-valor); (ii) os c-valores são ordenados em ordem crescente; (iii) o agrupamento é executado sobre o conjunto de valores c_i . O objetivo da busca é minimizar o número de grupos (C), o número de pontos não pertencentes a nenhum grupo (U) e o somatório dos coeficientes de variação de cada grupo (c_v).

Ao invés de um problema de otimização com três objetivos, os autores utilizaram uma abordagem de soma ponderada dos objetivos e, assim, o problema pode ser definido pela Equação 3.3, onde índice i indica o i -ésimo princípio de projeto, d_i são números inteiros

que representam a quantidade de regiões utilizadas pelo algoritmo de agrupamento ao agrupar os c -valores desta solução e $b_{ij}, j = 1, \dots, N$ são números reais que representam os expoentes de cada termo ϕ_j . A primeira restrição, $U = 0$ foi adicionada para direcionar a busca para princípios que sejam aplicáveis à maior parte da Frente de Pareto. A última restrição, $|b_{ij}| \geq 0,1$, garante que a solução trivial do problema ($b_{ij} = 0 \forall j$) não seja encontrada.

$$\begin{aligned}
 \text{Min} \quad & U + C + \sum_{\text{grupos}} c_v \times 100 \\
 \text{sujeito a} \quad & U = 0 \\
 & S \geq S_{\min}, \\
 & 1 \leq d_i \leq m, \\
 & -1 \leq b_{ij} \leq 1 \forall j, \\
 & |b_{ij}| \geq 0,1 \forall j
 \end{aligned} \tag{3.3}$$

O algoritmo proposto, que pode ser classificado como um algoritmo de agrupamento em grade, divide os c -valores em d regiões igualmente espaçadas, sendo d um parâmetro. Inicialmente são identificados sub-grupos, que são regiões contendo no mínimo m/d pontos, sendo m o número de c -valores. Sub-grupos adjacentes são então aglomerados para formar grupos maiores. Os pontos pertencentes às regiões que não formaram sub-grupos compõem o conjunto dos pontos não agrupados (U).

Um exemplo de agrupamento pode ser visto na Figura 3.1, em que $m = 15$ (c -valores) devem ser agrupados utilizando $d = 7$ (regiões) e com critério de formação de sub-grupos igual a $m/d = 2$. Neste exemplo, os c -valores variam no intervalo $[0,7]$ para facilitar a visualização. Utilizando este critério, quatro sub-grupos são formados: $G_1 = 1,2,3$, $G_2 = 5,6,7,8$, $G_3 = 10,11$ e $G_4 = 12,13,14$. Os sub-grupos G_3 e G_4 são aglomerados por serem adjacentes, enquanto os pontos 4 e 9 permanecem não agrupados.

Um pseudo-código para o procedimento de agrupamento é mostrado no Algoritmo 4. O primeiro passo é a inicialização das variáveis U , C , c_v e $flag$. O laço mais externo do algoritmo, entre as linhas 3 e 22, percorre cada uma das d divisões. Na linha 4 obtém-se os pontos que pertencem à divisão corrente. Caso o número de pontos ($|N|$) seja inferior ao critério para formação de sub-grupos, adiciona-se $|N|$ a U , que representa a quantidade de pontos não agrupados. Caso contrário, foi encontrado um sub-grupo. Se $flag = 1$, significa que a última divisão verificada não era um sub-grupo. Neste caso, incrementa-

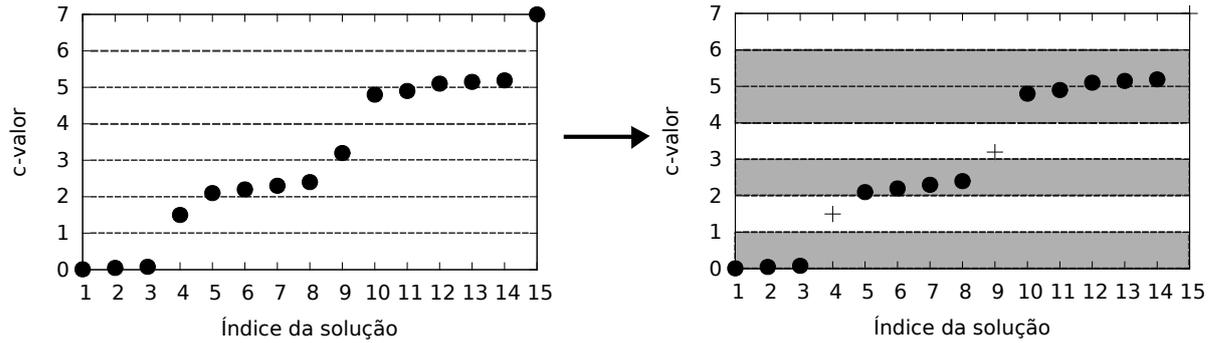


Figura 3.1: Exemplo de agrupamento de 15 c -valores, com $d = 7$ e critério de formação de sub-grupos igual a 2. Adaptado de Bandaru et al. (2015)

se C , que representa o número de grupos. Os pontos que formam o grupo atual são guardados em $Grupo[C]$. Quando $flag = 0$, a última divisão visitada era um sub-grupo e, assim, o sub-grupo atual ($Grupo[C]$) deve ser mesclado àquele (N) e, neste caso, não incrementa-se o contador C . Finalmente, calcula-se o somatório do coeficiente de variação de cada grupo.

Diversas meta-heurísticas poderiam ser utilizadas para resolver o problema de otimização. Em Bandaru e Deb (2010), o problema foi resolvido adotando-se um AG com recombinação binária e mutação bit-a-bit para as variáveis inteiras (d). Para as variáveis reais foram adotadas a recombinação binária simulada (SBX) e mutação polinomial. As restrições foram tratadas utilizando o Critério de Factibilidade sem parâmetros proposto por Deb (2000).

Um dos desafios da tarefa é a obtenção de diferentes princípios de projeto¹, o que foi solucionado pelos autores através do uso de nichos, favorecendo a existência de espécies distintas na mesma população e, assim, soluções com características diferentes podem ser obtidas em uma única execução.

O Algoritmo 5 mostra um pseudo-código para o processo automatizado de *Innovization* (Bandaru e Deb 2011b). O algoritmo começa inicializando os parâmetros do AG e criando sua população inicial. O laço entre as linhas 5 e 19 é repetido por um número de gerações passado como parâmetro. Para cada indivíduo i da população, calcula-se sua aptidão (linhas 7 a 12): o princípio correspondente é avaliado em cada solução, gerando um conjunto de c -valores, que são então ordenados e agrupados, derivando-se sua aptidão. Apesar de aqui se adotar o termo (de inspiração biológica) “aptidão” para representar a qualidade das soluções candidatas, considera-se que valores *menores* são preferíveis, uma

¹Sem uma técnica para garantir diversidade, os AGs podem convergir para uma solução única.

Algoritmo 4: Pseudo-código do procedimento de agrupamento (adaptado de Bandaru e Deb (2010)).

Entrada: *pontos*: vetor ordenado de *c*-valores; *m*: tamanho da base de dados;
d: número de divisões
Saída: *U*: número de pontos sem grupo; *C*: número de grupos;
Soma_c_v: soma dos coeficientes de variação de cada grupo

```

1 início
2    $U \leftarrow 0$ ;  $C \leftarrow 0$ ;  $Soma\_c_v \leftarrow 0$ ;  $flag \leftarrow 1$ ;
3   para  $i \leftarrow 1$  ate  $d$  faça
4      $N \leftarrow$  pontos na divisão  $i$ ;
5     se  $|N| \leq \lfloor m/d \rfloor$  então
6        $U \leftarrow U + |N|$ ;
7        $flag \leftarrow 1$ ;
8     fim se
9     senão
10      se  $flag = 1$  então
11         $C \leftarrow C + 1$ ;
12         $Grupo[C] \leftarrow N$ ;
13      fim se
14      senão
15         $Grupo[C] \leftarrow Grupo[C] \cup N$ ;
16      fim se
17       $flag \leftarrow 0$ ;
18    fim se
19  fim para
20  para  $i \leftarrow 1$  ate  $C$  faça
21     $Soma\_c_v \leftarrow Soma\_c_v + DesvioPadrão(Grupo[i])/Media(Grupo[i])$ ;
22  fim para
23 fim

```

vez que se trata de um problema de minimização. Após a atribuição da aptidão, as etapas de seleção, recombinação, mutação e substituição do AG são executadas.

3.1.2 *Innovization baseado em Programação Genética*

3.1.2.1 *Introdução*

Ao invés de fixar a estrutura dos princípios de projeto às regras de potência, como na Equação 3.1, pode-se pensar em utilizar uma técnica capaz de inferir não somente as constantes numéricas, mas a própria estrutura do modelo. Uma técnica adequada a esta tarefa é a Programação Genética, uma meta-heurística robusta capaz de evoluir estruturas complexas (Poli, Langdon e McPhee 2008). Na próxima seção a técnica de Programação Genética será apresentada e, logo a seguir, o processo de *Innovization* que a utiliza como

Algoritmo 5: Pseudo-código do processo automatizado de *Innovization*.

Entrada: *Dados*: m soluções com variáveis, objetivos, restrições de uma Frente de Pareto;

(*Geracoes*, *Tamanho_populacao*, *Taxa_mutacao*, *Taxa_recombinacao*): parâmetros do AG

Saída: Relações na forma $\prod_{j=1}^N \phi_j(\mathbf{x})^{b_{ij}} = c$

```

1 início
2   Inicializa parâmetros do AG;
3   Inicializa população;
4   geracao ← 1;
5   enquanto geracao ≤ Geracoes faça
6     para  $i \leftarrow 1$  até Tamanho_populacao faça
7       para  $l \leftarrow 1$  até  $m$  faça
8          $c[l] \leftarrow \prod_{j=1}^N \phi_j(\text{Dados}[l])^{b_j^{(i)}}$ ;
9       fim para
10      Ordena( $c$ );
11      ( $U, C, \text{Soma}_c$ ) ← Agrupa( $c, m, d^{(i)}$ ); /* Algoritmo 4 */
12       $\text{aptidao}^i \leftarrow U + C + \text{Soma}_c \times 100$ ;
13    fim para
14    SeleçãoAG();
15    RecombinaçãoAG();
16    MutaçãoAG();
17    SubstituiçãoAG();
18    geracao ← geracao + 1;
19  fim enquanto
20 fim

```

mecanismo de busca.

3.1.2.2 Programação Genética

A Programação Genética (PG), largamente difundida pelo trabalho desenvolvido por Koza (1992), é um AE capaz de otimizar estruturas de dados complexas de forma automatizada, sem que o usuário conheça ou especifique a estrutura da solução desejada (Poli, Langdon e McPhee 2008). Pode ser considerada uma extensão dos AGs (Glover e Kochenberger 2003), na qual a população é formada por programas de computador. Na literatura pode-se encontrar aplicações da PG a diversas áreas, como criação de modelos de regressão e classificadores, otimização de topologias em circuitos eletrônicos, controladores, antenas de transmissão, entre outros (Glover e Kochenberger 2003).

De acordo com Poli, Langdon e McPhee (2008), a PG é uma técnica robusta que pode ser aplicada a vários tipos de problemas, desde que (i) suas soluções possam ser representadas por um programa de computador; e (ii) seja possível comparar a qualidade de duas soluções.

Na PG as soluções são geralmente representadas por árvores, nas quais as folhas (nós sem sucessores) contém os terminais, que são as constantes, variáveis e funções sem parâmetros. Os outros nós da árvore contém as funções, como operações lógicas e aritméticas, condicionais, ou até mesmo sub-rotinas indicadas pelo usuário. A Figura 3.2 ilustra a representação em árvore para um programa que calcula o valor de $X * X + Y/2$. Neste caso, são utilizadas as funções soma (+), multiplicação (*) e divisão (/), enquanto as variáveis X e Y e a constante 2 são os nós terminais.

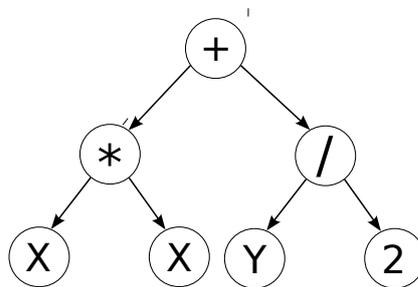


Figura 3.2: Árvore para um programa que calcula o valor de $X \times X + \frac{Y}{2}$.

Os conceitos utilizados na implementação de uma PG são os mesmos encontrados nos AGs, diferindo na manipulação das estruturas de dados, que são notadamente mais complexas. O mesmo pseudo-código de um AG clássico pode ser utilizado para representar

uma PG (Algoritmo 6).

Algoritmo 6: Pseudo-código de uma PG.

```

1 início
2    $pop_{corrente} \leftarrow \text{InicializaPopulação}();$ 
3   Avaliação( $pop_{corrente}$ );
4   enquanto Critério de parada não for atingido faça
5      $indivíduos_{selecionados} \leftarrow \text{Seleção}(pop_{corrente});$ 
6      $pop_{filhos} \leftarrow \text{Recombinação}(indivíduos_{selecionados});$ 
7     Mutação( $pop_{filhos}$ );
8     Avaliação( $pop_{filhos}$ );
9      $pop_{corrente} \leftarrow \text{Substituição}(pop_{corrente}, pop_{filhos});$ 
10  fim enquanto
11 fim

```

Usualmente a população inicial é criada de forma aleatória, como ocorre em outras meta-heurísticas. Os métodos mais simples para inicialização são o *full* e *grow*. A abordagem comumente adotada na literatura consiste numa combinação dos dois métodos, denominada *Ramped half-and-half*. Todos esses métodos de inicialização da população constroem as árvores limitadas em uma altura máxima definida pelo usuário. A altura de uma árvore é definida como a maior distância em nós que deve ser percorrida a partir do nó inicial (raiz) até uma de suas folhas.

No método *full*, somente nós do conjunto de funções são escolhidos até atingir-se a altura máxima definida -1 , a partir da qual são escolhidos os nós terminais. Apesar de possibilitar a criação de árvores completas com a altura máxima especificada, tal método não favorece a variabilidade das formas das árvores. O método *grow*, por sua vez, seleciona aleatoriamente funções e terminais até atingir a altura máxima -1 . Neste ponto somente terminais são escolhidos, como ocorre no *full*. Koza (1992) propôs uma combinação dos dois métodos, *ramped half-and-half*, em que metade da população é inicializada utilizando *full* e o restante com *grow*. Além disso, diferentes alturas são utilizadas para favorecer a diversidade da população.

Várias estratégias podem ser utilizadas para selecionar os indivíduos que irão gerar filhos e formar a nova população, sendo aqui adotado o torneio.

Após a seleção dos pais, utiliza-se o operador de recombinação, que é aplicado de acordo com uma probabilidade, definida por um parâmetro do algoritmo. A recombinação de sub-árvores foi adotada aqui, como sugerido por Bandaru e Deb (2013a). Inicialmente duas sub-árvores são selecionadas aleatoriamente, uma proveniente de cada pai. Os novos

indivíduos são então gerados pela permutação destas duas sub-árvores. Esta abordagem é ilustrada na Figura 3.3. São realizadas recombinações até formar-se uma nova população completa.

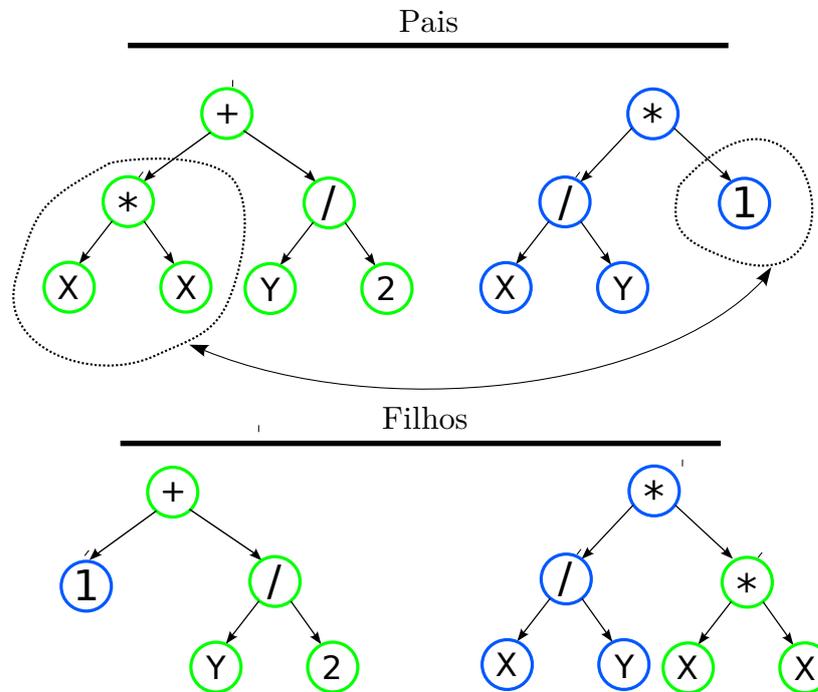


Figura 3.3: Exemplo de recombinação na PG com dois pais gerando dois filhos.

Após a recombinação, cada novo indivíduo gerado pode passar por um processo de mutação, de acordo com uma taxa definida também por um parâmetro a ser indicado pelo usuário. Neste trabalho utiliza-se a mutação de ponto, assim como proposto por Bandaru e Deb (2013a). Este operador de mutação, que é equivalente à mutação bit-a-bit dos AGs, em que somente uma posição do genótipo binário é invertida. Na PG, a mutação de ponto consiste em percorrer todos os nós da árvore e, caso um número real randômico seja inferior à taxa de mutação, troca-se o valor do nó. Se for um nó de função, troca-se a função por outra de mesma aridade, aleatoriamente selecionada; caso contrário, troca-se a variável, constante ou função sem parâmetros por outra selecionada aleatoriamente. A Figura 3.4 ilustra um exemplo de mutação de ponto.

