

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Gabriel Augusto Marson

**Análise Comparativa dos Algoritmos Genéticos
Many-Objective em problemas de otimização
discreta**

Uberlândia, Brasil

2017

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Gabriel Augusto Marson

**Análise Comparativa dos Algoritmos Genéticos
Many-Objective em problemas de otimização discreta**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau
de Bacharel em Ciência da Computação.

Orientador: Gina Maira Barbosa de Oliveira

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2017

Resumo

Esta investigação tem como objetivo mostrar o desempenho de alguns algoritmos genéticos multi-objetivos, em especial aqueles conhecidos como *many-objective*, em um problema discreto da computação, o problema da mochila.

Além disso, será apresentado um estudo da evolução dos algoritmos genéticos que é uma abordagem não determinística para resolver problemas, passando pelos multi-objetivos, que resolvem problemas de 2 ou 3 funções objetivo, e finalizando nos *many-objectives*, que são capazes de resolver problemas com várias funções objetivo, ou seja, a partir de 4. É feita uma avaliação comparativa de desempenho dos algoritmos NSGA-II(*Nondominated Sorting Genetic Algorithm II*) e SPEA2(*Strength Pareto Evolutionary Algorithm 2*), considerados multi-objetivos e dos mais recentes que são considerados *many-objectives* como o MOEA/D(*Multi-objective Evolutionary Algorithm Based on Decomposition*), AEMMT(*Multiobjective Evolutionary Algorithm With Many Tables*) e AEMMD(Algoritmo Evolutivo Multiobjetivo com Múltiplas Dominâncias). De acordo o tempo de execução e outras métricas de desempenho, será feita uma avaliação comparativa entre esses algoritmos para lidar com quatro ou mais critérios de otimização.

Este trabalho, portanto comprova uma maior eficácia dos algoritmos *many-objective* em detrimento dos multi-objetivos para uma quantidade de objetivos mais elevada. Ele também mostra que, a medida que a complexidade do problema aumenta, os *many-objectives* apresentam um maior desempenho do que o SPEA2 e NSGA-II.

Palavras-chave: algoritmos genéticos, otimização discreta, otimização multi-objetivos, estratégias *many-objective*.

Lista de ilustrações

Figura 1 – Subpopulações de um problema com quatro objetivos (BRASIL; DEL-BEM; SILVA, 2013)	16
Figura 2 – Comparativo Detalhado de Erro	32
Figura 3 – Comparativo Detalhado de <i>Pareto Subset</i>	33
Figura 4 – Comparativo Detalhado de <i>Generational Distance</i>	34
Figura 5 – Comparativo Detalhado de <i>Generational Distance</i> sem o AEMMD . . .	35
Figura 6 – Comparativo de Erro	36
Figura 7 – Comparativo de <i>Pareto Subset</i>	37
Figura 8 – Comparativo de <i>Generational Distance</i>	38
Figura 9 – Comparativo de <i>Generational Distance</i> sem o AEMMD	38
Figura 10 – Comparativo de <i>Tempo de Execução</i>	39

Lista de tabelas

Tabela 1 – Cardinalidade das Fronteiras de Pareto	23
Tabela 2 – 2 Objetivos - 30 Itens	23
Tabela 3 – 2 Objetivos - 50 Itens	24
Tabela 4 – 2 Objetivos - 100 Itens	24
Tabela 5 – 3 Objetivos - 30 Itens	25
Tabela 6 – 3 Objetivos - 50 Itens	25
Tabela 7 – 3 Objetivos - 100 Itens	26
Tabela 8 – 4 Objetivos - 30 Itens	26
Tabela 9 – 4 Objetivos - 50 Itens	27
Tabela 10 – 4 Objetivos - 100 Itens	27
Tabela 11 – 5 Objetivos - 30 Itens	28
Tabela 12 – 5 Objetivos - 50 Itens	28
Tabela 13 – 5 Objetivos - 100 Itens	29
Tabela 14 – 6 Objetivos, 30 Itens	30
Tabela 15 – 6 Objetivos - 50 Itens	30
Tabela 16 – 6 Objetivos - 100 Itens	31

Lista de abreviaturas e siglas

NSGA-II	<i>Nondominated Sorting Genetic Algorithm II</i>
SPEA2	<i>Strength Pareto Evolutionary Algorithm 2</i>
AGMO	Algoritmo Genético Multi-objetivo
MEANT	<i>Multiobjective Evolutionary Algorithm With Many Tables</i>
AEMMD	Algoritmo Evolutivo Multiobjetivo com Múltiplas Dominâncias
AEMO	Algoritmo Evolutivo Multiobjetivo
MOEA/D	<i>Multi-objective Evolutionary Algorithm Based on Decomposition</i>