3.1.2.3 Processo de Innovization baseado em Programação Genética

Um passo rumo à obtenção de princípios de projeto mais gerais foi a adoção da Programação Genética (PG) como mecanismo de busca para o *Innovization*, como proposta por Bandaru e Deb (2013a). Como comentado na seção anterior, pode-se dizer

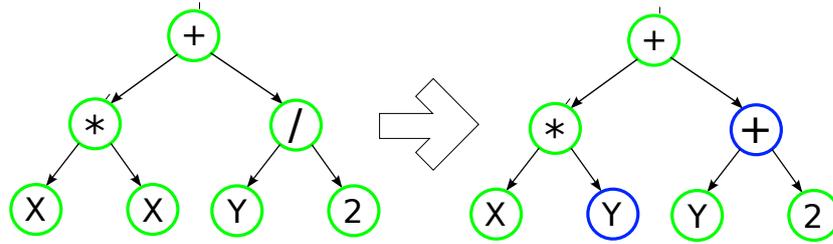


Figura 3.4: Exemplo de mutação de ponto na PG, em que dois nós (um de função e outro terminal) são modificados.

que as principais diferenças entre um AG e uma técnica de PG são (i) a representação das soluções candidatas e (ii) os operadores que geram os novos indivíduos. Assim, a descoberta de princípios de projeto utilizando PG segue as mesmas ideias do Processo Automatizado de *Innovization* com AGs, calculando-se a aptidão dos indivíduos através de um algoritmo de agrupamento.

A técnica apresentada por Bandaru e Deb (2013a) consiste em utilizar uma PG tradicional para evoluir uma população de modelos que representem os princípios de projeto, além de um AG para encontrar os valores ótimos para d e para os parâmetros do algoritmo de agrupamento. Pode-se considerar então que cada indivíduo i é aqui composto por uma árvore (que representa o modelo candidato Psi_i), associado ao número inteiro d_i . Neste trabalho foram adotadas a recombinação de sub-árvore e mutação de ponto, enquanto no AG foram utilizadas a recombinação binária simulada e a mutação polinomial, como sugerido por Bandaru e Deb (2013a).

Assim como nos AGs, no *Innovization* baseado em PG a recombinação explora a noção de nichos. Na etapa de seleção, apenas modelos que utilizem as mesmas variáveis podem participar do torneio e, assim, somente modelos que compartilham características semelhantes são recombinaados. Tal estratégia é implementada com o auxílio de um vetor binário de N elementos, sendo N o número de variáveis. Dado um modelo candidato, cada posição j deste vetor será preenchida por valores iguais a 0 ou 1 indicando, respectivamente, se a j -ésima variável é utilizada ou não como um terminal. A Figura 3.5 ilustra duas árvores que utilizam as mesmas variáveis (X_1 e X_3) e, portanto, pertencem ao mesmo nicho e podem ser recombinaadas.

Neste trabalho a mesma estratégia de nichos proposta na referência foi adotada. Entretanto, outras abordagens podem ser exploradas. Inicialmente, pode-se utilizar mais informação na definição dos nichos, como as funções utilizadas. Além disso, a

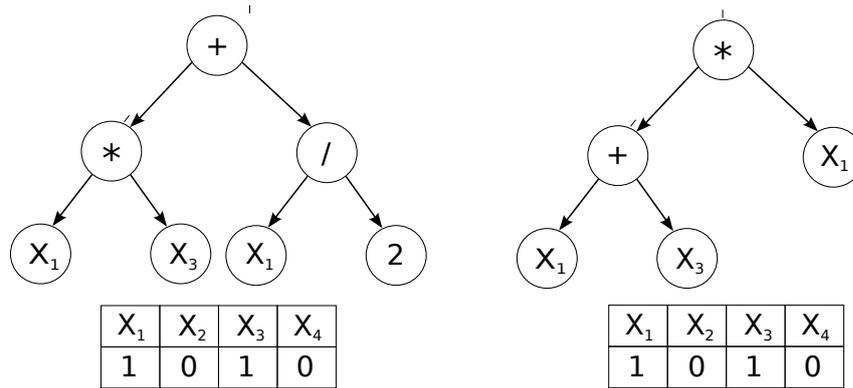


Figura 3.5: Exemplo de duas árvores que pertencem ao mesmo nicho.

determinação do pertencimento de uma solução a um nicho pode ser realizada empregando informações semânticas, ao invés de analisar somente a estrutura dos modelos (fenótipo). Na PG semântica (Krawiec 2016) a distância entre modelos é calculada utilizando seus comportamentos, isto é, as saídas obtidas através de suas execuções.

Além da mudança de representação, uma modificação necessária para utilizar a PG como mecanismo de busca remete à consistência entre as unidades das variáveis do conjunto de dados. Além de representarem invariâncias nas Frentes de Pareto, as inovações devem usar corretamente as unidades das variáveis de decisão, objetivos e restrições. Para tratar isso, Bandaru e Deb (2013a) propõem o uso de uma PG em que a consistência de unidades é verificada durante o processo de busca, de forma a desincentivar a geração de modelos candidatos com combinações incorretas entre as unidades envolvidas. A abordagem escolhida pelos autores foi a penalização a partir de restrições, tornando inviável a solução candidata que apresentar inconsistência no emprego das unidades. As unidades de cada variável, objetivo e restrição são conhecidas *a priori* e são entradas do algoritmo.

Na Tabela 3.1 são exibidas as operações permitidas entre dois nós terminais T_1 e T_2 . Toda operação inválida retorna como expoente o valor Z , que corresponde a um valor maior que expoente máximo permitido (E_{max}). No caso de soma e subtração, somente quantidades comensuráveis podem ser operadas, ou seja, todas as unidades devem apresentar os mesmos expoentes. Nas operações de multiplicação e divisão, deve-se verificar se o expoente final irá ultrapassar E_{max} . Finalmente, na operação de potência, além de testar se o expoente final é maior que E_{max} , o terminal T_2 deve ser uma constante, isto é, o expoente de cada unidade deve ser igual a 0.

Tabela 3.1: Cálculo do expoente E em operações entre dois terminais T_1 e T_2 .

Operação	Operação transformada
$T_1 + T_2$	$E = e_1$, se $e_1 = e_2$ $E = Z$, caso contrário
$T_1 - T_2$	$E = e_1$, se $e_1 = e_2$ $E = Z$, caso contrário
$T_1 \times T_2$	$E = Z$, se $\max\{ e_1 , e_2 \} > E_{max}$ $E = e_1 + e_2$, caso contrário
T_1/T_2	$E = Z$, se $\max\{ e_1 , e_2 \} > E_{max}$ $E = e_1 - e_2$, caso contrário
$T_1^{T_2}$	$E = e_1 \times T_2$, se $ e_1 \leq E_{max}$ e $e_2 = 0$ $E = Z$, caso contrário

Assim o problema de otimização pode ser definido como na Equação 3.4, na qual TS é o número de nós da árvore e S é a significância do princípio. Nessa formulação do problema de otimização, a função objetivo original do *Innovization* é multiplicada por TS para favorecer a inferência de modelos menores e, portanto, mais legíveis (Bandaru e Deb 2013a). Para calcular a significância, definida pela Equação 3.5 (Bandaru e Deb 2013a), executa-se o algoritmo de agrupamento novamente, adotando-se o critério para formação de sub-grupos igual a $\lfloor m/d \rfloor + \epsilon$, sendo ϵ um valor inteiro. O valor obtido (U') é então utilizado como na Equação 3.5.

$$\begin{aligned} \text{Min} \quad & TS(U + C + \sum_{\text{grupos}} c_v \times 100) \\ \text{sujeito a} \quad & U \leq 0 \\ & 1 \leq d \leq m, \end{aligned} \tag{3.4}$$

$$S \geq S_{min},$$

$$|E| \leq E_{max} \forall \text{unidades}$$

$$S = \frac{m - U'}{m} \times 100 \tag{3.5}$$

4 MÉTODOS PROPOSTOS

Neste trabalho é proposta uma solução alternativa para tratamento da consistência entre unidades, em que operações protegidas são criadas para ignorar os termos inválidos das expressões. Desta forma, os modelos inválidos não são eliminados via penalização, mas corrigidos. Propõe-se também uma estratégia para evitar a obtenção de soluções triviais, quando uma PG é utilizada. Finalmente, propõe-se a utilização de um arquivo externo para manter as soluções de interesse encontradas durante o processo de busca.

4.1 ALTERNATIVA PARA TRATAMENTO DE UNIDADES

Ao invés de penalizar princípios de projeto que contenham operações inválidas em relação a unidades incomensuráveis, aqui propõe-se ignorar as operações inconsistentes dos modelos utilizando operações protegidas.

Durante a avaliação de uma árvore, seus nós são percorridos recursivamente, realizando as operações definidas nos nós e retornando o valor resultante. Aqui, além do valor correspondente à execução de cada nó, mantém-se também uma estrutura de dados que contém os expoentes de cada unidade. Para executar uma função, opera-se sobre os valores numéricos e atualiza-se os expoentes das unidades, de acordo com a operação. É importante ressaltar que as unidades das variáveis são conhecidas e as constantes são definidas como adimensionais.

Antes de executar um nó de função, o algoritmo verifica se há inconsistência entre as unidades de seus operandos, de acordo com cada operador. Caso não haja inconsistência, a função é executada normalmente e retorna-se o valor resultante e também os expoentes das unidades, que podem ser alterados de acordo com o operador. Caso a operação não possa ser realizada, retorna-se uma *flag* indicando ao nó predecessor que a operação é inválida.

A partir dessa informação, cabe ao nó predecessor decidir como deve executar sua função, dado que (ao menos) um de seus operandos é inválido. Tal decisão é dependente da função a ser executada e será delineada nos próximos parágrafos.

Nos experimentos realizados aqui o conjunto de funções $F = \{+, -, *, /, \wedge\}$ foi adotado. As operações protegidas para cada função são definidas a seguir, onde A e B são dois nós da árvore utilizados como parâmetros, $Expoente_A^i$ representa o expoente da i -ésima unidade da quantidade representada pelo nó A e E_{max} é o maior expoente permitido. As operações protegidas de soma e subtração, multiplicação, divisão e potência são definidas, respectivamente, pelas Equações 4.1, 4.2, 4.3 e 4.4.

$$A \pm B = \begin{cases} A \pm B, & \text{se } A \text{ e } B \text{ são válidos e } Expoente_A^i = Expoente_B^i \forall i \\ A, & \text{se } B \text{ é inválido e } A \text{ é válido,} \\ B, & \text{se } A \text{ é inválido e } B \text{ é válido,} \\ A, & \text{se } B \text{ é adimensional e } A \text{ é válido,} \\ B, & \text{se } A \text{ é adimensional e } B \text{ é válido,} \\ \text{operação inválida,} & \text{caso contrário} \end{cases} \quad (4.1)$$

$$A * B = \begin{cases} A * B, & \text{Se } A \text{ e } B \text{ são válidos} \\ A, & \text{se } B \text{ é inválido e } A \text{ é válido,} \\ B, & \text{se } A \text{ é inválido e } B \text{ é válido,} \\ \text{operação inválida,} & \text{caso contrário} \end{cases} \quad (4.2)$$

$$A/B = \begin{cases} A/B, & \text{Se } A \text{ e } B \text{ são válidos} \\ A, & \text{se } B \text{ é inválido e } A \text{ é válido,} \\ 1/B, & \text{se } A \text{ é inválido e } B \text{ é válido,} \\ \text{operação inválida,} & \text{caso contrário} \end{cases} \quad (4.3)$$

$$A^B = \begin{cases} \text{operação inválida,} & \text{Se } A \text{ é inválido} \\ A, & \text{se } B \text{ é inválido e } A \text{ é válido,} \\ A, & \text{se } Expoente_B^i \neq 0 \text{ para algum } i, \\ A, & \text{se } B * Expoente_A^i > E_{max} \text{ para algum } i, \\ A^B, & \text{caso contrário} \end{cases} \quad (4.4)$$

A Figura 4.1 ilustra o processo de avaliação de uma solução candidata, onde abaixo de cada nó já visitado são mostrados os expoentes das unidades. Neste exemplo são utilizadas 2 variáveis, X_1 e X_2 e 3 unidades (kg, m e s). A variável X_1 possui unidade m^3 (volume), enquanto X_2 apresenta unidade m^2 (área). A Figura é dividida em 3 estágios:

a árvore inicial (1); a árvore após a execução das operações de soma e subtração (2) o nó final contendo o modelo representado pela árvore (3). A primeira operação realizada é a soma (sub-árvore à esquerda), na qual os parâmetros são os nós contendo as variáveis X_1 e X_2 . Os expoentes das unidades destas duas variáveis são distintos, resultando, assim, numa operação inválida (como definido em 4.1), representada por Inv no estágio (2). A sub-árvore da direita, que representa a divisão entre X_1 e X_2 é executada normalmente, gerando-se o valor numérico X_1/X_2 e subtraindo-se os expoentes de cada unidade, como pode ser observado no estágio (2). Neste ponto há uma multiplicação entre dois operandos, em que o parâmetro é oriundo de uma operação inválida. Assim, ignora-se o termo inválido retornando a sub-árvore à direita, como se vê no estágio (3).

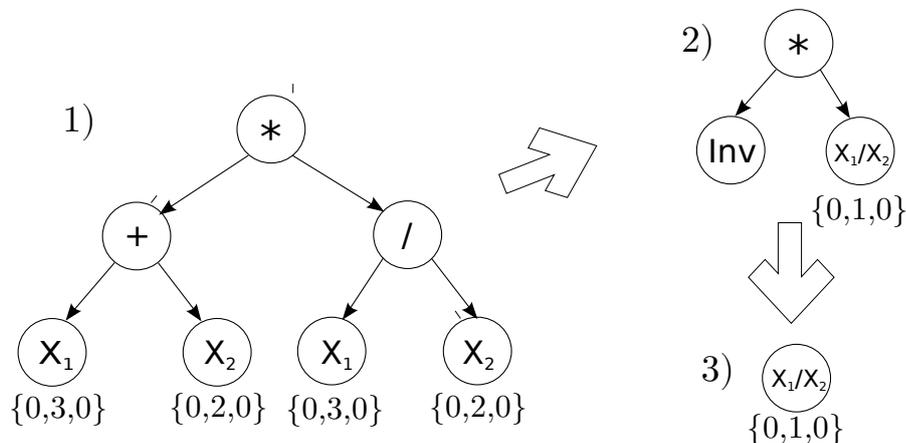


Figura 4.1: Exemplo de avaliação de uma árvore utilizando as operações protegidas definidas neste trabalho. São utilizadas duas variáveis (X_1 e X_2) e 3 unidades.

4.2 ESTRATÉGIA PARA EVITAR A OBTENÇÃO DE SOLUÇÕES TRIVIAIS

Ao utilizar a técnica de *Innovization* para descobrir princípios de projeto, deseja-se minimizar (i) o número de grupos, (ii) o número de pontos não agrupados e (iii) o coeficiente de variação de cada grupo obtido a partir de um algoritmo de agrupamento. Assim, a solução ótima corresponde aos princípios em que o agrupamento dos c -valores gere somente um grupo com valores constantes, minimizando assim (i), (ii) e (iii).

Entretanto, o processo de busca pode dar origem a soluções triviais que, apesar de viáveis (e até mesmo ótimas em alguns casos), não revelam conhecimento algum sobre

o problema tratado. No caso do *Innovization* baseado em AGs, tais soluções ocorrem quando todas as potências (termos b_{ij} da Equação 3.1) são iguais a 0, gerando assim um princípio de projeto equivalente a uma constante. Naquele caso, foram incluídas restrições para impedir que as potências assumam valores nulos, evitando assim a obtenção da solução trivial (Bandaru e Deb 2011b).

Quando uma PG é utilizada, a estrutura dos modelos não é fixa como ocorre no caso anterior, o que pode levar à obtenção de soluções triviais de diferentes formas. A Tabela 4.1 apresenta alguns exemplos de soluções triviais, que são princípios de projeto ótimos de acordo com a definição do *Innovization*, mas que representam valores constantes e portanto são irrelevantes. Neste caso, como a estrutura dos princípios de projeto não é conhecida, torna-se difícil definir um espaço de busca que contenha somente soluções não-triviais. Na proposta de *Innovization* baseado em PG (Bandaru e Deb 2013a) os autores não relataram este problema, tampouco indicaram a adoção de uma estratégia para este fim. Assim propõe-se aqui um método para inibir a ocorrência de tais soluções.

Tabela 4.1: Exemplos de soluções triviais para *Innovization* que podem ser obtidas por uma PG.

Princípio de projeto	Valor constante
$x_1 - x_1$	0
x_1/x_1	1
$(x_1/x_1) - 1$	0
x_1^0	1

As soluções triviais podem, todavia, ser facilmente identificadas, considerando que os princípios de projeto podem não ser matematicamente e numericamente precisos (em razão, por exemplo, de não se conhecer a Frente de Pareto ótima), aqui é utilizada a hipótese de que quando todos os c-valores são iguais, trata-se de uma solução trivial. É importante notar que tal estratégia explora uma falha do processo de otimização, uma vez que idealmente se deseja encontrar princípios que possuam valores constantes em regiões da Frente de Pareto. Um método alternativo é a simplificação simbólica, que pode revelar quando um princípio não depende de fato de nenhuma das variáveis de entrada. Entretanto, esta abordagem possui alto custo computacional e poderia classificar incorretamente alguns princípios como não-triviais.

Quando uma solução trivial é encontrada, atribui-se a ela o pior valor de aptidão possível (pena de morte), de forma a inibir sua ocorrência. O Algoritmo 7 apresenta

um pseudo-código para verificar se um vetor ordenado de c -valores corresponde a uma solução trivial. Se a diferença entre quaisquer valores adjacentes for superior a um dado ϵ , a solução não é constante e o algoritmo retorna *Falso*. Caso contrário retorna *Verdadeiro*. Neste trabalho foi adotado $\epsilon = 10^{-4}$.

Algoritmo 7: Pseudo-código do procedimento para verificar se uma solução é uma constante.

Entrada: c : vetor ordenado de c -valores; m : número de soluções;
 ϵ : valor mínimo considerado

```

1 início
2   para  $i \leftarrow 1$  ate  $m - 1$  faça
3     se  $abs(c[i] - c[i + 1]) > \epsilon$  então
4       retorna Falso;
5     fim se
6   fim para
7   retorna Verdadeiro;
8 fim
```

4.3 UTILIZAÇÃO DE ARQUIVO EXTERNO

A técnica de *Innovization* baseada em PG proposta por Bandaru e Deb (2013a) não utiliza elitismo, uma vez que se deseja obter diferentes princípios de projeto e a adoção de tal estratégia pode levar à convergência do algoritmo a apenas algumas poucas soluções. Contudo, em experimentos preliminares aqui realizados, identificou-se que soluções candidatas promissoras, presentes nas gerações iniciais não eram observadas na população final, em razão dos operadores de exploração do espaço de busca. Desta forma, aqui se propõe manter uma lista contendo todos os modelos que apresentem significância igual ou superior ao mínimo indicado ao longo das gerações. Assim, propõe-se aqui a incorporação de um arquivo externo ao algoritmo de busca para guardar as soluções de interesse.

Não é difícil perceber que um modelo pode ser representado por árvores distintas. Para que isto ocorra, basta, por exemplo, que sejam trocadas as sub-árvores direita e esquerda em algum nó da árvore com operador de soma ou multiplicação. Além disso, modelos iguais a menos de uma soma ou multiplicação por constante numérica representam o mesmo princípio de projeto, uma vez que sua estrutura genérica é da forma $\Psi_i(\mathbf{x}, \mathbf{f}(\mathbf{x}), \mathbf{g}(\mathbf{x})) = c_i$, onde c é uma constante. Na Figura 4.2 são ilustradas 3

representações em árvore para o mesmo modelo, $X_1 \times X_3 + X_1 = c$. Aqui denominamos por **família** todas as representações possíveis de um mesmo modelo. É utilizada então uma lista de famílias de modelos, sendo que a cada família são adicionados as diferentes representações encontradas pelo algoritmo de busca.

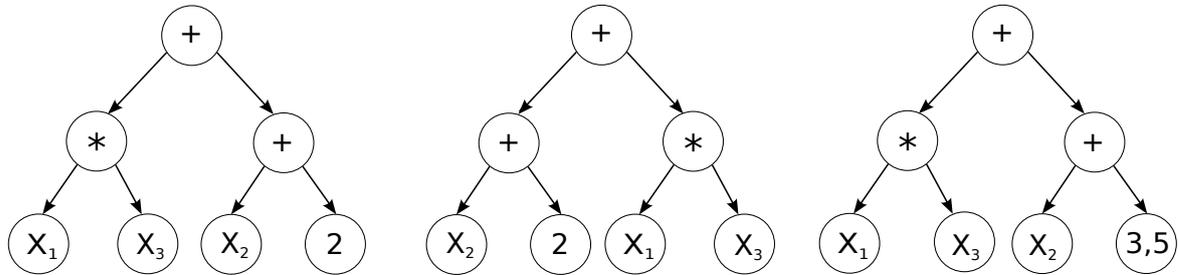


Figura 4.2: Exemplo de árvores que representam o mesmo princípio de projeto, $X_1 \times X_3 + X_2 = c$.

Ao final de cada geração, todos os indivíduos viáveis obtidos são inseridos na lista. Entretanto, para inserir um novo indivíduo, deve-se antes verificar se o mesmo já foi incluído. Pode-se verificar se um modelo pertence à uma família utilizando a correlação entre os c-valores gerados por cada modelo. Quando há correlação positiva (1) ou negativa (-1) assume-se que o modelo pertence àquela família. Tal procedimento é mostrado no Algoritmo 8. Para cada família f , verifica-se a correlação entre o novo modelo (M) e um representante de f (linhas 4 a 6). É importante notar que qualquer modelo de f pode ser utilizado para comparação, uma vez que todos são correlacionados entre si. Caso exista correlação, M é adicionado à família f . Esta inserção é realizada de forma ordenada, sendo o primeiro elemento da lista aquele com o melhor valor de aptidão. Caso os c-valores do novo modelo não tenham correlação com os de nenhuma família presente na lista, uma nova família é então criada (linhas 12 a 14).

A estratégia adotada aqui consiste em utilizar um arquivo de capacidade ilimitada, em que os novos modelos viáveis são sempre adicionados às suas respectivas famílias. Contudo, outras possibilidades podem ser exploradas futuramente, como a limitação do tamanho do arquivo externo, para reduzir a quantidade de dados a serem armazenados e a utilização de um critério adicional para inclusão de novas soluções. No que tange a este último, a técnica *Novelty Search* (NS) (Lehman e Stanley 2011) pode ser utilizada. NS é uma abordagem proposta para PG em que as soluções candidatas são avaliadas utilizando uma medida de inovação, ao invés da função objetivo, que representa o

Algoritmo 8: Procedimento de inserção de um modelo no arquivo externo de soluções.

Input: M : Novo modelo; $c_valores$: c -valores obtidos ao executar M ;
 $Lista$: Lista de famílias de modelos, $Dados$: Base de dados

```

1 início
2    $encontrou \leftarrow Falso$ ;
3   para cada família  $f \in Lista$  faça
4      $representante\_f \leftarrow f[0]$ ;
5      $c\_valores\_aux \leftarrow ExecutaModelo(representante\_f, Dados)$ ;
6      $correl \leftarrow CalculaCorrelação(c\_valores, c\_valores\_aux)$ ;
7     se  $abs(correl) > 1 - 10^5$  então
8        $encontrou \leftarrow Verdadeiro$ ;
9        $AdicionaModeloNaFamília(f, M)$ ;
10    fim se
11  fim para cada
12  se  $encontrou = Falso$  então
13     $CriaNovaFamília(Lista, M)$ ;
14  fim se
15 fim

```

comportamento do indivíduo ao ser exposto a diferentes entradas. Assim, tal critério poderia ser utilizado para inserir somente soluções com comportamento inovador, em relação às já presentes no arquivo. Além disso, poderia haver interações entre a população corrente e arquivo externo, de forma a que este influencie a busca para obtenção de soluções mais diversas e inovadoras.

5 ESTUDOS DE CASOS

5.1 INTRODUÇÃO

Este capítulo descreve os estudos de casos tratados neste trabalho juntamente com as correspondentes formulações. Os resultados dos experimentos computacionais para esses problemas são apresentados e analisados no Capítulo 6.

5.2 ESTUDO DE CASO 1: PROJETO DE UMA TRELIÇA DE 2 BARRAS

O primeiro problema tratado aqui consiste na otimização do projeto de uma treliça de 2 barras (Figura 5.1), com os objetivos de:

1. minimizar o volume total do material das barras (V) e
2. minimizar a tensão máxima atuante nas barras (S).

Originalmente estudado como um problema mono-objetivo, foi tratado como multiobjetivo por Palli et al. (1998). A primeira abordagem evolucionista para obter uma aproximação da Frente de Pareto é atribuída a Deb (2001).

A estrutura da treliça considerada é ilustrada na Figura 5.1, na qual os nós A e B são fixos e é aplicada uma carga de 100kN ao nó C.

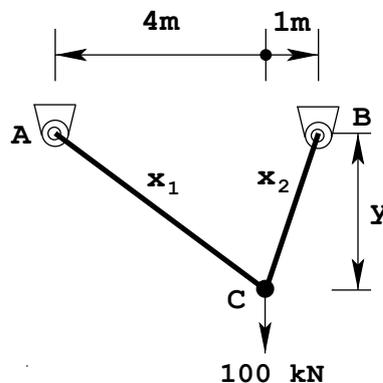


Figura 5.1: Ilustração de uma treliça de 2 barras (Deb e Srinivasan 2006).

O problema possui três variáveis de decisão:

- x_1 , que representa a área da seção transversal da barra que liga os nós A e C;
- x_2 , que representa a área da seção transversal da barra que liga os nós B e C; e
- y , que é a distância vertical entre os nós A e C.

x_1 e x_2 possuem unidade m^2 , enquanto y é medido em metros.

A formulação matemática do problema se escreve:

$$\begin{aligned}
 \text{Min} \quad & f_1(\mathbf{x}, y) \equiv V(\mathbf{x}, y) = x_1 \sqrt{16 + y^2} + x_2 \sqrt{1 + y^2} \\
 \text{Min} \quad & f_2(\mathbf{x}, y) \equiv S(\mathbf{x}, y) = \max\{\sigma_{AC}, \sigma_{BC}\} \\
 \text{sujeito a} \quad & g_1(\mathbf{x}) = \max\{\sigma_{AC}, \sigma_{BC}\} \leq Tensao_{max}, \\
 & 0 \leq x_1, x_2 \leq Area_{max}, \\
 & 1 \leq y \leq 3,
 \end{aligned} \tag{5.1}$$

onde :

$$\begin{aligned}
 \sigma_{AC}(\mathbf{x}) &= \frac{20\sqrt{16 + y^2}}{yx_1}, \\
 \sigma_{BC}(\mathbf{x}) &= \frac{80\sqrt{1 + y^2}}{yx_2}
 \end{aligned}$$

Apesar de sua simplicidade, a escolha por iniciar as investigações por tal problema se deu pelo fato de ser possível derivar analiticamente expressões que representam a Frente de Pareto real. Assim, pode-se verificar se os resultados obtidos pelas técnicas de aprendizado são relevantes e, ainda, permite avaliar se uma técnica é capaz de encontrar todos os resultados esperados.

5.3 ESTUDO DE CASO 2: PROJETO DE UMA VIGA SOLDADA

O problema da viga soldada é um PO estudado na literatura em sua versão mono-objetivo (Ragsdell e Phillips 1976; Deb 1991), enquanto sua versão multiobjetivo pode ser encontrada em Deb e Srinivasan (2006). O problema consiste em encontrar os parâmetros para uma viga que deve ser soldada e suportar uma carga externa F , como ilustrado na Figura 5.2. As mesmas características definidas por Ragsdell e Phillips (1976) são

utilizadas aqui, em que $F = 6000lb$ e a barra a ser soldada é composta de Aço 1010 e possui comprimento $L = (14 + l)in$, sendo l o comprimento da solda.

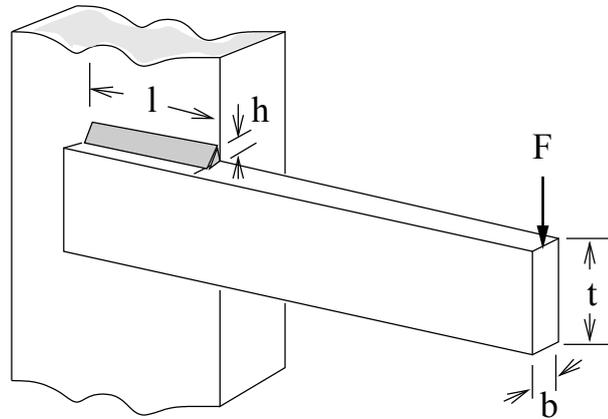


Figura 5.2: Ilustração de uma viga soldada (Deb e Srinivasan 2006).

Os mesmos objetivos utilizados por Deb e Srinivasan (2006) são adotados aqui:

1. minimização do custo de produção da viga (C) e
2. minimização da deflexão vertical da barra (D) ao ser submetida à carga F , que consiste no desvio em relação à posição original.

Pode-se observar que estes dois objetivos são claramente conflitantes. As variáveis de decisão são a base e altura da viga (b e t , respectivamente), além do comprimento e espessura da solda (l e h , respectivamente).

De acordo com Ragsdell e Phillips (1976) os principais custos presentes num processo de solda como este são (a) o custo de preparação das máquinas e materiais; (b) o custo da mão de obra do funcionário que irá soldar a peça e (c) o custo do material utilizado.

O custo de preparação (a) foi desprezado na formulação de Ragsdell e Phillips (1976), assumindo uma linha de produção em que os materiais e máquinas necessários estão prontamente disponíveis.

Para o cálculo do custo da mão de obra (b), considera-se que o funcionário recebe \$10,00 por hora e é capaz de soldar uma polegada cúbica em 6 minutos. Assim o custo (b) pode ser escrito como:

$$custo_b = \left(10 \frac{\$}{h}\right) \times \left(\frac{1}{60} \frac{h}{min}\right) \times \left(6 \frac{min}{in^3}\right) \times V_S = \left(1 \frac{\$}{in^3}\right) \times V_S,$$

onde V_S é o volume de solda utilizado, em in^3 .

O custo do material (c) é dado pela expressão:

$$\text{custo}_c = c_1 V_S + c_2 V_B,$$

onde :

$$\begin{aligned} V_S &= \text{Volume de solda utilizada,} \\ V_B &= \text{Volume da barra a ser soldada,} \end{aligned} \quad (5.2)$$

$$c_1 = \text{Custo da solda} = 0,10471 \frac{\$}{\text{in}^3},$$

$$c_2 = \text{Custo do material da barra} = 0,04811 \frac{\$}{\text{in}^3},$$

Os volumes de solda utilizado e da nova barra são calculados, respectivamente, como:

$$V_S = 2 \left(\frac{1}{2} h^2 l \right) = h^2 l \quad (5.3)$$

e

$$V_B = tb(L + l) \quad (5.4)$$

Substituindo 5.3 e 5.3 na Equação 5.2, pode-se reescrever o custo do material:

$$\text{custo}_c = 0,10471 h^2 l + 0,04811 tb(L + l) \quad (5.5)$$

Portanto, o custo total (C), que é calculado como a soma dos custos de mão de obra e dos materiais empregados, se escreve:

$$\begin{aligned} C &= (\text{custo}_b) + (\text{custo}_c) = \\ &= (h^2 l) + (0,10471 h^2 l + 0,04811 tb(14,0 + l)) = \\ &= 1,10471 h^2 l + 0,04811 tb(14,0 + l) \end{aligned}$$

A deflexão observada na extremidade livre da barra é calculada por:

$$D = \frac{4FL^3}{E \times tb^3} = \frac{2,1952}{t^3 b},$$

onde $E = 30 \times 10^6$ psi é o módulo de Young do material.

Assim, a formulação do problema, adaptada de Deb e Srinivasan (2006), se escreve:

$$\begin{aligned}
\text{Min} \quad & f_1(\mathbf{x}) \equiv C(\mathbf{x}) = 1,10471h^2l + 0,04811tb(14,0 + l) \\
\text{Min} \quad & f_2(\mathbf{x}) \equiv D(\mathbf{x}) = \frac{2,1952}{t^3b} \\
\text{sujeito a} \quad & g_1(\mathbf{x}) = 13600 - \tau(\mathbf{x}) \geq 0, \\
& g_2(\mathbf{x}) = 30000 - \sigma(\mathbf{x}) \geq 0, \\
& g_3(\mathbf{x}) = b - h \geq 0, \\
& g_4(\mathbf{x}) = P_c(\mathbf{x}) - 6000 \geq 0, \\
& 0,125 \leq h, b \leq 5,0, \\
& 0,1 \leq l, t \leq 10,0,
\end{aligned} \tag{5.6}$$

onde:

$$\begin{aligned}
\tau(\mathbf{x}) &= \sqrt{(\tau'(\mathbf{x}))^2 + (\tau''(\mathbf{x}))^2 + \frac{l\tau'(\mathbf{x})\tau''(\mathbf{x})}{\sqrt{0,25(l^2 + (h + t)^2)}}}, \\
\tau'(\mathbf{x}) &= \frac{6000}{\sqrt{2hl}}, \\
\tau''(\mathbf{x}) &= \frac{6000(14 + 0,5l)\sqrt{0,25(l^2 + (h + t)^2)}}{1,414hl(\frac{l^2}{12} + 0,25(h + t)^2)}, \\
\sigma(\mathbf{x}) &= \frac{504000}{bt^2}, \\
P_c(\mathbf{x}) &= 64746,022(1 - 0,02823446t)tb^3
\end{aligned}$$

Além de otimizar simultaneamente os dois objetivos, as soluções devem satisfazer a quatro restrições. As restrições g_1 e g_2 garantem que as tensões de cisalhamento e normal sofridas na região de suporte da viga sejam inferiores aos limites estabelecidos para o material. A tensão de cisalhamento $\tau(\mathbf{x})$ é calculada como apresentado por Shigley, Mischke e Budynas (2004). A restrição g_3 exige que a base da viga não seja menor que a espessura da solda. Finalmente, g_4 requer que a carga de flambagem lateral da viga seja maior que a carga aplicada F . A carga de flambagem é determinada utilizando a aproximação proposta por Timoshenko e Gere (1961) para barras retangulares estreitas.

5.4 ESTUDO DE CASO 3: PROCESSO DE CORTE DE UMA BARRA DE METAL

Este caso envolve a otimização do processo de corte de metal a partir da rotação de um torno mecânico, como descrito em Sardiñas, Santana e Brindis (2006). A Figura 5.3 ilustra o processo, em que remove-se material de uma peça metálica utilizando uma ferramenta de corte abrasiva.

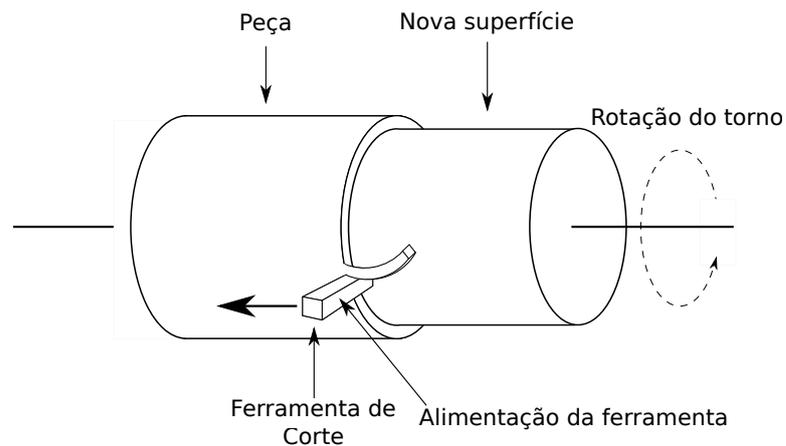


Figura 5.3: Esquema de corte de uma peça metálica em um torno mecânico.

O problema proposto consiste em utilizar uma ferramenta de corte de carbeto de tungstênio de raio $r_n = 0,8mm$ e um torno mecânico com motor de potência máxima $P^{max} = 10kW$ e eficiência $\eta = 75\%$ para remoção de $219912mm^3$ de material da peça.