Sumário

1	INTRODUÇÃO	8
1.1	Objetivos	9
1.1.1	Objetivos gerais	9
1.1.2	Objetivos específicos	9
1.1.3	Justificativas	9
1.1.4	Metodologia	10
1.1.5	Estrutura do texto	10
2	REFERENCIAL TEÓRICO	11
2.1	Algoritmos Genéticos	11
2.2	Fronteira de Pareto	12
2.3	Algoritmos Evolutivos Multi-Objetivos	12
2.3.1	<i>Non-dominated Sorting Genetic Algorithm II (NSGA-II)</i>	12
2.3.2	<i>Strength Pareto Evolutionary Algorithm 2 (SPEA2)</i>	14
2.3.2.1	Cálculo de Aptidão	14
2.3.2.2	Algoritmo de Truncamento (Environmental Selection)	15
2.4	Algoritmos Evolutivos <i>Many-Objective</i>	15
2.4.1	<i>Multiobjective Evolutionary Algorithm With Many Tables (AEMMT)</i>	15
2.4.2	Algoritmo Evolutivo Multiobjetivo com Múltiplas Dominâncias (AEMMD)	17
2.4.3	Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D)	17
2.5	Outros Algoritmos	18
2.5.1	<i>Bi-goal evolution for many-objective optimization problems - (BiGE)</i>	18
2.5.2	<i>HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization (HypE)</i>	19
2.6	Métricas de desempenho dos AEMOs	19
2.7	Problema da Mochila (<i>Knapsack Problem</i>)	20
2.8	Trabalhos Correlatos	20
2.8.1	Algoritmos Evolutivos Many-Objectives Aplicados ao Problema de Roteamento Multicast com Qualidade de Serviço	20
2.8.2	A Comparative Analysis of MOEAs Considering Two Discrete Optimization Problems	21
3	EXPERIMENTOS E RESULTADOS	22
3.1	Especificação do método de avaliação	22
3.1.1	Parâmetros	22
3.2	Fronteiras de Pareto	23

3.3	Experimento 1: Problema da Mochila com Dois Objetivos	23
3.3.1	30 Itens	23
3.3.2	50 Itens	24
3.3.3	100 Itens	24
3.4	Experimento 2: Problema da Mochila com Três Objetivos	25
3.4.1	30 Itens	25
3.4.2	50 Itens	25
3.4.3	100 Itens	26
3.5	Experimento 3: Problema da Mochila com Quatro Objetivos	26
3.5.1	30 Itens	26
3.5.2	50 Itens	27
3.5.3	100 Itens	27
3.6	Experimento 4: Problema da Mochila com Cinco Objetivos	28
3.6.1	30 Itens	28
3.6.2	50 Itens	28
3.6.3	100 Itens	29
3.7	Experimento 5: Problema da Mochila com Seis Objetivos	30
3.7.1	30 Itens	30
3.7.2	50 Itens	30
3.7.3	100 Itens	31
3.8	Análise comparativa por métrica	31
3.8.1	Erro	32
3.8.2	<i>Pareto Subset</i>	33
3.8.3	<i>Generational Distance</i>	34
3.9	Análise comparativa da média das métricas	36
3.9.1	Erro Médio	36
3.9.2	<i>Pareto Subset</i> Médio	37
3.9.3	<i>Generational Distance</i> Médio	38
3.10	Tempo de execução	39
4	CONCLUSÃO	41
4.1	Dificuldades Encontradas	42
4.2	Trabalhos Futuros	43
	REFERÊNCIAS	44

1 Introdução

Algoritmos genéticos (AGs) apresentam uma abordagem de busca baseada na teoria da evolução de Darwin, na qual executam uma sequência de passos para atingir um determinado objetivo. Nesses algoritmos, indivíduos mais aptos passarão suas características aos seus descendentes, similar ao observado na natureza.

Apesar de ter sido constatado o sucesso dos AGs para certos tipos de problemas, alguns outros permanecem como desafio. Os problemas de otimização discreta com múltiplos objetivos são mais desafiadores pois requerem a maximização ou minimização de mais de uma função objetivo.

Consideremos, por exemplo, o problema do Roteamento Multicast (PRM). Como foi dito em (BUENO, 2010), esse problema pode ser modelado como um grafo direcionado onde os vértices representam os hosts e as arestas representam os enlaces de comunicação. Dessa forma, dado o grafo G que representa a rede e um fluxo ϕ , deve-se calcular árvores enraizadas de G para carregar ϕ em G , partindo-se de um vértice que representa o início de uma conexão ou envio de pacote e passando por um subconjunto de vértices de G que tenha todos os nós folha como destino. Diversos atributos podem influenciar o desempenho de uma rede, tais como: custo, delay, capacidade e tráfego atual.

Um AG simples apenas consegue administrar um critério por vez. Variações conhecidas por Algoritmos Genéticos Multiobjetivos (AGMOs) foram propostas ((SRINIVAS; DEB, 1994), (ZITZLER; THIELE, 1998), (DEB et al., 2002), (ZITZLER et al., 2001)), na qual o processo de avaliação dos indivíduos leva em consideração diferentes objetivos ou critérios. Os AGMOs constituem uma abordagem mais adequada para resolver problemas com múltiplos objetivos categorizando possíveis melhores respostas em virtude dos critérios avaliados.

A fim de considerar todos os objetivos no cálculo de aptidão, os AGMO's utilizam o conceito de fronteira de Pareto (VELDHUIZEN; LAMONT, 1998) que estabelece uma região do espaço de busca na qual estarão os melhores indivíduos de acordo com várias funções objetivo. O critério para a classificação de qual indivíduo permanece em qual fronteira é a dominância. Dados dois indivíduos A e B , pode-se dizer que A domina B se, e somente se, A não é pior que B em nenhuma das funções objetivos e A é melhor que B em pelo menos uma delas.

O NSGAII (DEB et al., 2002) é um dos AGMOs mais conhecidos. Nesse modelo é realizado agrupamento por fronteiras de dominância, ou *rank*. Cada fronteira armazena as soluções da população que são não dominadas entre si e a primeira fronteira armazena as soluções que dominam todas as outras fronteiras. O método utiliza também a métrica

crowding distance, a qual consiste em priorizar os resultados mais diferentes como critério de desempate na escolha do indivíduo caso estejam na mesma fronteira de dominância e, melhorando a variabilidade genética da população. O objetivo dessa métrica é que os indivíduos na primeira fronteira da população final sejam mais diversificados possível e próximos das soluções da Fronteira de Pareto associada ao problema.

Este trabalho realiza uma comparação entre vários algoritmos genéticos multi-objetivo, em especial aqueles conhecidos por AGMOs *many-objective*. Busca-se saber qual o AGMO mais indicado para determinada situação, qual tem o menor custo computacional e qual possui maior capacidade de conceder respostas corretas e diversificadas.

1.1 Objetivos

1.1.1 Objetivos gerais

Em virtude de bons resultados gerados pelos AGMOs em problemas com múltiplos objetivos, foram investigados diferentes métodos evolutivos multi e *many-objective*, tais como, NSGA-II (DEB et al., 2002), SPEA2 (ZITZLER et al., 2001), MOEA/D (ZHANG; LI, 2007), AEMMD (LAFETÁ, 2016) e AEMMT (BRASIL; DELBEM; SILVA, 2013) aplicados ao problema da Mochila. Enquanto o NSGAII e o SPEA2 são dois dos principais métodos evolutivos multi-objetivos elitistas, o MOEA/D, o AEMMT, e o AEMMD representam a geração mais recentes dos AGMOs, voltada a solução de problemas com muitos objetivos (*many-objective*). Dessa forma, esse trabalho busca estabelecer qual algoritmo é mais indicado dado um determinado conjunto de objetivos.

1.1.2 Objetivos específicos

- Implementar em linguagem JAVA os algoritmos SPEA2, NSGA-II, MOEA/D, AEMMT e AEMMD;
- Esse trabalho, portanto, tem como objetivo realizar uma análise comparativa dos algoritmos SPEA2, NSGAII, MOEA/D, AEMMD e AEMMT aplicados no problema da mochila.

1.1.3 Justificativas

Existem muitos problemas de otimização discreta multi-objetivo ainda não resolvidos de forma eficiente. O estudo desses problemas, assim como dos algoritmos que os resolvem, constituem prática importante para embasar novas descobertas com o intuito de otimizar as soluções já existentes ou de criar novas.

1.1.4 Metodologia

Será realizada uma pesquisa descritiva de caráter explicativo com o intuito de estabelecer uma comparação entre os algoritmos SPEA2, NSGAI, MOEA/D, AEMMT e AEMMD. Pretende-se, também, explicar em quais circunstâncias é indicado o uso de cada um.

Os critérios de comparação são as métricas de erro, *pareto subset* e *generational distance*. Também será considerado o tempo de execução de cada algoritmo.

Espera-se que o desempenho dos algoritmos SPEA2 e NSGAI diminuam consideravelmente a medida que uma maior quantidade de funções é requerida pelo problema, portanto, torna-se necessário o estudo dos algoritmos genéticos *many-objective*.

Após isso, será feita a análise e interpretação dos resultados gerados pela execução desses algoritmos aplicados ao problema da mochila.

1.1.5 Estrutura do texto

Este trabalho possui a seguinte organização textual:

O capítulo 2 descreve o referencial teórico utilizado nessa pesquisa. Esse capítulo consiste em um detalhamento dos algoritmos utilizados e do problema da mochila.

O capítulo 3 apresenta os experimentos e resultados obtidos utilizando as métricas propostas.

No capítulo 4 são apresentadas as conclusões, trabalhos futuros e dificuldades encontradas.

2 Referencial Teórico

Nesta seção, serão abordados conceitos necessários para a compreensão deste trabalho, como o funcionamento dos algoritmos genéticos multi-objetivos.