Os objetivos são:

1. a minimização do tempo total da operação (T_p) e
2. a minimização da redução da vida útil da ferramenta de corte (ξ).

As variáveis de decisão são a velocidade de rotação do torno (v , medida em m/min), a taxa de alimentação (f , medida em mm/rev) e a profundidade do corte (a , medida em mm).

O primeiro objetivo envolve tempo de operação (T_p) para remoção de material e é dado por (Sardiñas, Santana e Brindis 2006):

$$T_p = T_s + \frac{V}{MRR} \left(1 + \frac{T_{tc}}{T} \right) + T_0, \quad (5.7)$$

onde T_s , T_{tc} e T_0 correspondem aos tempos de preparação do equipamento, de troca e de inatividade da ferramenta de corte, respectivamente; V é o volume de material a ser removido; T é a vida útil da ferramenta de corte e MMR é a taxa de remoção de material. De acordo com Sardiñas, Santana e Brindis (2006) a vida útil da ferramenta pode ser calculada como a seguir:

$$T = C_T v^\alpha f^\beta a^\gamma, \quad (5.8)$$

onde os coeficientes C_T, α , β e γ devem ser obtidos experimentalmente. Ainda, a taxa de remoção de material MMR é dada pela expressão:

$$\text{MMR}(\mathbf{x}) = 1000 v f a \quad (5.9)$$

O segundo objetivo corresponde à minimização do percentual da vida útil da ferramenta de corte que é consumido durante o processo e é dado por:

$$\xi(\mathbf{x}) = \frac{V}{\text{MMR}(\mathbf{x})T(\mathbf{x})} \times 100 \quad (5.10)$$

A formulação do problema foi adaptada de Sardiñas, Santana e Brindis (2006) (onde

os coeficientes foram obtidos empiricamente) e se escreve:

$$\begin{aligned}
 \text{Min} \quad & f_1(\mathbf{x}) \equiv T_p(\mathbf{x}) = 0,15 + 219912 \left(\frac{1 + \frac{0,20}{T(\mathbf{x})}}{\text{MRR}(\mathbf{x})} \right) + 0,05 \\
 \text{Min} \quad & f_2(\mathbf{x}, y) \equiv \xi(\mathbf{x}) = \frac{219912}{\text{MRR}(\mathbf{x})T(\mathbf{x})} \times 100 \\
 \text{sujeito a} \quad & P(\mathbf{x}) \leq \eta P^{max}, \\
 & F_c \leq F_c^{max}, \\
 & R(\mathbf{x}) \leq R^{max}, \\
 & 250 \leq v \leq 400 \text{ m/min}, \\
 & 0,15 \leq f \leq 0,55 \text{ mm/rev}, \\
 & 0,5 \leq a \leq 6 \text{ mm},
 \end{aligned} \tag{5.11}$$

onde:

$$\begin{aligned}
 T(\mathbf{x}) &= \frac{5,48 \times 10^9}{v^{3,46} f^{0,696} a^{0,460}}, \\
 \text{MRR}(\mathbf{x}) &= 1000 v f a, \\
 P(\mathbf{x}) &= \frac{v F_c(\mathbf{x})}{60000}, \\
 F_c(\mathbf{x}) &= \frac{6,56 \times 10^3 f^{0,917} a^{1,10}}{v^{0,286}}, \\
 R(\mathbf{x}) &= \frac{125 f^2}{r_n}
 \end{aligned}$$

O problema apresenta três restrições, além dos limites das variáveis. A primeira restrição limita a potência de corte à potência máxima do motor. A segunda, restringe a força de corte à força máxima suportada pelo sistema. Finalmente, a terceira restrição garante que a rigidez da nova superfície seja menor que um valor máximo dado.

5.5 ESTUDO DE CASO 4: PROJETO DE ENGRENAGENS COMPOSTAS

O problema das engrenagens compostas consiste em encontrar o número de dentes de cada uma de 4 engrenagens conectadas, de forma a (Kannan e Kramer 1994; Deb e Srinivasan 2006):

1. minimizar o erro entre a taxa de rotação obtida e a desejada, de 6.931 : 1 (*Erro*) e
2. minimizar o número de dentes da maior engrenagem (*Max*).

A Figura 5.4 ilustra as 4 engrenagens compostas. Como o número de dentes de uma engrenagem é diretamente proporcional ao seu diâmetro, pode-se utilizar os diâmetros das engrenagens (x_1, x_2, x_3, x_4) como variáveis de decisão do problema (Deb e Srinivasan 2006).

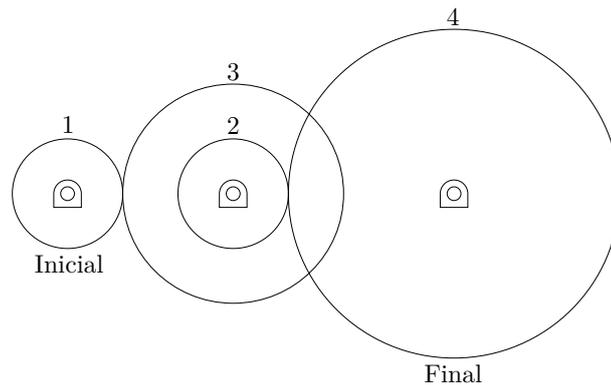


Figura 5.4: Representação do projeto de 4 engrenagens compostas. Adaptado de Deb e Srinivasan (2006).

A formulação do problema, adaptada de Deb e Srinivasan (2006), se escreve:

$$\begin{aligned}
 \text{Min} \quad & f_1(\mathbf{x}) \equiv \text{Erro} = \left| 6.931 - \frac{x_3 x_4}{x_1 x_2} \right| \\
 \text{Min} \quad & f_2(\mathbf{x}) \equiv \text{Max} = \max\{x_1, x_2, x_3, x_4\} \\
 \text{sujeito a} \quad & \frac{f_1(\mathbf{x})}{6,391} \leq 0,5, \\
 & 12 \leq x_1, x_2, x_3, x_4 \leq 60
 \end{aligned} \tag{5.12}$$

Uma restrição é utilizada para garantir que o erro entre as taxas de transmissão seja inferior a 50%.

5.6 ESTUDO DE CASO 5: PROJETO DE UMA TRELIÇA DE 10 BARRAS

Outro problema de otimização estrutural a ser tratado aqui é o projeto de uma treliça de 10 barras e 6 nós, ilustrada na Figura 5.5.

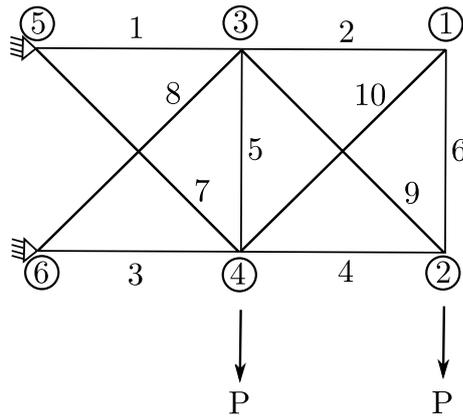


Figura 5.5: Ilustração de uma treliça de 10 barras (Vargas 2015).

Dois objetivos comumente utilizados e adotados neste trabalho são:

1. a minimização do peso da estrutura e
2. a minimização da soma dos deslocamentos dos nós 2 e 4, aos quais são aplicadas cargas externas.

Os nós 5 e 6 são mantidos fixos enquanto são aplicadas verticalmente duas cargas externas de 100 kips sobre os nós 2 e 4. Trabalhos envolvendo a utilização de MOEAs para resolução deste problema podem ser encontrados em Hajela e Lin (1992), Luh e Chueh (2004), Su et al. (2011), Angelo, Bernardino e Barbosa (2015) e Vargas et al. (2016).

As variáveis de decisão são as áreas das seções transversais de cada barra (X_1, X_2, \dots, X_{10}), que podem variar entre $0,1in^2$ e $40in^2$. Outros dados do problema são a densidade do material, $\rho = 0,1lb/in^3$, a tensão normal máxima permitida, ± 25 ksi e o módulo de Young do material $E = 10^4$ ksi.

6 EXPERIMENTOS COMPUTACIONAIS

6.1 INTRODUÇÃO

Experimentos computacionais foram realizados para avaliar se os métodos propostos contribuem para a obtenção de uma maior diversidade de princípios de projeto. Este capítulo possui uma seção para cada estudo de caso aqui tratado, onde são apresentados os experimentos para resolução do problema multiobjetivo, busca por princípios de projeto utilizando *Innovization* a partir das Frentes de Pareto obtidas e análise dos resultados.

Para cada problema multiobjetivo, foram realizadas 30 execuções independentes do algoritmo NSGA-II¹, uma vez que se trata de uma técnica estocástica. Todas as soluções encontradas para um problema são reunidas e uma Frente de Pareto é então gerada pela combinação dessas soluções, contendo todas as soluções não-dominadas obtidas. Em cada caso são mostradas figuras contendo duas Frentes de Pareto: uma obtida na primeira execução independente e aquela formada pela união de todas as execuções. Os parâmetros adotados em cada estudo de caso foram obtidos da literatura e são exibidos em tabelas nas respectivas seções.

Após resolver o problema multiobjetivo, foram utilizadas as modificações propostas para descoberta de conhecimento via *Innovization* sobre os estudos de caso. A técnica foi implementada em linguagem C², a partir da biblioteca smallGP³ disponível em <http://sourceforge.net/p/smallgp/news>., assim como feito por Bandaru e Deb (2013a). Em cada estudo de caso, foram realizadas 50 execuções independentes utilizando como entrada a Frente de Pareto obtida pela união das 30 execuções independentes do NSGA-II. O tratamento de unidades proposto aqui foi utilizado e nos experimentos foi verificada a influência da utilização de um arquivo externo para manutenção das soluções de interesse. Quando a estratégia de arquivo externo não é utilizada, a criação das famílias de modelos ocorre uma vez, ao final do processo evolutivo, sendo apresentados somente

¹Foi utilizada a implementação em linguagem C disponível em <http://www.egr.msu.edu/~kdeb/codes.shtml> (versão 1.1.6).

²O código fonte pode ser acessado em <http://bitbucket.org/ciml-ufjf/ciml-lib>.

representantes de famílias distintas.

Para cada estudo de caso são apresentadas tabelas contendo os princípios de projeto de maior aptidão. Nestas tabelas são exibidas as expressões que representam os princípios, além de sua significância e unidades básicas utilizadas. Os parâmetros utilizados para os estudos de caso 1, 2 e 3 foram obtidos da literatura, sendo os demais escolhidos empiricamente. Os parâmetros adotados são apresentados em cada seção. Nos estudos de caso 1, 2, 3 e 4, os princípios de projeto encontrados aqui são comparados qualitativamente àqueles obtidos na literatura.

Um experimento adicional foi realizado para o estudo de caso 5 (treliça de 10 barras). Um dos objetivos do *Innovization* é a busca de características que identifiquem as características das melhores soluções, de forma a possibilitar a obtenção de boas soluções sem a necessidade de um novo processo de otimização, que muitas vezes é computacionalmente custoso. Neste sentido, alguns princípios obtidos para o estudo de caso 5 foram utilizados para reduzir o número de variáveis do problema e assim, diminuir o custo computacional necessário para resolver o problema multiobjetivo.

6.2 ESTUDO DE CASO 1: PROJETO DE UMA TRELIÇA DE 2 BARRAS

6.2.1 *Problema Multiobjetivo*

O problema foi resolvido utilizando o NSGA-II com os parâmetros exibidos na Tabela 6.1, conforme adotado por Bandaru e Deb (2013a). A Frente de Pareto obtida pela primeira execução independente do algoritmo é mostrada na Figura 6.1, enquanto as soluções não-dominadas encontradas em 30 execuções independentes podem ser observadas na Figura 6.2. Cada execução independente foi capaz de encontrar 500 soluções não-dominadas, como reportado em Bandaru e Deb (2013a), enquanto 3569 soluções não-dominadas foram obtidas através da união de todas estas. Os resultados obtidos são semelhantes aos reportados por Bandaru e Deb (2013a), no que tange a convergência e diversidade.

Tabela 6.1: Parâmetros adotados para o NSGA-II no problema da treliça de 2 barras.

Parâmetro	Valor
População	500
Gerações	500
Recombinação	SBX com $p_c = 0,9$ e $\eta_c = 10$
Mutação	Polinomial com $p_m = 0,05$ e $\eta_m = 50$

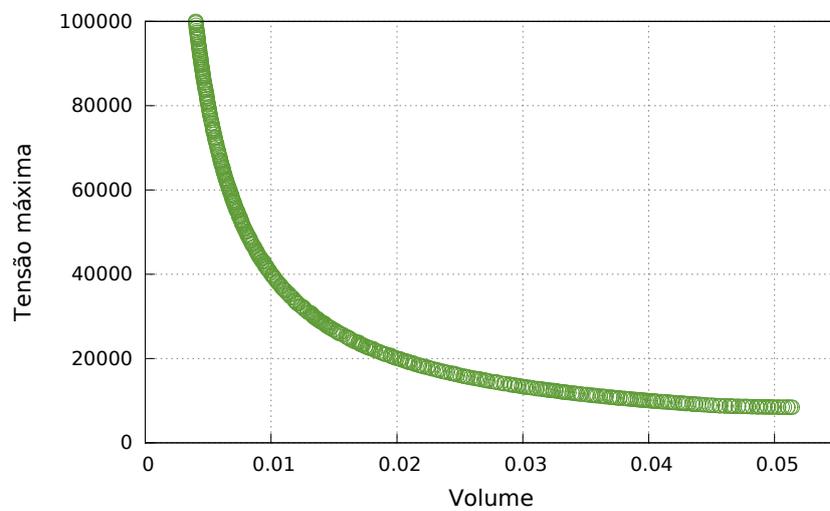


Figura 6.1: Frente de Pareto encontrada em uma execução do NSGA-II para o problema da treliça de duas barras.

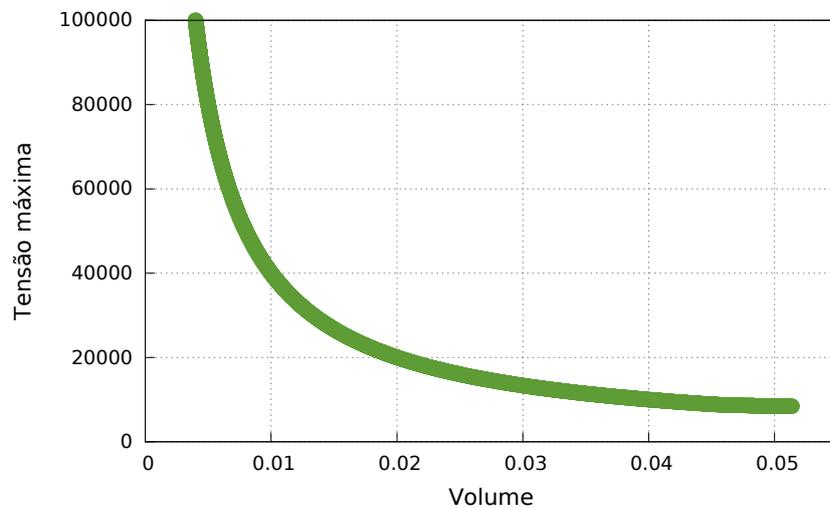


Figura 6.2: Frente de Pareto com as soluções não-dominadas obtidas por 30 execuções do NSGA-II para o problema da treliça de duas barras.

6.2.2 *Descoberta de Conhecimento via Innovization*

Os parâmetros adotados para o processo de *Innovization* foram obtidos em Bandaru e Deb (2013a) e são mostrados na Tabela 6.2. As unidades de cada variável do conjunto de terminais é mostrada na Tabela 6.3.

Tabela 6.2: Parâmetros adotados para o *Innovization* no problema da treliça de 2 barras.

Parâmetro	Valor
População	1000
Gerações	100
Recombinação (PG)	Recombinação de sub-árvores com $p_c = 0,9$
Mutação (PG)	Mutação de ponto com $p_m = 0,2$ por nó
Recombinação (AG)	SBX com $p_c = 0,9$ e $\eta_c = 10$
Mutação (AG)	Polinomial com $p_m = 0,05$ e $\eta_m = 50$
Conjunto de funções	$\{+, -, *, /, \wedge\}$
Conjunto de terminais	$\{V, S, x_1, x_2, y, R\}$
Constantes efêmeras (R)	$\{-10,0, -9,5, -9,0, \dots, 9,0, 9,5, 10,0\}$
Dimensões básicas	$\{\text{massa(kg)}, \text{comprimento(m)}, \text{tempo(s)}\}$
Significância mínima (S_{min})	80%
Constante de agrupamento (ϵ)	3

Tabela 6.3: Unidades das variáveis utilizadas no problema da treliça de 2 barras.

Variável	Dimensões		
	Massa	Comprimento	Tempo
V	0	3	0
S	1	-1	-2
x_1	0	2	0
x_2	0	2	0
y	0	1	0

Considerando a utilização de arquivo externo, os resultados obtidos em 50 execuções independentes são mostrados na Tabela 6.4. A partir deste resultado pode-se notar que

a adoção desta estratégia possibilita a obtenção de um maior número de princípios de projeto. Em relação à aptidão, a utilização do arquivo externo possibilitou encontrar uma solução melhor. Pode-se perceber também que a média das aptidões piora ao utilizar o arquivo externo, como se poderia esperar, uma vez que um número muito maior de soluções é encontrado no início do processo de busca. Diferentemente do que comumente ocorre em experimentos de computação evolutiva, aqui as análises envolvem todas as soluções finais, não apenas a melhor, uma vez que se deseja encontrar múltiplas soluções.

Tabela 6.4: Resultados quantitativos obtidos no problema da treliça de 2 barras, considerando a utilização de arquivo externo.