2.1 Algoritmos Genéticos

Publicada em 1859, a obra **Na Origem das Espécies – Sob o Conhecimento da Seleção Natural**, de Charles Darwin, foi alvo de críticas pois explicitava que os seres de um ambiente mudam de uma determinada geração para outra e são selecionados de acordo com essas mudanças. A partir disso, Darwin escreveu que todos os seres vivos possuem um ancestral comum, contrastando com a visão criacionista da época.

A teoria de Darwin constituiu significativo embasamento para que John Holland formulasse os algoritmos genéticos. Físico formado pelo MIT (*Massachusetts Institute of Technology*), Holland elaborou um algoritmo não determinístico com foco em resolução de problemas de custo muito alto.

O AG consiste em dar uma interpretação para um conjunto de dados e evoluí-los para uma solução ótima. Todo o conjunto de indivíduos (dados) é uma população e cada indivíduo representa uma solução para o problema. O algoritmo avalia cada membro usando uma métrica de aptidão, atribuindo uma nota para cada indivíduo que simboliza as chances dele transmitir suas características para a próxima geração.

O algoritmo é iniciado através de uma geração inicial de indivíduos que é feita geralmente de forma aleatória. Depois da geração da população inicial, os indivíduos são avaliados e é feito um processo de seleção deles para a etapa de reprodução. Alguns métodos podem ser usados nesta etapa de seleção. O processo de troca de material genético chamado *crossover* retorna um ou mais filhos sendo que, geralmente, existe uma pequena chance de ocorrer uma mutação no material genético dos filhos, em virtude de aumentar a variabilidade na população.

Após o cruzamento, os filhos resultantes são reinseridos na população. Com o intuito de manter o custo computacional não crescente e de eliminar indivíduos fracos, utilizam-se métodos de reinserção para diminuir a população. Esses métodos avaliam o desempenho de cada indivíduo na função objetivo e mantêm os que obtiveram melhores resultados.

Este ciclo se repete por um número determinado de gerações ou até quando todos os indivíduos possuírem o mesmo material genético (é improvável que a mutação nas próximas gerações produza um indivíduo mais apto).

Os primeiros algoritmos genéticos investigados na literatura trabalhavam com uma única métrica na avaliação de um indivíduo. Entretanto, alguns problemas possuem múltiplos objetivos o que torna a avaliação escalar ineficiente. Dessa forma, métodos evolutivos capazes de tratar diferentes objetivos foram propostos dando origem aos algoritmos genéticos multi-objetivos (AGMOs). Vários AGMOs trabalham com o conceito de Fronteira de Pareto.

2.2 Fronteira de Pareto

A Fronteira de Pareto (VELDHUIZEN; LAMONT, 1998) consiste em uma curva n -dimensional no espaço de soluções do problema que determina todas as soluções não dominadas existentes. O objetivo de um algoritmo genético com mais de um objetivo é aproximar essa curva.

1. para qualquer f pertencente a F , $f(x) \leq f(y)$
2. existe f pertencente a F tal que $f(x) < f(y)$

onde,

F é o conjunto de funções objetivo. O problema deseja minimizar os valores dessas funções .

Esse tipo de classificação pode ser melhor compreendida se analisarmos um problema simples como a compra de um carro. Vários critérios podem ser utilizados para comprá-lo como preço e desempenho. Na fronteira de Pareto, estariam, portanto, os indivíduos não dominados em termos de preço e desempenho, ou seja, os que não tiverem essas características piores que algum outro. Estas duas métricas são normalmente conflitantes, quanto melhor o preço, pior o desempenho. Carros com bom preço e bom desempenho estarão mais próximos da fronteira de Pareto. Portanto, não existe uma única resposta para esse tipo de problema sendo que na fronteira de Pareto estarão as melhores respostas possíveis considerando preço e desempenho.

2.3 Algoritmos Evolutivos Multi-Objetivos

2.3.1 *Non-dominated Sorting Genetic Algorithm II (NSGA-II)*

O *Non-dominated Sorting Genetic Algorithm* (DEB et al., 2002) é um algoritmo baseado em uma ordenação elitista por dominância. Tem como objetivo classificar os indivíduos de um conjunto M por fronteiras F sendo que na fronteira F_1 estariam os melhores indivíduos por critérios de dominância de todo o conjunto M . Na fronteira

F_2 , estariam todos os indivíduos não dominados de $M - F_1$ e assim sucessivamente. A classificação dos indivíduos ocorre utilizando-se o conceito de dominância que pode ser explicado usando os seguintes critérios.

1. Nd_i o número de soluções que dominam a solução i .
2. U_i o conjunto de soluções que são dominadas por i . Indivíduos com menor Nd_i possível ($Nd_i = 0$) estão na primeira fronteira. As demais fronteiras são estabelecidas pelos indivíduos com Nd_i subsequente.

O NSGA-II trabalha com duas populações P_t e Q_t sendo que P_t é a população inicial da geração corrente e Q_t são os filhos de P_t resultantes de um processo de cruzamento após a seleção. Ao fim de cada geração t , essas duas populações serão concatenadas em uma só e ocorrerá um processo de seleção de todos os indivíduos para formar uma nova população P_{t+1} que será a inicial do próximo ciclo. Serão escolhidos, portanto, os indivíduos que estão nas primeiras fronteiras até o número total de P_{t+1} ser preenchido. Como critério de desempate para o caso de a última fronteira ser maior do que o restante de elementos faltantes em P_{t+1} , utiliza-se o critério de desempate *Crowding Distance* em que são priorizados os indivíduos com maior variabilidade genética.

O cálculo do *Crowding Distance* é feito de forma tal que serão selecionados os indivíduos que estão mais distantes um dos outros na fronteira. O processo é realizado de acordo com cada critério de otimização imposto no algoritmo, ou seja, para cada função objetivo, temos que:

$$Crowding_i = Crowding_i + \frac{f_{i+1} - f_{i-1}}{f_{max} - f_{min}}$$

Sendo que :

- $Crowding_i$: distância do indivíduo i ;
- f_{i+1} : função objetivo correspondente a um critério de otimização do próximo indivíduo;
- f_{i-1} : função objetivo correspondente a um critério de otimização do indivíduo anterior;
- f_{max} : maior valor da função objetivo correspondente a um critério de otimização aplicada em uma fronteira;
- f_{min} : maior valor da função objetivo correspondente a um critério de otimização aplicada em uma fronteira

Tendo em vista que um problema possua N critérios de otimização, cada indivíduo possui N resultados $Crowding_i$ e o valor de $Crowding$ final é o somatório de todos esses resultados.

A aptidão de cada solução é determinada usando-se os seguintes critérios.

1. *Rank*(ou fronteira) em que o indivíduo está. Essa nota sempre será utilizada.
2. *Crowding Distance* como desempate para indivíduos na mesma fronteira.

Ao fim desse processo espera-se que vários, ou todos, indivíduos estejam na primeira fronteira com variabilidade elevada. A solução mais desejada é a primeira fronteira tornar-se a fronteira de Pareto, ou seja, todas as soluções ótimas possíveis foram encontradas.

2.3.2 *Strength Pareto Evolutionary Algorithm 2 (SPEA2)*

O *Strength Pareto Evolutionary Algorithm 2* (ZITZLER et al., 2001) é um algoritmo que, assim como o NSGA-II, trabalha com duas populações P e $ARCHIVE$ sendo que P é a população inicial e em $ARCHIVE$ serão apenas armazenadas, as soluções não dominadas encontradas pelo algoritmo. Serão denotados por P_t e $ARCHIVE_t$ as populações para a geração $t = 1, 2, \dots, N_{iter}$.

2.3.2.1 Cálculo de Aptidão

Inicia-se estabelecendo uma população inicial aleatória para P_1 e $ARCHIVE_1 = \emptyset$ classificando todos os indivíduos de $R_t = P_t \cup ARCHIVE_t$. Um dos critérios que irá compor a função de aptidão (*fitness*) é o *strength* de cada indivíduo que pode ser estabelecido por:

$$strength_i = \{j, j \in R_t, | i \prec j\}$$

A interpretação para o valor *strength* é a quantidade de elementos em R_t que são dominados por i , como foi dito em (TICONA; DELBÉM, 2008). Calcula-se, também o valor de *raw fitness* denotado na equação como:

$$raw_i = \sum_{j \in R_t, j \leq i} strength(j)$$

Sendo assim, raw_i é o somatório dos *strengths_j* tal que $j \in R_t$ e $j \prec i$. Quanto menor for o valor raw_i de um indivíduo, menos dominado ele será, sendo que $raw_i = 0$ é o melhor valor de R para um indivíduo.

Pode ocorrer que apenas com a atribuição de R servindo como *fitness* final seja insuficiente para um bom desempenho do algoritmo pois indivíduos iguais teriam a mesma nota. Dessa maneira, criou-se a métrica de densidade que concederia uma nota menor para indivíduos em áreas muito concentradas. É importante destacar que essa medida confere ao algoritmo mais diversidade de soluções ao final do processo, pois atribui maior chance de indivíduos em áreas menos densas de serem selecionados. A fórmula para se calcular a densidade de indivíduos é a seguinte:

$$D(i) = \frac{1}{\sigma_k^i + 2}$$

Sendo que σ_k^i é a distância de um indivíduo para o k -ésimo vizinho mais próximo e k é definido como $\sqrt[2]{N + \bar{N}}$ sendo que N é o tamanho de P e $\bar{N}$ é o tamanho de $ARCHIVE$.