Arquivo externo	Número de soluções			Aptidão		
	Mínimo	Máximo	Média	Mínimo	Máximo	Média
Não	5	23	10,9±5,09	18.0	6090,56	1450,95±1072,08
Sim	37	166	91,7±45,05	5,00	6202,09	1758,09±1128,54

Os 20 melhores (em relação à aptidão) princípios de projeto obtidos aqui são mostrados na Tabela 6.5. Os princípios DP4, DP6, DP7 e DP10 correspondem aos previstos pela solução analítica (Bandaru e Deb 2013a). Além disso, 9 princípios mostrados por Bandaru e Deb (2013a) e exibidos na Tabela 6.6 foram encontrados. Um princípio interessante não reportado em Bandaru e Deb (2013a) é o princípio DP6, que sugere uma relação constante entre as áreas x_1 e x_2 .

Tabela 6.5: Resultados obtidos para o estudo de caso da estrutura de 2 barras. As unidades básicas são massa (kg), comprimento (m) e comprimento(s).

Notação	Princípio	S	Aptidão	Unidades		
				kg	m	s
DP1	$S/x_2 = c$	90,31%	5,00	1,0	-3,0	-2,0
DP2	$1/(S/V) = c$	93,11%	8,43	-1,0	-2,0	2,0
DP3	$x_2/V = c$	92,04%	11,47	0,0	-1,0	0,0
DP4	$y = c$	91,96%	13,63	0,0	1,0	0,0
DP5	$x_1 + x_1 - x_2 = c$	89,89%	15,09	0,0	2,0	0,0
DP6	$x_2 \times S = c$	88,93%	24,18	1,0	1,0	-2,0
DP7	$S \times x_1 = c$	88,60%	28,43	1,0	1,0	-2,0
DP8	$1/y = c$	89,60%	30,40	0,0	-1,0	0,0
DP9	$(S \times V)/y = c$	87,76%	34,86	1,0	1,0	2,0
DP10	$x_1/x_2 = c$	88,29%	37,02	0,0	0,0	0,0
DP11	$x_1/V = c$	89,30%	49,58	0,0	-1,0	0,0
DP12	$(x_2 \times S)/y = c$	87,25%	70,02	1,0	0,0	-2,0
DP13	$(x_2 \times V)/y = c$	88,32%	72,08	0,0	4,0	0,0
DP14	$(x_1 \times S)/y = c$	87,28%	88,50	1,00	0,00	-2,00
DP15	$y \times y = c$	91,76%	94,91	0,00	2,00	0,00
DP16	$V/x_1 = c$	91,29%	122,56	0,00	1,00	0,00
DP17	$x_2 \times (S \times y) = c$	92,21%	159,28	1,00	2,00	-2,00
DP18	$(y \times x_2)/x_1 = c$	91,54%	161,49	0,00	1,00	0,00
DP19	$V \times y \times S = c$	91,31%	168,69	1,00	3,00	-2,00
DP20	$(x_1 - x_2) \times S = c$	86,52%	180,99	1,00	1,00	-2,00

Tabela 6.6: Resultados obtidos em Bandaru e Deb (2013a) para o estudo de caso da estrutura de 2 barras. As unidades básicas são massa (kg), comprimento (m) e comprimento(s).

Notação	Princípio	S	Unidades básicas		
			kg	m	s
DP1	$y = c$	86,60%	0,0	1,0	0,0
DP2	$S \times V = c$	87,00%	1,0	2,0	-2,0
DP3	$S \times x_1 = c$	85,00%	1,0	1,0	-2,0
DP4	$S \times V \times y = c$	87,00%	1,0	3,0	-2,0
DP5	$(V \times y)/x_2 = c$	86,20%	0,0	2,0	0,0
DP6	$(V \times y)/x_1 = c$	88,20%	0,0	2,0	0,0
DP7	$V/x_1 = c$	86,40%	0,0	1,0	0,0
DP8	$V/(S \times x_1 \times x_2) = c$	87,20%	-1,0	0,0	2,0
DP9	$V^2/(x_1 \times x_2) = c$	87,40%	0,0	2,0	0,0
DP10	$y/(S \times x_1) = c$	88,00%	-1,0	0,0	2,0
DP11	$x_2/x_1 = c$	83,80%	0,0	0,0	0,0
DP12	$(S \times V \times x_2 \times y)/x_1 = c$	88,00%	1,0	3,0	-2,0
DP13	$V/x_2 = c$	86,80%	0,0	1,0	0,0
DP14	$(S \times V^2 \times y)/x_1 = c$	87,20%	1,0	4,0	-2,0
DP15	$(x_2 \times y)/x_1 = c$	86,40%	0,0	1,0	0,0
DP16	$x_2/(S \times x_1^2) = c$	86,40%	-1,0	-1,0	2,0
DP17	$V^2/(x_1 \times x_2 \times y) = c$	91,40%	0,0	1,0	0,0
DP18	$(S \times V^2)/x_2 = c$	87,20%	1,0	3,0	-2,0
DP19	$S \times x_2 \times y = c$	87,00%	1,0	2,0	-2,0
DP20	$(x_2 \times y)/(S \times x_1^2) = c$	86,80%	-1,0	0,0	2,0

6.3 ESTUDO DE CASO 2: PROJETO DE UMA VIGA SOLDADA

6.3.1 Problema Multiobjetivo

Este problema foi resolvido com os mesmos parâmetros utilizados no problema da treliça de 2 barras (Tabela 6.1), como adotado por Bandaru e Deb (2013a). As Figuras 6.3 e 6.4 ilustram, respectivamente, as Frentes de Pareto obtidas através de uma e 30 execuções independentes. Cada uma das 30 execuções encontrou 500 soluções não-dominadas, como reportado por Bandaru e Deb (2013a). A Frente de Pareto total gerada com a união de todas as execuções contém 4063 pontos.

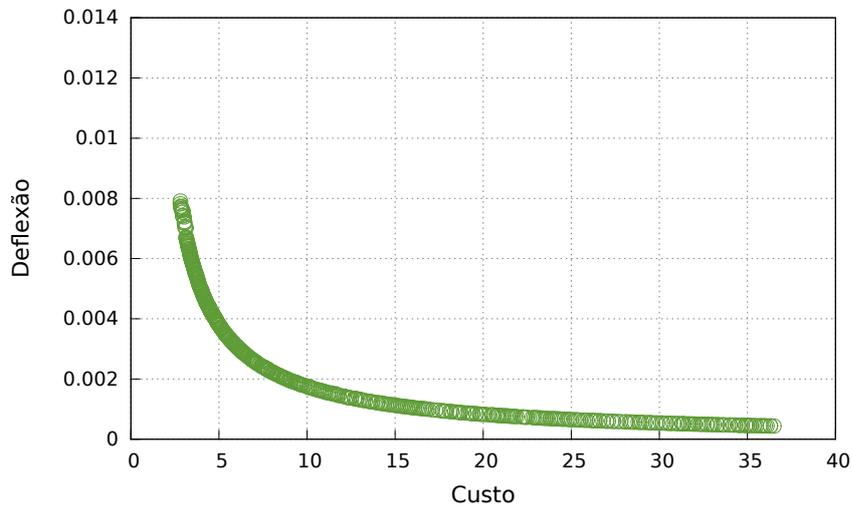


Figura 6.3: Frente de Pareto encontrada em uma execução do NSGA-II para o problema da viga soldada.

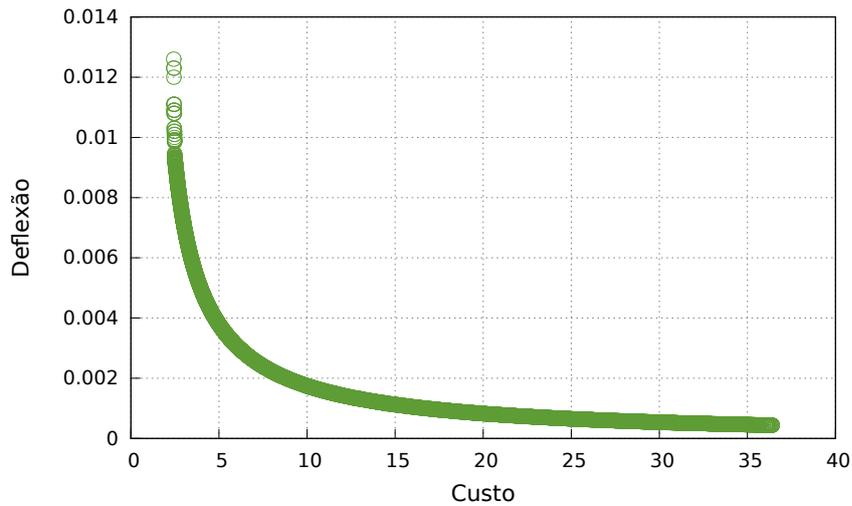


Figura 6.4: Frente de Pareto com as soluções não-dominadas obtidas por 30 execuções do NSGA-II para o problema da viga soldada.

6.3.2 *Descoberta de Conhecimento via Innovization*

Na Tabela 6.7 são mostrados os parâmetros adotados para o estudo de caso da viga soldada, baseados em Bandaru e Deb (2013a). As unidades de cada variável de entrada são mostradas na Tabela 6.8.

Tabela 6.7: Parâmetros adotados para o *Innovization* no problema da viga soldada.

Parâmetro	Valor
População	3000
Gerações	100
Recombinação (PG)	Recombinação de sub-árvores com $p_c = 0,9$
Mutação (PG)	Mutação de ponto com $p_m = 0,2$ por nó
Recombinação (AG)	SBX com $p_c = 0,9$ e $\eta_c = 10$
Mutação (AG)	Polinomial com $p_m = 0,05$ e $\eta_m = 50$
Conjunto de funções	$\{+, -, *, /, \wedge\}$
Conjunto de terminais	$\{C, D, b, t, l, h, \sigma, \tau, P_c, R\}$
Constantes efêmeras (R)	$\{-10,0, -9,5, -9,0, \dots, 9,0, 9,5, 10,0\}$
Dimensões básicas	$\{\text{massa}(\text{lb}), \text{comprimento}(\text{in}), \text{tempo}(\text{s}), \text{custo}\}$
Significância mínima (S_{min})	90%
Constante de agrupamento (ϵ)	3

Tabela 6.8: Unidades das variáveis utilizadas no estudo de caso da viga soldada.

Variável	Unidades			
	Massa	Comprimento	Tempo	Custo
C	0	0	0	1
D	0	1	0	0
b	0	1	0	0
t	0	1	0	0
l	0	1	0	0
h	0	1	0	0
σ	1	-1	-2	0
τ	1	-1	-2	0
P_c	1	0	0	0

Na Tabela 6.9 são mostrados os resultados obtidos em 50 execuções independentes, considerando o uso ou não do arquivo externo. Assim como no estudo de caso anterior, a adoção do arquivo externo possibilitou a obtenção de um número maior de modelos e uma aptidão melhor.

Tabela 6.9: Resultados quantitativos obtidos para o estudo de caso da viga soldada, considerando a utilização de arquivo externo.

Arquivo externo	Número de soluções			Aptidão		
	Mínimo	Máximo	Média	Mínimo	Máximo	Média
Não	9	23	9,38±2,9	3,0	5211,00	444,12±501,39
Sim	132	201	153,38±13,99	1,08	13868,18	1683,57±1343,11

Os princípios de projeto obtidos são mostrados na Tabela 6.10. Os princípios DP1, DP6 e DP9 eram esperados, pois foram encontrados em estudos anteriores de *Innovization* (Deb e Srinivasan 2006; Bandaru e Deb 2011a; Bandaru e Deb 2013a). Em relação aos resultados obtidos em por Bandaru e Deb (2013a), pode destacar que aqui foram obtidos princípios envolvendo a restrição τ , conforme Deb e Srinivasan (2006) haviam sugerido. O princípio DP2 indica que o valor da restrição τ é constante ao longo de grande parte da Frente de Pareto. Ainda, observou-se que seu valor aproxima-se

do máximo permitido para esta restrição, sugerindo que não é possível obter soluções melhores para este problema, sem que a restrição seja modificada (por exemplo, pode-se decidir utilizar outro material). Dos 10 princípios apresentados aqui, 5 foram também obtidos por Bandaru e Deb (2013a): DP1, DP4, DP6, DP9 e DP10 (Tabela 6.10). Dos 8 princípios encontrados nesta referência 2 não foram redescobertos aqui: DP4 e DP8 (Tabela 6.11). Contudo, DP4 e DP8 podem ser obtidos a partir da combinação de outros princípios e, assim, pode-se considerar que todas inovações apresentadas pela referência foram encontradas neste experimento.

Tabela 6.10: Princípios obtidos para o estudo de caso da viga soldada. As unidades básicas são massa (lb), comprimento (in), tempo (s) e custo.

Notação	Princípio	S	Aptidão	Unidades básicas			
				lb	in	s	custo
DP1	$t = c$	99,29%	2,00	0,0	1,0	0,0	0,0
DP2	$\tau = c$	99,21%	2,00	1,0	-1,0	-2,0	0,0
DP3	$P_c/D = c$	96,90%	3,00	1,0	-1,0	0,0	0,0
DP4	$D - t = c$	98,82%	3,65	0,0	1,0	0,0	0,0
DP5	$b \times C = c$	99,95%	9,00	0,0	1,0	0,0	1,0
DP6	$b \times \sigma = c$	98,84%	9,19	1,0	0,0	-2,0	0,0
DP7	$(P_c/D) \times C = c$	90,50%	10,00	1,0	-1,0	0,0	1,0
DP8	$(\sigma \times \tau)/(l - h)$	97,76%	16,29	2,0	-3,0	-4,0	0,0
DP9	$b \times D = c$	98,55%	74,86	0,0	3,0	0,0	0,0
DP10	$\sigma/D = c$	97,39%	141,71	1,0	-2,0	-2,0	0,0

Tabela 6.11: Resultados obtidos por Bandaru e Deb (2013a) para o estudo de caso da viga soldada.

Notação	Princípio	S	Unidades básicas			
			lb	in	s	custo
DP1	$(D + t) = c$	95,20%	0,0	1,0	0,0	0,0
DP2	$t = c$	95,60%	0,0	1,0	0,0	0,0
DP3	$D \times b = c$	95,00%	0,0	2,0	0,0	0,0
DP4	$D \times b \times t = c$	95,60%	0,0	3,0	0,0	0,0
DP5	$\sigma \times b = c$	94,80%	1,0	0,0	-2,0	0,0
DP6	$\sigma \times b \times t = c$	95,60%	1,0	1,0	-2,0	0,0
DP7	$D/\sigma = c$	95,60%	-1,0	2,0	2,0	0,0
DP8	$D/(\sigma \times t) = c$	95,60%	-1,0	1,0	2,0	0,0

6.4 ESTUDO DE CASO 3: PROCESSO DE CORTE DE UMA BARRA DE METAL

6.4.1 Problema Multiobjetivo

Os parâmetros utilizados neste problema são os mesmos dos problemas da treliça de 2 barras (Tabela 6.1), com exceção do tamanho da população, adotando-se aqui 1000 indivíduos, como sugerido por Bandaru e Deb (2013a). Nas Figuras 6.5 e 6.6 são mostradas as Frentes de Pareto obtidas através da execução de uma e 30 execuções independentes, respectivamente. O número de soluções não-dominadas encontradas em cada uma das execuções independentes foi igual a 1000, enquanto que observou-se 6048 soluções não-dominadas ao juntar as soluções de todas as execuções independentes.

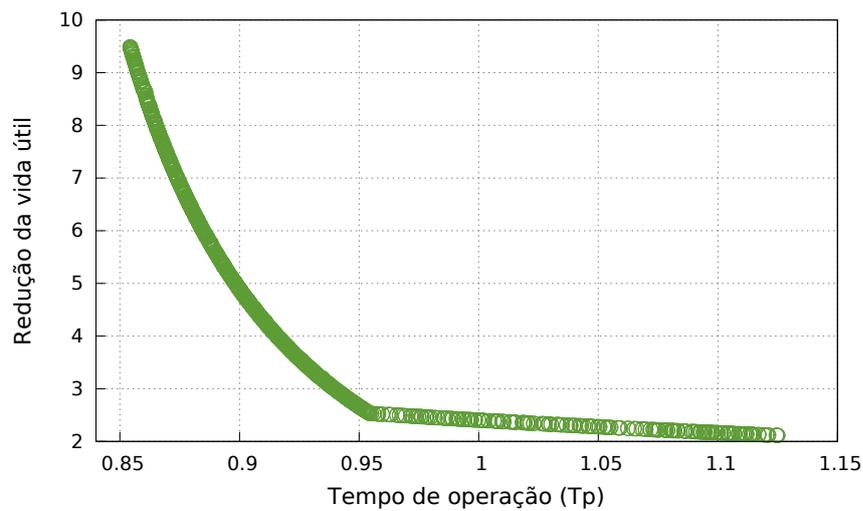


Figura 6.5: Frente de Pareto encontrada em uma execução do NSGA-II para o problema de otimização do processo de corte de uma peça metálica.

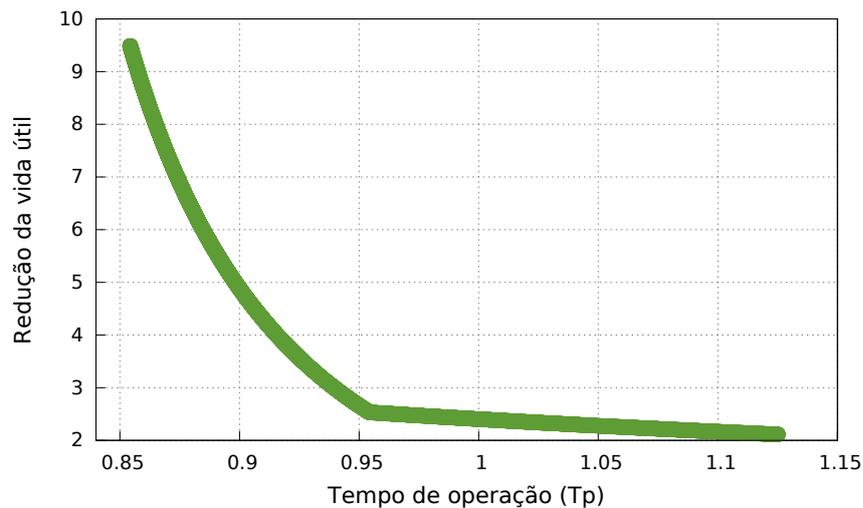


Figura 6.6: Frente de Pareto com as soluções não-dominadas obtidas por 30 execuções do NSGA-II para o problema de otimização do processo de corte de uma peça metálica.