Dessa forma, compõem o cálculo de aptidão a *raw fitness* e a densidade.

$$F(i) = R(i) + D(i)$$

2.3.2.2 Algoritmo de Truncamento (Environmental Selection)

Ao final do cálculo de aptidão, os elementos não dominados de $P \cup ARCHIVE$ vão para $ARCHIVE$. Se $ARCHIVE_{size} < \bar{N}$ então preenche-se o restante do arquivo com os indivíduos menos dominados de P . Caso $ARCHIVE_{size} > \bar{N}$ então realiza-se um processo de truncamento que irá remover os indivíduos do arquivo até que $ARCHIVE_{size} = \bar{N}$.

Após o processo de seleção, inicia-se o *crossover* com possibilidade de mutação seguido pela reinserção dos filhos na população de próxima geração.

2.4 Algoritmos Evolutivos *Many-Objective*

2.4.1 *Multiobjective Evolutionary Algorithm With Many Tables (AEMMT)*

O AEMMT (Algoritmo Evolutivo Multiobjetivo com Muitas Tabelas) consiste em um algoritmo *many-objective* proposto inicialmente para a predição de estruturas de proteínas (BRASIL; DELBEM; SILVA, 2013). Nesse tipo de ambiente, existem quatro funções objetivo a serem minimizadas e os algoritmos multi-objetivo testados mostraram-se ineficazes para a resolução desse problema.

O estudo propõe a criação de um algoritmo baseado em tabelas, onde cada tabela corresponderia a uma subpopulação, criada de acordo com os objetivos do problemas. É importante destacar que, durante o processo de geração de subpopulações, serão considerados subconjuntos das funções objetivos.

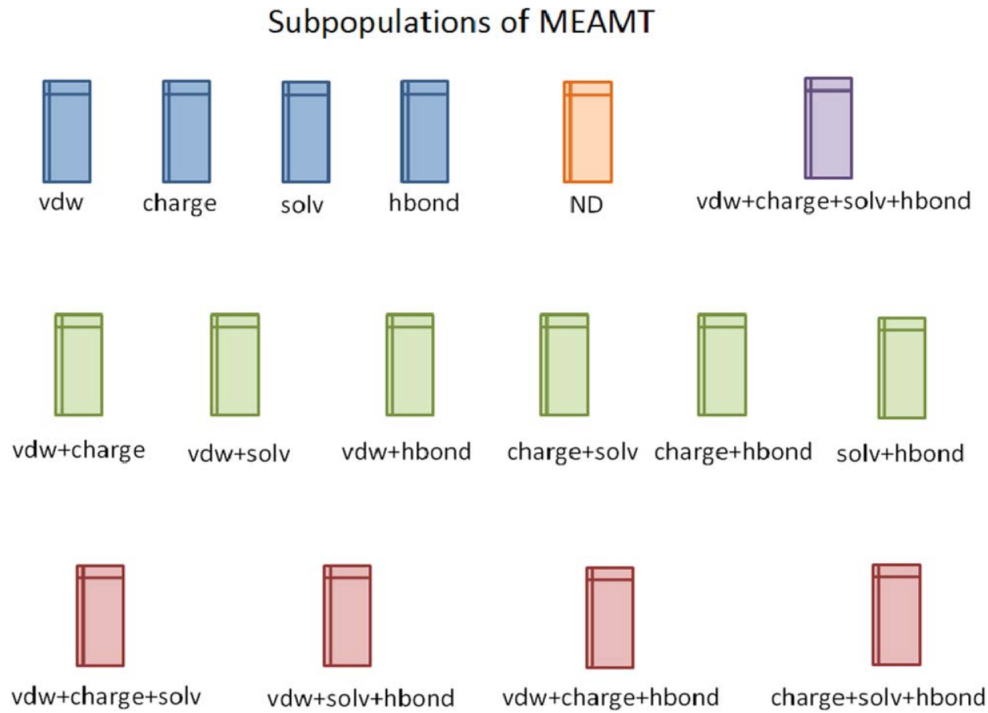


Figura 1 – Subpopulações de um problema com quatro objetivos (BRASIL; DELBEM; SILVA, 2013)

A Figura 1, retirada de (BRASIL; DELBEM; SILVA, 2013), mostra populações de acordo com um objetivo(em azul) e depois, conjuntos de indivíduos formados pelo critério de dois objetivos(em verde), três(em vermelho) e quatro(em roxo). A população em laranja mostra os indivíduos não dominados dessas populações.

Cada tabela tem um *score* de contribuição, que representa quantos indivíduos gerados por ela foram inseridos em outros conjuntos. Uma vez que o AEMMT opera com um torneio de tabelas para decidir qual irá contribuir com o próximo indivíduo, um conjunto com maior contribuição possui mais chance de ser selecionado.

À exceção da tabela de indivíduos não dominados (*ND*, na figura acima), a métrica para classificação delas é a média ponderada das funções objetivo avaliadas na tabela. Consequentemente, para o indivíduo ser inserido em alguma tabela, basta que ele tenha uma média ponderada melhor do que o pior indivíduo da tabela. No caso do conjunto *ND*, o indivíduo será inserido se não for dominado por nenhum outro indivíduo.