6.4.2 *Descoberta de Conhecimento via Innovization*

Os parâmetros utilizados neste estudo de caso são mostrados na Tabela 6.12, enquanto as unidades das variáveis constam na Tabela 6.13. Assim como nos casos anteriores, a utilização do arquivo externo para a manutenção das soluções de interesse possibilitou encontrar um número maior de soluções e uma aptidão melhor, como pode ser observado na Tabela 6.14.

Tabela 6.12: Parâmetros adotados para o *Innovization* no estudo de caso do corte da barra de metal.

Parâmetro	Valor
População	1000
Gerações	100
Recombinação (PG)	Recombinação de sub-árvores com $p_c = 0,9$
Mutação (PG)	Mutação de ponto com $p_m = 0,2$ por nó
Recombinação (AG)	SBX com $p_c = 0,9$ e $\eta_c = 10$
Mutação (AG)	Polinomial com $p_m = 0,05$ e $\eta_m = 50$
Conjunto de funções	$\{+, -, *, /, \wedge\}$
Conjunto de terminais	$\{T_p, \xi, v, f, a, R\}$
Constantes efêmeras (R)	$\{-10,0, -9,5, -9,0, \dots, 9,0, 9,5, 10,0\}$
Dimensões básicas	{massa(lb), comprimento(in), tempo(s), vida útil}
Significância mínima (S_{min})	70%
Constante de agrupamento (ϵ)	3

Tabela 6.13: Unidades das variáveis utilizadas no estudo de caso do corte da barra metálica.

Variável	Unidades			
	Massa	Comprimento	Tempo	Vida útil
T_p	0	0	1	0
ξ	0	0	0	1
v	0	1	-1	0
f	0	1	0	0
a	0	1	0	0

Tabela 6.14: Resultados quantitativos obtidos para o estudo de caso do corte de uma barra metálica, considerando a utilização de arquivo externo.

Arquivo externo	Número de soluções			Aptidão		
	Mínimo	Máximo	Média	Mínimo	Máximo	Média
Não	2	8	5,27±1,77	3,09	7725,99	1593,65±1597,19
Sim	51	132	71,73±19,00	3,05	23277,42	2844,51±2542,35

Os princípios de projeto obtidos são listados na Tabela 6.15. Os modelos DP1 e DP11 eram esperados, uma vez que também foram encontrados por Deb e Datta (2011) e Bandaru e Deb (2013a). DP1 e DP2 sugerem que a taxa de alimentação f e a profundidade de corte a são praticamente constantes para as soluções Pareto Ótimas encontradas. O princípio DP11 indica, como se poderia esperar, que o tempo total de operação T_p está relacionado à velocidade de rotação v e à profundidade de corte a . O tempo de operação também está relacionado à taxa de alimentação da ferramenta, de acordo com o princípio DP3. O modelo DP14 apresentado aqui é assemelhante ao princípio DP12 apresentado na referência, diferindo somente pelo expoente da profundidade de corte.

Tabela 6.15: Princípios obtidos para o estudo de caso do corte de uma barra metálica.

Notação	Princípio	S	Aptidão	Unidades básicas			
				lb	in	s	vida útil
DP1	$f = c$	72,16%	3,04	0,0	1,0	0,0	0,0
DP2	$a = c$	71,96%	14,33	0,0	1,0	0,0	0,0
DP3	$f/T_p = c$	72,20%	15,26	0,0	1,0	-1,0	0,0
DP4	$1,00/(\xi \times f) = c$	72,67%	44,31	0,0	-1,0	0,0	-1,0
DP5	$v \times f = c$	71,67%	53,38	0,0	2,0	0,0	0,0
DP6	$1,00/(a \times v) = c$	76,88%	60,86	0,0	-2,0	1,0	0,0
DP7	$T_p^2 \times f = c$	79,88%	63,50	0,0	1,0	2,0	0,0
DP8	$T_p = c$	70,90%	67,72	0,0	0,0	1,0	0,0
DP9	$f/v = c$	79,45%	68,85	0,0	0,0	0,0	0,0
DP10	$a/f = c$	73,81%	239,94	0,0	0,0	0,0	0,0
DP11	$v \times T_p \times a = c$	73,86%	479,83	0,0	2,0	0,0	0,0
DP12	$(a^{4,5} \times f)/T_p = c$	77,74%	6075,4	0,0	5,5	-1,0	0,0
DP13	$T_p^2 \times v \times a = c$	72,70%	6107,12	0,0	2,0	0,0	0,0
DP14	$\xi \times a^{3,0} \times v = c$	81,71%	6292,75	0,0	4,0	-1,0	1,0
DP15	$a^{3,0} \times T_p \times \xi = c$	78,06%	6239,53	0,0	3,0	1,0	1,0

Tabela 6.16: Resultados obtidos por Bandaru e Deb (2013a) para o estudo de caso do corte de uma barra de metal.

Notação	Princípio	S	Unidades básicas			
			lb	in	s	vida útil
DP1	$v/(f^2 \times \xi) = c$	72,70%	0,0	-1,0	-1,0	-1,0
DP2	$(a \times v)/f = c$	74,60%	0,0	1,0	-1,0	0,0
DP3	$v/(f^2 \times T_p \times \xi) = c$	73,40%	0,0	-1,0	-2,0	-1,0
DP4	$f = c$	72,90%	0,0	1,0	0,0	0,0
DP5	$a/(f \times T_p) = c$	72,90%	0,0	0,0	-1,0	0,0
DP6	$(a^{5,5} \times f \times \xi)/T_p = c$	77,50%	0,0	6,5	-1,0	1,0
DP7	$(a \times T_p \times v)/f = c$	74,20%	0,0	1,0	0,0	0,0
DP8	$a^{5,5} \times T_p \times \xi = c$	82,60%	0,0	5,5	1,0	1,0
DP9	$a \times T_p \times v = c$	74,10%	0,0	2,0	0,0	0,0
DP10	$(a^2 \times T_p \times \xi)/v = c$	74,40%	0,0	1,0	2,0	1,0
DP11	$(a^2 \times \xi)/v = c$	76,00%	0,0	1,0	1,0	1,0
DP12	$a^{5,5} \times f \times \xi = c$	76,80%	0,0	6,5	0,0	1,0

6.5 ESTUDO DE CASO 4: PROJETO DE ENGRENAGENS COMPOSTAS

6.5.1 *Problema Multiobjetivo*

Como as variáveis do problema são discretas, foi utilizada uma codificação binária de 6 bits, que são suficientes para representar o intervalo desejado, como em Deb e Srinivasan (2006). Os parâmetros utilizados neste experimento estão listados na Tabela 6.17. O tamanho da população e número de gerações utilizados foram aqueles utilizados nos experimentos anteriores. Assim como adotado por Deb e Srinivasan (2006), aqui foi utilizada recombinação de um ponto e mutação bit-a-bit. As probabilidade de recombinação e taxa de mutação não foram apresentadas na referência e foram portanto determinadas empiricamente.

Tabela 6.17: Parâmetros adotados para o NSGA-II no problema das engrenagens compostas.

Parâmetro	Valor
População	500
Gerações	500
Recombinação	De um ponto, com probabilidade de 90%
Mutação	Bit-a-bit, com probabilidade de 30%

As Figuras 6.7 e 6.8 ilustram as Frentes de Pareto encontradas após uma e 30 execuções independentes do NSGA-II, respectivamente. O número de soluções não-dominadas encontradas foi igual a 16, no primeiro caso e 17, no segundo. É importante destacar que, neste caso, a abordagem de combinar as Frentes de Pareto obtidas em diferentes execuções possibilitou encontrar uma solução extrema não relatada por Deb e Srinivasan (2006). Tal solução consiste no vetor $\mathbf{x} = (16, 19, 49, 43)$, que apresenta erro (f_1) igual a $7,89 \times 10^{-5}$ e raio da maior engrenagem (f_2) igual a 49. Em relação ao objetivo f_1 isoladamente, a melhor solução reportada pela referência possui valor igual a $2,31 \times 10^{-4}$.

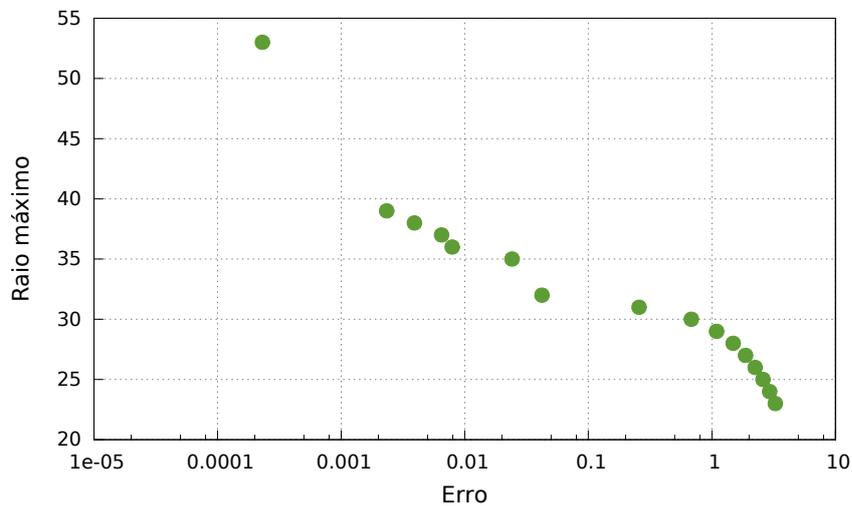


Figura 6.7: Frente de Pareto encontrada em uma execução do NSGA-II para o problema das engrenagens compostas.

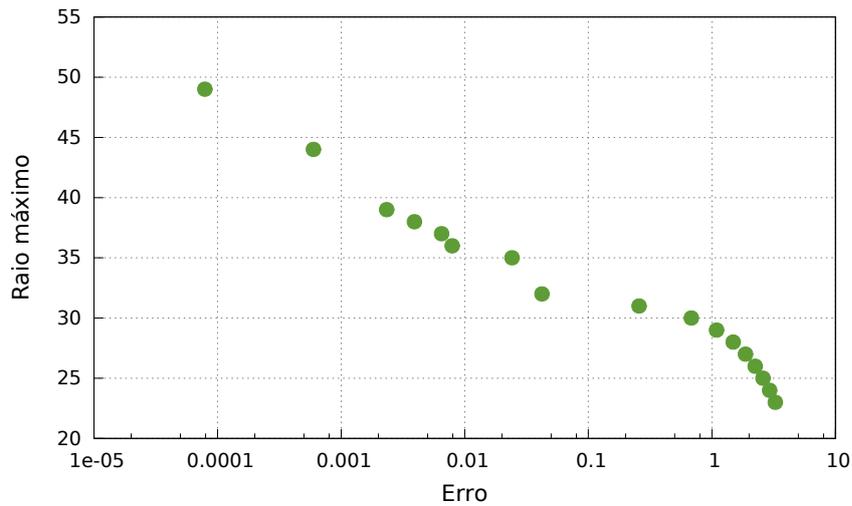


Figura 6.8: Frente de Pareto com as soluções não-dominadas obtidas por 30 execuções do NSGA-II para o problema das engrenagens compostas.

6.5.2 *Descoberta de Conhecimento via Innovization*

Os parâmetros adotados neste estudo de caso foram os mesmos utilizados na seção anterior, excetuando-se o conjunto de terminais, que neste caso é:

$$F = \{Erro, Max, x_1, x_2, x_3, x_4, R\}.$$

Foram utilizadas duas dimensões básicas, o comprimento e o erro, atrelado a uma das funções objetivo do problema. Como pode ser observado na Tabela 6.18 e assim como observados nos casos anteriores, a utilização de um arquivo externo favoreceu a obtenção de mais princípios de projeto e uma aptidão melhor.

Tabela 6.18: Resultados quantitativos obtidos para o estudo de caso das engrenagens compostas, considerando a utilização de arquivo externo.

Arquivo externo	Número de soluções			Aptidão		
	Mínimo	Máximo	Média	Mínimo	Máximo	Média
Não	3	47	18,03±12,46	3	3494,68	557,32±662,58
Sim	112	546	273,03±115,41	1	3503,68	736,16±731,74

Os modelos obtidos para o presente estudo de caso são exibidos na Tabela 6.19. Este problema foi abordado na proposta inicial de *Innovization* (Deb e Srinivasan 2006), em

que alguns princípios foram obtidos. Diferentemente dos casos anteriores, no trabalho citado não foi apresentada uma tabela com as inovações encontradas e suas respectivas significâncias, uma vez que foi o estudo inicial de *Innovization*, em que a análise foi realizada manualmente, sem o emprego de um algoritmo de agrupamento. O princípio DP1, que possui a maior significância dentre os mostrados, sugere que x_2 é constante em quase 90% da Frente de Pareto. Os princípios DP3 e DP7 indicam relações entre as variáveis x_3 e x_4 , e o objetivo *Max*, que consiste em minimizar o valor da maior engrenagem. Assim, pode-se inferir que estas duas engrenagens, x_3 e x_4 , são as que assumem os maiores valores, assim como indicado pela análise de Deb e Srinivasan (2006). De acordo com esta referência, as variáveis x_1 e x_2 sempre assumem os menores valores possíveis, de forma a atingir o segundo objetivo. O princípio DP10 apresentado aqui sugere uma relação constante entre x_1 e x_2 .

Tabela 6.19: Princípios obtidos para o estudo de caso das engrenagens compostas.

Notação	Princípio	S	Aptidão	Unidades básicas	
				Comprimento	Erro
DP1	$x_2 = c$	88,23%	1,00	1,0	0,0
DP2	$(Max \times x_4)/x_2 = c$	76,47%	5,00	1,0	0,0
DP3	$x_4 - Max = c$	76,47%	6,00	1,0	0,0
DP4	$(x_4 - Max)/(x_3 - x_2) = c$	76,47%	7,00	0,0	0,0
DP5	$x_1 \times Erro = c$	88,23%	9,57	1,0	1,0
DP6	$(x_1 - x_2)/Erro = c$	88,23%	10,29	1,0	0,0
DP7	$x_3 - Max = c$	76,47%	12,00	1,0	0,0
DP8	$x_3/x_4 = c$	70,59%	25,72	0,0	0,0
DP9	$x_2 - x_4 = c$	70,59%	64,24	1,0	0,0
DP10	$(x_1 + x_2)^{-2,0} = c$	76,47%	64,30	-2,0	0,0

6.6 ESTUDO DE CASO 5: PROJETO DE UMA TRELIÇA DE 10 BARRAS

6.6.1 Problema Multiobjetivo

Neste caso, torna-se mais conveniente medir a qualidade dos modelos candidatos utilizando um simulador para realizar os cálculos de peso, tensões das barras e deslocamentos dos nós da estrutura. Neste trabalho foi utilizado um simulador desenvolvido em C++³, o qual foi aqui acoplado à etapa de avaliação do NSGA-II.

O problema foi resolvido utilizando os parâmetros presentes na Tabela 6.20, conforme recomendado por Vargas (2015). A Frente de Pareto obtida pela primeira execução independente é mostrada na Figura 6.9, enquanto a Figura 6.10 exibe a Frente de Pareto contendo as soluções não dominadas de 30 execuções independentes.

Tabela 6.20: Parâmetros adotados para o NSGA-II no problema da treliça de 10 barras.

Parâmetro	Valor
População	500
Gerações	500
Recombinação	SBX com $p_c = 1$ e $\eta_c = 2$
Mutação	Polinomial com $p_m = 0,05$ e $\eta_m = 100$

³Disponível em <http://sourceforge.net/projects/eurekaoptima>.

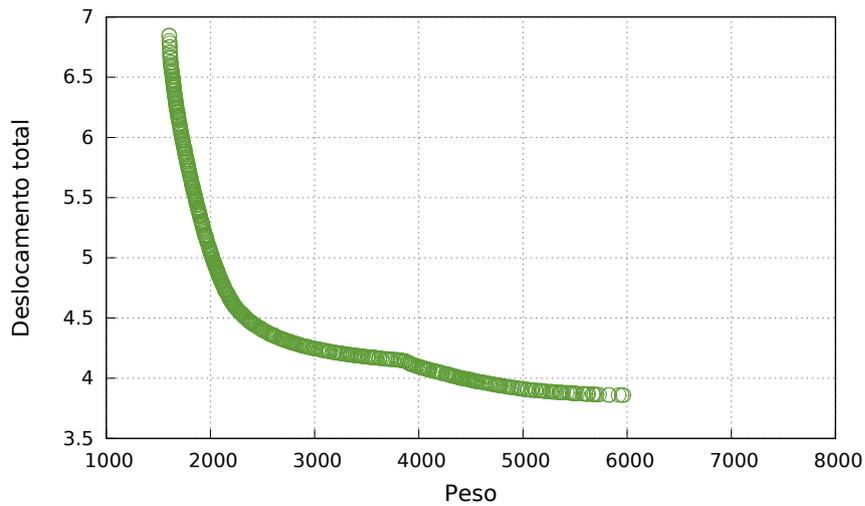


Figura 6.9: Frente de Pareto encontrada em uma execução do NSGA-II para o problema da treliça de 10 barras.

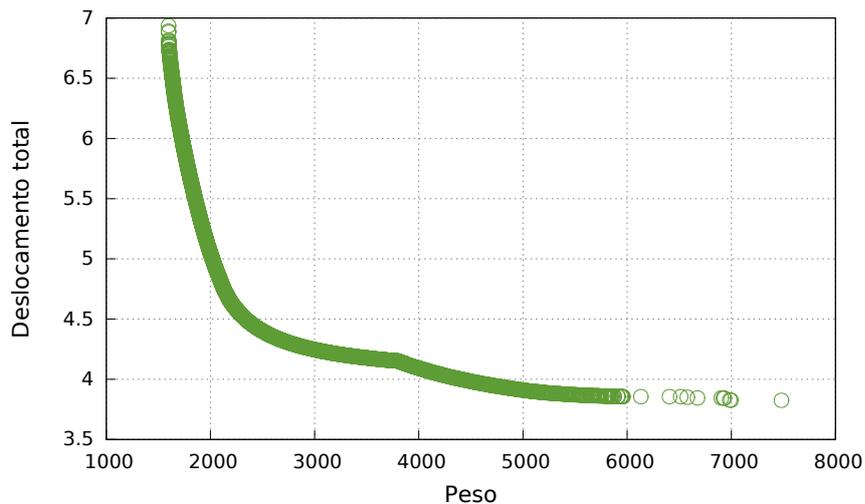


Figura 6.10: Frente de Pareto com as soluções não-dominadas obtidas por 30 execuções do NSGA-II para o problema da treliça de 10 barras.

6.6.2 *Descoberta de Conhecimento via Innovization*

Para a busca por princípios de projetos envolvendo o estudo de caso da treliça de 10 barras, foram adotados os mesmos parâmetros utilizados para a treliça de 2 barras, com exceção do conjunto de terminais, que passa a ser $F = \{Peso, Deslocamento, x_1, x_2, \dots, x_{10}\}$, onde x_i representa a área da seção transversal da i -ésima barra. Foram utilizadas duas dimensões básicas: massa (lb) e comprimento (in). A utilização do arquivo externo foi vantajosa neste estudo de caso, como pode ser observado na Tabela 6.21 pelo aumento do

número de soluções obtidas e pela aptidão melhor.

Tabela 6.21: Resultados quantitativos obtidos para o estudo de caso da estrutura de 10 barras, considerando a utilização de arquivo externo.

Arquivo externo	Número de soluções			Aptidão		
	Mínimo	Máximo	Média	Mínimo	Máximo	Média
Não	10	101	31,83±26,08	13,95	33342,67	671,06±1424,35
Sim	167	2201	608,47±465,62	4,09	43460,66	1107,95±1974,84

Diversos princípios de projeto foram obtidos neste estudo de caso, sendo os 50 melhores (em relação à sua aptidão) mostrados na Tabelas 6.22 e 6.23. Pode-se notar que a maioria dos modelos encontrados apresenta significância alta, acima de 90%. Os princípios DP1, DP2, DP4, DP5, DP29, DP36 e DP48 representam áreas de seção transversal de barras com valores constantes em algumas regiões da Frente de Pareto. Ao todo foram obtidos 26 princípios envolvendo a razão ou multiplicação entre áreas: DP7, DP9, DP10, DP12, DP13, DP14, DP15, DP17, DP19, DP20, DP24, DP25, DP27, DP32, DP34, DP35, DP39, DP40, DP42, DP43, DP44, DP45, DP46, DP49 e DP50. Alguns princípios envolvem a soma e diferença entre áreas, como os princípios DP11, DP16, DP22, DP23, DP26, DP27, DP30, e DP47. Relacionados ao objetivo de minimização do peso da estrutura foram encontrados 5 princípios: DP3, DP6, DP18, DP31 e DP33. Os princípios DP21, DP38 e DP41 apresentam relações entre deslocamento total e áreas das barras.

É importante ressaltar que apesar de haver sido encontrados princípios em que as áreas das seções transversais de algumas barras são constantes, estes podem não apresentar o mesmo valor em toda a Frente de Pareto. A medida de qualidade das soluções é dada por um algoritmo de agrupamento, sendo os objetivos do processo de otimização encontrar princípios que geram poucos grupos com baixo coeficiente de variação. Desta forma, um princípio pode apresentar baixa variação em cada grupo, mas possuir grupos com valores distantes entre si. Desta forma, apesar da notação utilizada ser, por exemplo, $X_4 = c$, o princípio pode apresentar alguns valores distintos, cada um praticamente constante ao longo de um subconjunto da Frente de Pareto. Nas Figuras 6.11 e 6.12 são mostrados, respectivamente, os grupos obtidos pelo princípio DP1 e uma ilustração Frente de Pareto em que são destacadas as regiões correspondentes a cada grupo. Os grupos são identificados nas duas Figuras por suas cores. Pode-se perceber que há

um grupo (destacado em vermelho) que engloba aproximadamente 1900 soluções, das 2334 utilizadas, apresentando baixa variação. Todavia, há grupos que apresentam outros valores, que correspondem a regiões distintas da Frente de Pareto.

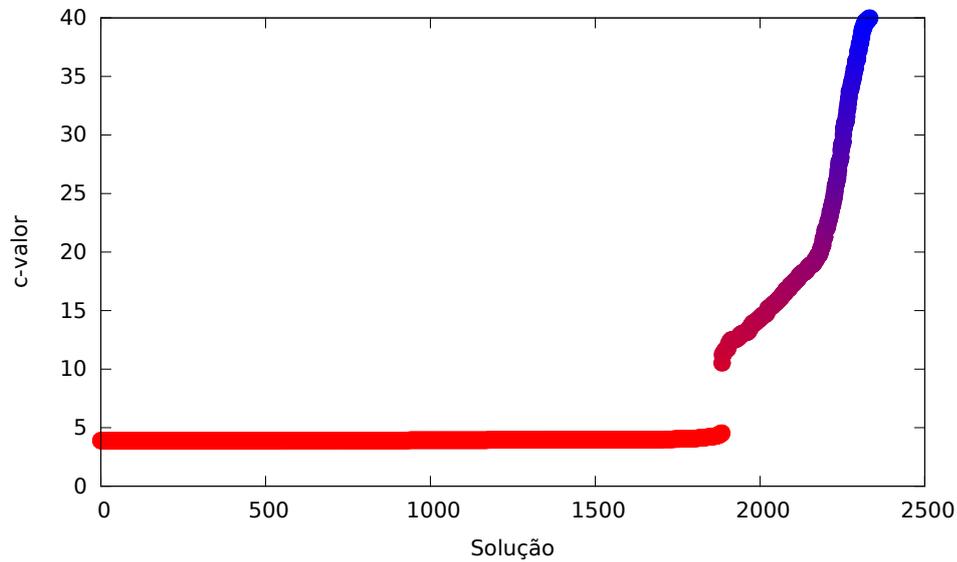


Figura 6.11: Ilustração dos grupos obtidos para o princípio DP1, juntamente com os valores obtidos para cada solução.

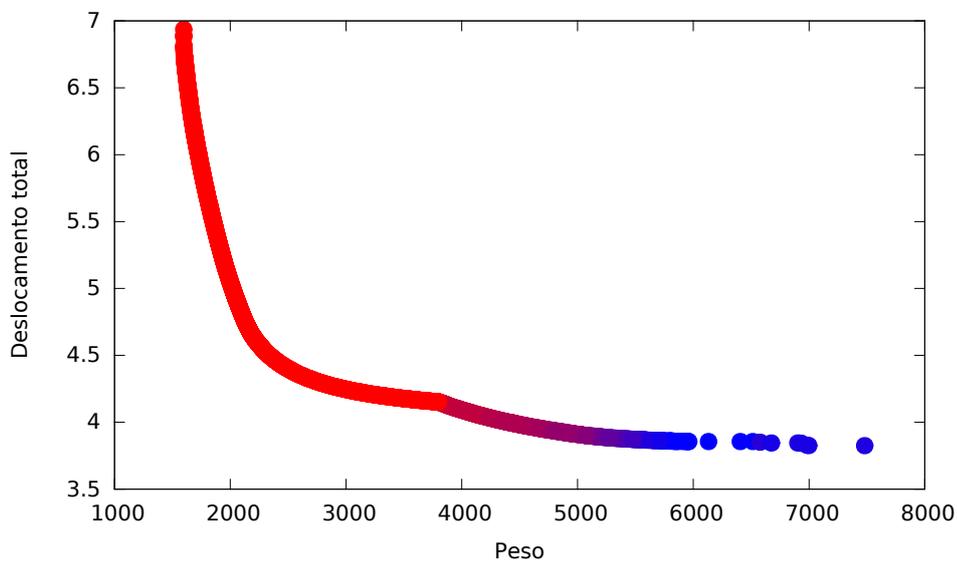


Figura 6.12: Ilustração da Frente de Pareto obtida para o problema das 10 barras, em que são destacadas as regiões correspondentes aos grupos mostrados na Figura 6.11.

Tabela 6.22: Primeira parte dos princípios de projeto obtidos para o estudo de caso da estrutura de 10 barras. As dimensões básicas utilizadas são massa (lb) e comprimento(in).

Notação	Princípio	S	Aptidão	Unidades básicas	
				lb	in
DP1	$X_4 = c$	80,76%	3,59	0,00	2,00
DP2	$X_5 = c$	98,97%	4,09	0,00	2,00
DP3	$Peso^2/X_6 = c$	99,40%	5,00	2,00	-2,00
DP4	$X_8 = c$	99,96%	5,00	0,00	2,00
DP5	$X_9 = c$	99,70%	5,00	0,00	2,00
DP6	$Peso^2/X_2 = c$	99,53%	5,00	2,00	-2,00
DP7	$X_2/X_6 = c$	82,73%	5,39	0,00	0,00
DP8	$(X_8 - X_1) \times X_5 = c$	99,44%	6,24	0,00	4,00
DP9	$X_3 \times X_5 = c$	98,67%	7,11	0,00	4,00
DP10	X_5/X_3	98,67%	7,32	0,00	0,00
DP11	$X_{10} + 6,00X_4 = c$	80,76%	7,41	0,00	2,00
DP12	$(X_5 - X_3) \times X_5 = c$	98,84%	7,71	0,00	4,00
DP13	$(X_5 - X_{10}) \times X_5 = c$	99,49%	7,82	0,00	4,00
DP14	$X_7/(X_7 - X_5) = c$	99,27%	7,83	0,00	0,00
DP15	$X_4/X_1 = c$	80,76%	8,44	0,00	0,00
DP16	$8,00/(2X_6 - X_7) = c$	99,87%	8,67	0,00	-2,00
DP17	$X_1 \times X_6 \times X_{10}^2 = c$	99,66%	9,06	0,00	8,00
DP18	$X_4 \times X_{10} \times X_5 \times Peso = c$	99,53%	9,24	1,00	6,00
DP19	$(X_2 - X_3) \times X_5 = c$	98,11%	9,26	0,00	4,00
DP20	$(1,00/X_4) \times X_6^2 \times X_{10} = c$	99,19%	9,36	0,00	4,00
DP21	$(X_7 \times Deslocamento)^6$	99,96%	10,00	0,00	3,00
DP22	$X_5 + (X_1 - X_4) = c$	82,78%	10,37	0,00	2,00
DP23	$X_5 - X_6 = c$	90,57%	10,62	0,00	2,00
DP24	$X_6 \times X_{10} = c$	99,53%	10,65	0,00	4,00
DP25	$X_7/(-2X_6 - X_7) = c$	99,87%	11,84	0,00	4,00

Tabela 6.23: Segunda parte dos princípios de projeto obtidos para o estudo de caso da estrutura de 10 barras. As dimensões básicas utilizadas são massa (lb) e comprimento(in).

Notação	Princípio	S	Aptidão	Unidades básicas	
				lb	in
DP26	$(X_8 - X_1) = c$	97,90%	11,91	0,00	2,00
DP27	$(X_{10} - X_4) = c$	80,76%	12,78	0,00	2,00
DP28	$(X_2/(X_7 - X_8)) = c$	98,11%	13,30	0,00	0,00
DP29	$X_7 = c$	84,23%	14,42	0,00	2,00
DP30	$X_2 - X_4 = c$	80,29%	14,72	0,00	2,00
DP31	$X_5 \times Peso = c$	98,84%	14,96	1,00	2,00
DP32	$(X_6 \times X_{10})/X_2 = c$	99,66%	14,99	0,00	2,00
DP33	$X_{10} \times Peso^3 = c$	99,57%	15,00	3,00	2,00
DP34	$X_3 \times X_{10} \times X_6 = c$	99,53%	15,05	0,00	6,00
DP35	$X_8/X_7 = c$	80,76%	15,63	0,00	0,00
DP36	$X_1 = c$	98,29%	16,19	0,00	2,00
DP37	$X_{10} \times X_5$	99,14%	17,64	0,00	4,00
DP38	$Deslocamento/(X_6 - X_3)$	99,01%	23,60	0,00	-1,00
DP39	$(X_2/X_5) = c$	80,63%	29,28	0,00	0,00
DP40	$(X_8/X_1) = c$	88,30%	29,41	0,00	0,00
DP41	$Deslocamento \times X_2 = c$	90,62%	39,37	0,00	3,00
DP42	$(X_6/X_5) = c$	80,72%	40,08	0,00	0,00
DP43	$(X_2/X_6) = c$	83,38%	42,13	0,00	0,00
DP44	$(X_1/X_8) = c$	89,25%	43,17	0,00	0,00
DP45	$X_3/(X_5 + X_3) = c$	98,11%	43,23	0,00	0,00
DP46	$(X_7/X_5) = c$	89,46%	65,55	0,00	0,00
DP47	$(X_7 - X_{10}) = c$	94,86%	65,60	0,00	2,00
DP48	$X_{10} = c$	89,93%	65,40	0,00	2,00
DP49	$X_6/X_2 = c$	99,14%	96,12	0,00	0,00
DP50	$X_5/X_7 = c$	99,06%	284,62	0,00	0,00

Tomando os princípios encontrados para este estudo de caso, foi realizado um estudo inicial para verificar se o conhecimento obtido pode ser utilizado para simplificar o

processo de otimização, de forma a obter-se soluções para o problema multiobjetivo mais rapidamente. Como foram encontradas relações envolvendo as áreas das seções de uma e duas barras, tal informação poderia ser utilizada para reduzir o número de variáveis do problema original, uma vez que o valor de algumas delas é conhecido ao longo da Frente de Pareto. Em trabalhos futuros, o conhecimento adquirido poderá ser aplicado a outros problemas com características semelhantes.

Os princípios de projeto DP2, DP36, DP49 e DP50, que possuem alta significância, foram utilizados para transformar o problema original, que possui 10 variáveis, num outro (mais simples) contendo 6 variáveis. Para cada princípio foi calculado o valor médio de seu maior grupo para representar a constante do lado direito das expressões. Assim foram utilizadas as transformações $x_1 = 8,4600$, $x_5 = 0,1053$, $x_6 = x_2 \times 0,9990$ e $x_5 = x_7 \times 75,7790$. O novo problema multiobjetivo foi resolvido com o mesmo método utilizado anteriormente, gerando-se uma Frente de Pareto a partir de 30 execuções independentes do NSGA-II. Os parâmetros utilizados foram os mesmos descritos nesta seção.

Na Figura 6.13, são mostradas as Frentes de Pareto obtidas resolvendo-se os problemas original e simplificado. Nesta Figura pode-se perceber que a Frente de Pareto encontrada para o problema simplificado se aproxima daquela obtida para o problema original.

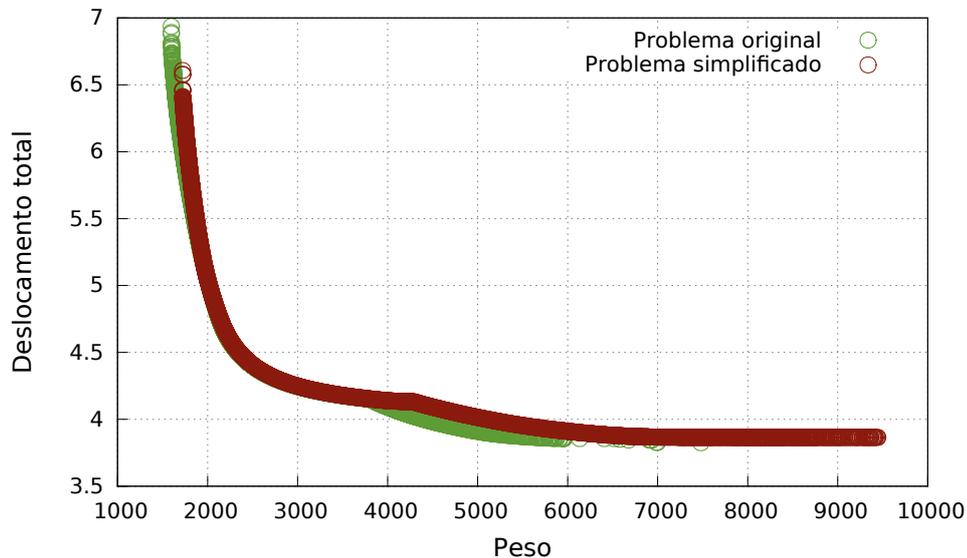


Figura 6.13: Frentes de Pareto obtidas ao resolver os problemas da treliça de 10 barras original e simplificado (utilizando 6 variáveis ao invés das 10 originais).

Para verificar a qualidade da solução obtida, o hipervolume da Frente de Pareto foi calculado e comparado ao daquela obtida através da resolução do problema original. O

cálculo foi realizado utilizando uma ferramenta baseada em Fonseca, Paquete e López-Ibáñez (2006) e disponível publicamente⁴. O ponto de referência adotado foi determinado pelos maiores valores de cada objetivo considerando as duas Frentes de Pareto, resultando no ponto $W = (9444,71, 6,94)$. O hipervolume da Frente de Pareto original é igual a 22401,38, enquanto o da nova Frente possui valor 21970,33. A diferença relativa entre os valores dos hipervolumes é de 1,9%, mostrando que a solução para o problema simplificado é uma boa aproximação para o problema original.

Neste capítulo foram apresentados experimentos computacionais com o intuito de verificar se as modificações propostas são capazes de incentivar a obtenção de princípios de projeto mais diversos. Cinco estudos de caso em engenharia foram analisados, sendo os quatro primeiros já abordados na literatura de *Innovization*, enquanto a aplicação do último, que consiste no projeto de uma treliça de 10 barras, foi proposta neste trabalho. Os resultados obtidos mostram que as propostas foram capaz de gerar princípios adicionais àqueles observados na literatura, aumentando assim a capacidade de geração de novos conhecimentos. Em todos os estudos de caso observou-se que a utilização do arquivo externo para armazenamento das soluções de interesse ao longo do processo de busca influencia positivamente a obtenção de maior número de soluções e na obtenção de um princípio com aptidão melhor.