Esse algoritmo difere dos demais no que diz respeito ao número de gerações pois é necessária uma quantidade mais elevada para que se obtenha bons resultados, tendo em vista que apenas um indivíduo é produzido a cada geração. Portanto, como critério de comparação de desempenho entre algoritmos *many-objective*, não é coerente usar o número de gerações e sim a quantidade de comparações feitas durante todo o processo.

Todas as tabelas possuem um tamanho fixo definido, e é importante ressaltar

que, por conta disto, os indivíduos novos não dominados que seriam inseridos na tabela de não dominância são desconsiderados. No entanto, existe uma variação do AEMMT que consiste não limitar o tamanho da tabela de dominância. Seriam limitadas apenas as tabelas que têm como base de seleção a média ponderada. Essa alteração foi feita pensando na possibilidade de a fronteira de Pareto ser maior do que o tamanho estipulado das tabelas, e dessa forma, indivíduos que pertenceriam ao Pareto poderiam ser descartados.

2.4.2 Algoritmo Evolutivo Multiobjetivo com Múltiplas Dominâncias (*AEMMD*)

Proposto por (LAFETÁ, 2016), o AEMMD é muito semelhante ao AEMMT e somente difere do seu precursor em alguns aspectos.

1. A métrica de média ponderada presente nas tabelas do AEMMT foi substituída pela métrica de dominância. Dessa forma, os conjuntos de objetivos de todas as tabelas serão de acordo com a dominância. A inserção de um novo indivíduo em uma tabela é feita somente se não for dominado por nenhum outro indivíduo de acordo com os objetivos descritos nela. Além disso, no AEMMD não existe tamanho definido para as tabelas.
2. As tabelas recebem pontos por convergência, ou seja, sempre que uma tabela receber um novo indivíduo, terá sua pontuação acrescida em um. Assim, na hora de selecionar as tabelas para o cruzamento, ao invés de se utilizar os pontos de contribuição, como no AEMMT, aplicam-se os pontos de convergência.
3. O uso da quantidade de tabelas é reduzido no AEMMD, uma vez que não são necessárias as tabelas de dominância por um objetivo, tendo em vista que a métrica de dominância é aplicada tão-somente com dois ou mais objetivos. Assim, a única tabela que está presente em ambos algoritmos é aquela baseada em dominância, incluindo todos os objetivos presentes.

Para o problema do Roteamento Multicast (PRM), o qual foi objeto de estudo do artigo em que foi proposto o AEMMD, observou-se que este algoritmo obteve desempenho superior aos outros em todas as variações estudadas do problema.

2.4.3 Multiobjective Evolutionary Algorithm Based on Decomposition (*MOEA/D*)

O MOEA/D é um algoritmo que tem como premissa principal dividir um problema em sub-problemas, que seriam os indivíduos, com um único objetivo. Cada indivíduo possui um vetor de pesos do tamanho da quantidade de objetivos do problema sendo que a soma dos pesos não pode ultrapassar o valor 1.

Uma vizinhança é composta pelos N (8, de acordo com o artigo original) indivíduos mais próximos do indivíduo atual. A proximidade é estabelecida pela distância euclidiana considerando o vetor de pesos de cada indivíduo.

Cada indivíduo da população otimiza uma função objetivo tendo como critério funções escalares relacionadas ao vetor de pesos. Foram citadas no artigo original as funções de *Soma ponderada*, *Tchebycheff* e *Penalty Based Boundary Intersection(PBI)*. Para a função de *Soma Ponderada*, dado um vetor de pesos $\lambda = (\lambda_1 \dots \lambda_n)$ onde n é a quantidade de funções objetivo, têm-se que a solução ótima para essa função escalar é:

$$solution = \sum_{i=1}^n \lambda_i * f_i$$

na qual f é a função objetivo. Dessa forma, o MOEA-D se comporta como um algoritmo buscando respostas priorizadas de acordo com o esquema de pesos do indivíduo.

Após a determinação das vizinhanças é atribuído a cada indivíduo uma métrica *fitness*. Esse atributo muda ao decorrer do algoritmo conforme são encontrados melhores indivíduos. A cada geração, são selecionados dois indivíduos de cada vizinhança para o *crossover*. Esse processo não relaciona vizinhanças distintas, portanto, o cruzamento é feito com indivíduos da mesma vizinhança, sendo que a inserção ocorre de forma análoga.

Todo o indivíduo filho, independentemente de ser adicionado à vizinhança dos pais que o geraram, é inserido em um arquivo o qual mantém controle dos indivíduos não dominados sem considerar nenhum esquema de pesos. O indivíduo permanece nesse arquivo se não existe nenhum outro que o domine. O algoritmo tem como resposta esse arquivo.