No primeiro estudo de caso, envolvendo o projeto de uma treliça de duas barras, todos os princípios previstos pela solução analítica foram redescobertos aqui, além de um princípio adicional relacionando a área das seções transversais das barras. O processo de descoberta de conhecimento foi também aplicado ao segundo estudo de caso, referente ao projeto de uma viga soldada, em que todas os princípios apresentados pela referência (Bandaru e Deb 2013a) foram encontrados, além de um outro sugerido que havia sido um estudo anterior (Deb e Srinivasan 2006). Este princípio adicional encontrado para o problema da viga soldada é relevante, pois indica que não é possível encontrar soluções melhores sem que uma restrição seja modificada (por exemplo, ao utilizar um material diferente). Para o terceiro problema tratado, o corte de uma barra metálica, foram encontrados 15 princípios, dos quais 2 estão entre os 12 apresentados pela referência. No estudo de caso 4 (projeto de engrenagens compostas), foram identificados 10 princípios, dos quais 3 foram sugeridos pelo trabalho de Deb e Srinivasan (2006). Neste último, dois

⁴A versão mais recente pode ser acessada em <http://lopez-ibanez.eu/hypervolume>

dos princípios relatados pela referência não foram encontrado aqui, que sugerem haver relações constantes envolvendo os diâmetros das engrenagens: x_3/x_1 e x_4/x_2 . No estudo de caso 5, o projeto de uma treliça de 10 barras, foram encontrados 50 princípios, dos quais 8 indicam áreas que apresentam valores constantes em algumas regiões da Frente de Pareto, 26 envolvem multiplicação e razão entre áreas das seções transversais das barras, 8 representam soma e subtração entre áreas, 5 cujas expressões utilizam o objetivo de minimização do peso da estrutura e 3 que apresentam relações entre o objetivo de minimização do deslocamento.

O conhecimento inferido no estudo de caso da treliça de 10 barras foi utilizado em um estudo inicial para reduzir o número de variáveis do problema multiobjetivo, uma vez que os valores de algumas variáveis se mantêm praticamente constantes ao longo da Frente de Pareto. Observou-se que a Frente de Pareto obtida ao resolver o problema de otimização multiobjetivo simplificado apresenta o mesmo comportamento da solução obtida para o problema original. Este método poderá ser estendido em trabalhos futuros, aplicando-se o conhecimento adquirido ao resolver um problema a outros que possuam características semelhantes.

7 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foram propostas três modificações para o processo de *Innovization* baseado em Programação Genética: (i) uma alternativa para promover a consistência na utilização das unidades; (ii) a utilização de um arquivo externo à busca para manter as soluções promissoras; e (iii) um procedimento para evitar a obtenção de soluções triviais e, portanto, irrelevantes. A partir dos resultados obtidos em cinco estudos de caso em engenharia pode-se concluir que as propostas possibilitam a descoberta de princípios de projeto, uma vez que a maioria das inovações presentes na literatura foi redescoberta aqui. Particularmente em relação à utilização de um arquivo externo à população, verificou-se que contribui para a obtenção de um número maior de soluções e, assim, pode propiciar a descoberta de novos princípios de projeto. Nos casos testados, além de maior diversidade foi também possível obter soluções com melhores valores de aptidão quando a estratégia de arquivo externo foi adotada.

Pode-se destacar ainda que os experimentos realizados com a estrutura da treliça de 10 barras apresentaram resultados promissores, possibilitando a descoberta de diversos princípios de projeto. Assim, a aplicação de *Innovization* a problemas de otimização estrutural representa um caminho interessante de pesquisa, dado seu potencial de descoberta de conhecimento sobre o domínio. Como trabalhos futuros, indica-se a aplicação da técnica a problemas de otimização estrutural de maior porte e mais complexos.

Além disso, a utilização do conhecimento obtido através do processo de *Innovization* para reduzir a dimensão de problemas de otimização pode ser explorada. Por exemplo, pode ser analisado o ganho em relação ao custo computacional, uma vez que o espaço de busca é reduzido. Outra possibilidade importante e útil aos projetistas é avaliar o impacto da replicação do conhecimento adquirido num problema a outros com características semelhantes.

Além da aplicação de *Innovization* a otimização no domínio de engenharia, que já se mostrou capaz de descobrir relações entre suas variáveis, objetivos e restrições,

um nicho potencial de aplicação da técnica são problemas de aprendizado de máquina supervisionado, como aqueles de regressão e classificação. Neste caso, a técnica pode expandir o conhecimento do problema em estudo, a partir da descoberta de relações simbólicas entre suas variáveis. A aplicação a problemas de agrupamento (aprendizado não supervisionado) também poderá ser explorada com o objetivo de identificar os padrões que levam à formação dos grupos.

Em relação ao método de *Innovization*, foram identificadas algumas ideias que podem ser exploradas em estudos futuros. Inicialmente, a definição dos nichos para a etapa de recombinação utiliza somente a informação de quais variáveis de entradas são empregadas por cada modelo. Outras informações poderiam ser incluídas nesta definição, como as funções utilizadas. Além disso, a noção de distância entre os modelos pode considerar não somente a estrutura das árvores (fenótipo), como também seu aspecto semântico, que considera o comportamento dos modelos, representado pela saída obtida a partir de suas execuções.

No método proposto para evitar a obtenção de soluções triviais, foi utilizada a hipótese de que se todos os c -valores obtidos a partir da execução de um princípio são iguais, trata-se de uma solução trivial, que não agrega conhecimento ao problema em estudo. Uma outra abordagem que poderia ser utilizada é a simplificação simbólica, que seria capaz de identificar quando um princípio não depende, de fato, de nenhuma das variáveis de entrada. Como desvantagens, esta abordagem possui alto custo computacional e poderia classificar incorretamente alguns princípios como não-triviais. Todavia, tais características não invalidam sua utilização, que pode ser considerada em trabalhos futuros.

REFERÊNCIAS

- [1] Jaqueline S Angelo, Heder S Bernardino e Helio JC Barbosa. “Ant colony approaches for multiobjective structural optimization problems with a cardinality constraint”. Em: *Advances in Engineering Software* 80 (2015), pp. 101–115.
- [2] Sameh Askar e Ashutosh Tiwari. “Finding Innovative Design Principles for Multiobjective Optimization Problems”. Em: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 41.4 (2011), pp. 554–559.
- [3] Sunith Bandaru e Kalyanmoy Deb. “A dimensionally-aware genetic programming architecture for automated innovization”. Em: *International Conference on Evolutionary Multi-Criterion Optimization*. Springer. 2013, pp. 513–527.
- [4] Sunith Bandaru e Kalyanmoy Deb. “Automated discovery of vital knowledge from Pareto-optimal solutions: First results from engineering design”. Em: *IEEE Congress on Evolutionary Computation*. IEEE. 2010, pp. 1–8.
- [5] Sunith Bandaru e Kalyanmoy Deb. “Automated innovization for simultaneous discovery of multiple rules in bi-objective problems”. Em: *International Conference on Evolutionary Multi-Criterion Optimization*. Springer. 2011, pp. 1–15.
- [6] Sunith Bandaru e Kalyanmoy Deb. “Higher and lower-level knowledge discovery from Pareto-optimal sets”. Em: *Journal of Global Optimization* 57.2 (2013), pp. 281–298.
- [7] Sunith Bandaru e Kalyanmoy Deb. “Towards automating the discovery of certain innovative design principles through a clustering-based optimization technique”. Em: *Engineering Optimization* 43.9 (2011), pp. 911–941.
- [8] Sunith Bandaru et al. “Generalized higher-level automated innovization with application to inventory management”. Em: *European Journal of Operational Research* 243.2 (2015), pp. 480–496.
- [9] Sunith Bandaru et al. “Higher-level innovization: A case study from friction stir welding process optimization”. Em: *2011 IEEE Congress of Evolutionary Computation (CEC)*. IEEE. 2011, pp. 2782–2789.

- [10] Harold P Benson. “Existence of efficient solutions for vector maximization problems”. Em: *Journal of Optimization Theory and Applications* 26.4 (1978), pp. 569–580.
- [11] Alexander E. I. Brownlee e Jonathan A Wright. “Solution analysis in multi-objective optimization”. Em: *2012 Building Simulation and Optimization*. Loughborough University© IBPSA-England, 2012, pp. 513–527.
- [12] Kazuhisa Chiba et al. “Design Exploration of Shielding Effect for Aircraft Engine Noise”. Em: *ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics, Egmond aan Zee, The Netherlands, September 5-8, 2006*. Delft University of Technology; European Community on Computational Methods in Applied Sciences (ECCOMAS). 2006, p. 2006.
- [13] Karim J Chichakly e Margaret J Eppstein. “Discovering design principles from dominated solutions”. Em: *IEEE Access* 1 (2013), pp. 275–289.
- [14] Carlos A Coello Coello, David A Van Veldhuizen e Gary B Lamont. *Evolutionary algorithms for solving multi-objective problems*. Vol. 242. Springer, 2002.
- [15] A. Colorni, M. Dorigo e V. Maniezzo. “Distributed optimization by ant colonies”. Em: *In Proc. of the European Conference on Artificial Life, pages 134-142, Paris, France. Elsevier* (1991).
- [16] C. Darwin. *On the origin of species*. John Murray, 1859.
- [17] Kalyanmoy Deb. “An efficient constraint handling method for genetic algorithms”. Em: *Computer Methods in Applied Mechanics and Engineering* 186.2 (2000), pp. 311–338.
- [18] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*. Vol. 16. John Wiley & Sons, 2001.
- [19] Kalyanmoy Deb. “Optimal design of a welded beam via genetic algorithms”. Em: *The American Institute of Aeronautics and Astronautics (AIAA) Journal* 29.11 (1991), pp. 2013–2015.
- [20] Kalyanmoy Deb e Rituparna Datta. “Hybrid evolutionary multi-objective optimization of machining parameters”. Em: *KanGAL Report* 201105 (2011).

- [21] Kalyanmoy Deb e Mayank Goyal. “A combined genetic adaptive search (GeneAS) for engineering design”. Em: *Computer Science and Informatics* 26 (1996), pp. 30–45.
- [22] Kalyanmoy Deb e Aravind Srinivasan. “Innovization: innovating design principles through optimization.” Em: *GECCO*. Ed. por Mike Cattolico. ACM, 2006, pp. 1629–1636. ISBN: 1-59593-186-4. URL: <http://dblp.uni-trier.de/db/conf/gecco/gecco2006.html#DebS06a>.
- [23] Kalyanmoy Deb et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. Em: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 182–197.
- [24] Stéphane Doncieux e Mohamed Hamdaoui. “Evolutionary algorithms to analyse and design a controller for a flapping wings aircraft”. Em: *New Horizons in Evolutionary Robotics*. Springer, 2011. Cap. 6, pp. 67–83.
- [25] J. Farmer, N. Packard e A Perelson. “The immune system, adaptation, and machine learning.” Em: *Physica D*, 2(1-3):187-204. (1986).
- [26] Carlos M Fonseca, Luís Paquete e Manuel López-Ibáñez. “An improved dimension-sweep algorithm for the hypervolume indicator”. Em: *2006 IEEE International Conference on Evolutionary Computation*. IEEE. 2006, pp. 1157–1163.
- [27] Fred W. Glover e Gary A. Kochenberger. *Handbook of Metaheuristics*. Hardcover. 2003.
- [28] David E Golberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, 1989.
- [29] Y. V. Haimes, L. S. Lasdon e D. A Wismer. “On a bicriterion formation of the problems of integrated system identification and system optimization”. Em: *IEEE Transactions on Systems, Man and Cybernetics* SMC-1.3 (jul. de 1971), pp. 296–297. ISSN: 0018-9472.
- [30] Prabhat Hajela e C-Y Lin. “Genetic search strategies in multicriterion optimal design”. Em: *Structural Optimization* 4.2 (1992), pp. 99–107.
- [31] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

- [32] B. K. Kannan e Steven N. Kramer. “An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design”. Em: *Journal of Mechanical Design* 116.2 (1994), pp. 405–411.
- [33] J. Kennedy e R. Eberhart. “Particle Swarm Optimization”. Em: *In Proc. IEEE International Conf. on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ* (1995).
- [34] Teuvo Kohonen. “The self-organizing map”. Em: *Proceedings of the IEEE* 78.9 (1990), pp. 1464–1480.
- [35] J. R Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, 1992.
- [36] Krzysztof Krawiec. “Semantic Genetic Programming”. Em: *Behavioral Program Synthesis with Genetic Programming*. Springer, 2016, pp. 55–66.
- [37] Fumiya Kudo e Tomohiro Yoshikawa. “Knowledge extraction in multi-objective optimization problem based on visualization of Pareto solutions”. Em: *2012 IEEE Congress on Evolutionary Computation*. IEEE. 2012, pp. 1–6.
- [38] Joel Lehman e Kenneth O Stanley. “Novelty search and the problem with objectives”. Em: *Genetic Programming Theory and Practice IX*. Springer, 2011, pp. 37–56.
- [39] Guan-Chun Luh e Chung-Huei Chueh. “Multi-objective optimal design of truss structure with immune algorithm”. Em: *Computers & Structures* 82.11 (2004), pp. 829–844.
- [40] Achille Messac e Christopher A Mattson. “Normal constraint method with guarantee of even representation of complete Pareto frontier”. Em: *The American Institute of Aeronautics and Astronautics (AIAA) Journal* 42.10 (2004), pp. 2101–2111.
- [41] Zbigniew Michalewicz. *Genetic algorithms and data structures - evolution programs*. Springer, 1996, pp. I–XX, 1–387. ISBN: 978-3-540-60676-5.
- [42] Shigeru Obayashi, Shinkyu Jeong e Kazuhisa Chiba. “Multi-objective design exploration for aerodynamic configurations”. Em: *The American Institute of Aeronautics and Astronautics (AIAA) Journal* (2005), p. 4666.

- [43] Shigeru Obayashi e Daisuke Sasaki. “Visualization and data mining of Pareto solutions using self-organizing map”. Em: *International Conference on Evolutionary Multi-Criterion Optimization*. Springer. 2003, pp. 796–809.
- [44] Akira Oyama, Taku Nonomura e Kozo Fujii. “Data mining of Pareto-optimal transonic airfoil shapes using proper orthogonal decomposition”. Em: *Journal of Aircraft* 47.5 (2010), pp. 1756–1762.
- [45] Ian SJ Packham e Ian C Parmee. “Data analysis and visualisation of cluster-oriented genetic algorithm output”. Em: *Information Visualization, 2000. Proceedings. IEEE International Conference on*. IEEE. 2000, pp. 173–178.
- [46] N Palli et al. “An interactive multistage ε -inequality constraint method for multiple objectives decision making”. Em: *Journal of Mechanical Design* 120.4 (1998), pp. 678–686.
- [47] Panos Y Papalambros e Douglass J Wilde. *Principles of optimal design: modeling and computation*. Cambridge University Press, 2000.
- [48] Riccardo Poli, William B. Langdon e Nicholas Freitag McPhee. *A field guide to genetic programming*. (With contributions by J. R. Koza). Published via <http://lulu.com> e freely available at <http://www.gp-field-guide.org.uk>, 2008.
- [49] Andy Pryke, Sanaz Mostaghim e Alireza Nazemi. “Heatmap visualization of population based multi objective algorithms”. Em: *International Conference on Evolutionary Multi-Criterion Optimization*. Springer. 2007, pp. 361–375.
- [50] K. M. Ragsdell e D. T. Phillips. “Optimal design of a class of welded structures using geometric programming”. Em: *Journal of Engineering for Industry* 98.3 (1976), pp. 1021–1025.
- [51] Ramón Quiza Sardiñas, Marcelino Rivas Santana e Eleno Alfonso Brindis. “Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes”. Em: *Engineering Applications of Artificial Intelligence* 19.2 (2006), pp. 127–133.
- [52] J David Schaffer. *Some experiments in machine learning using vector evaluated genetic algorithms*. Rel. téc. Vanderbilt Univ., Nashville, TN (USA), 1985.

- [53] Joseph E Shigley, Charles R Mischke e Richard G Budynas. *Mechanical engineering design*. McGraw-Hill, 2004.
- [54] Nidamarthi Srinivas e Kalyanmoy Deb. “Multiobjective optimization using nondominated sorting in genetic algorithms”. Em: *Evolutionary Computation* 2.3 (1994), pp. 221–248.
- [55] Ruiyi Su et al. “Multi-objective topology and sizing optimization of truss structures based on adaptive multi-island search strategy”. Em: *Structural and Multidisciplinary Optimization* 43.2 (2011), pp. 275–286.
- [56] Kazuyuki Sugimura et al. “Kriging-model-based multi-objective robust optimization and trade-off-rule mining using association rule with aspiration vector”. Em: *2009 IEEE Congress on Evolutionary Computation*. IEEE. 2009, pp. 522–529.
- [57] Heidi A Taboada e David W Coit. “Data mining techniques to facilitate the analysis of the Pareto-optimal set for multiple objective problems”. Em: *IIE Annual Conference. Proceedings*. Institute of Industrial Engineers-Publisher. 2006, p. 1.
- [58] Stephen P Timoshenko e James M Gere. *Theory of elastic stability*. 1961. Tata McGraw-Hill Education, 1961.
- [59] Tamara Ulrich. “Pareto-Set Analysis: Biobjective Clustering in Decision and Objective Spaces”. Em: *Journal of Multi-Criteria Decision Analysis* 20.5-6 (2013), pp. 217–234.
- [60] Tamara Ulrich, Dimo Brockhoff e Eckart Zitzler. “Pattern identification in Pareto-set approximations”. Em: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*. ACM. 2008, pp. 737–744.
- [61] D. E. C. Vargas. “Um Algoritmo de Evolução Diferencial com Penalização Adaptativa para Otimização Estrutural Multiobjetivo”. Tese de doutorado. Universidade Federal de Juiz de Fora, 2015.
- [62] D. E. C. Vargas et al. “Um algoritmo baseado em evolução diferencial para problemas de otimização estrutural multiobjetivo com restrições”. Em: *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería* 32.2 (2016), pp. 91–99.

- [63] Lofti Zadeh. “Optimality and non-scalar-valued performance criteria”. Em: *IEEE Transactions on Automatic Control* 8.1 (1963), pp. 59–60.
- [64] Eckart Zitzler e Lothar Thiele. “Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach”. Em: *IEEE Transactions on Evolutionary Computation* 3.4 (1999), pp. 257–271.