2.5 Outros Algoritmos

2.5.1 *Bi-goal evolution for many-objective optimization problems - (BiGE)*

O artigo(LI; YANG; LIU, 2015) propõe uma abordagem diferente dos *many-objective* para problemas com mais de três objetivos sob a premissa de que quanto maior for a quantidade de objetivos, o conflito de proximidade e diversidade das soluções é agravado além da métrica de dominância com a fronteira de Pareto perder gradativamente sua eficácia.

A proposta do algoritmo é converter qualquer problema *many-objective* em problemas *Bi-Goal* que constituem uma abordagem em duas frentes, proximidade e *crowding degree*. O algoritmo foi aplicado ao problema da mochila e demonstrou desempenho similar aos algoritmos genéticos *many-objective* estado da arte que foram testados(MOEA/D(ZHANG; LI, 2007), NSGA-III(YUAN; XU; WANG, 2014), HypE(BADER;

ZITZLER, 2011), FD-NSGA-II(HE; YEN; ZHANG, 2014) e AGE-II(BRINGMANN et al., 2011)).

2.5.2 HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization (HypE)

O algoritmo utiliza uma métrica de hipervolume no processo de *environmental selection*. Apesar de o cálculo de hipervolume ter um custo muito alto, o artigo explicita que, segundo estudos recentes, esse tipo de métrica é o único indicador que otimiza a dominância no contexto de fronteira de Pareto.

O artigo conclui que o HypE é altamente competitivo com os outros algoritmos que utilizam hipervolume como métrica de cruzamento e seleção sendo caracterizado como um excelente algoritmo para resolução de problemas multi-objetivo e um algoritmo satisfatório para *many-objective*.

2.6 Métricas de desempenho dos AEMOs

Sendo P^* uma estimativa do Pareto Ótimo obtida com a execução de todos AEMOs estudados e P o Pareto obtido por um AEMO (fronteira de não dominância da população final), têm-se:

- *Taxa de Erro (er)*: A taxa de erro é dada pela porcentagem de soluções encontradas em P que não fazem parte do Pareto real.

$$er = \frac{\sum_{i=1}^{|P|} e_i}{|P|}$$

Sendo que $e_i = 1$ se a solução $i \in P$ não fizer parte do Pareto real.

- *Generational Distance (gd)*: métrica que mede a convergência de P em P^* dado pela equação:

$$gd = \frac{\sqrt{\sum_{i=1}^{|P|} d_i^2}}{|P|}$$

Sendo que d_i^2 é dada pela menor distância euclidiana de cada indivíduo $i \in P$ com relação aos indivíduos $j \in P^*$

- *Pareto Subset (ps)*: é a quantidade absoluta de elementos de P que não são dominados por nenhum outro elemento em P^* .

2.7 Problema da Mochila (*Knapsack Problem*)

O problema de mochila consiste em uma otimização combinatória onde se deve colocar itens na mochila de acordo com seus atributos sem ultrapassar sua capacidade. Como exemplo têm-se o problema da mochila bidimensional em que a quantidade de atributos dos itens são duas (valor, peso). O objetivo, portanto, é obter o maior valor de objetos possível dado o peso que a mochila suporta como limitação.

Obter uma resposta desse problema por meio do método de força-bruta (testa-se todas as combinações possíveis) é inviável em decorrência do crescimento exponencial que apresenta. Existem outras abordagens para a resolução desse tipo de problema e que funcionam de forma satisfatória como, por exemplo, programação dinâmica e os algoritmos genéticos de um único objetivo.

No entanto, esses métodos citados não apresentam um bom desempenho quando o problema da mochila torna-se multidimensional, pois, dessa forma, deve-se avaliar, além do valor dos objetos colocados na mochila, outras características. Assim, o problema passa a ter vários objetivos.

Essa pesquisa pretende avaliar o desempenho de algoritmos evolutivos multi-objetivos como SPEA2 e NSGA-II e *many-objectives* como AEMMT, AEMMD e MOEA/D em problemas desse tipo para formulações com até 6 objetivos. Sabe-se pela literatura que, conforme a quantidade de objetivos aumenta, torna-se necessário o uso de algoritmos *many-objective*.

2.8 Trabalhos Correlatos

2.8.1 Algoritmos Evolutivos Many-Objectives Aplicados ao Problema de Roteamento Multicast com Qualidade de Serviço

O artigo(LAFETÁ, 2016) demonstra a evolução dos algoritmos genéticos, passando pelos multi-objetivos até os *many-objectives*. É justificado o uso dos algoritmos genéticos *many-objective* em virtude das abordagens precedentes não conseguirem resolver o problema do *roteamento multicast* em um tempo satisfatório.

Segundo o artigo, o intuito do estudo é garantir o roteamento com qualidade de serviço(QoS) para que seja obtido um nível adequado de comunicação fim-a-fim. Para isso, foi feito o uso de um algoritmo *many-objective* denominado AEMMT(BRASIL; DELBEM; SILVA, 2013). Sob o preceito de fortificar a convergência das soluções encontradas, foi proposto um novo algoritmo baseado no AEMMT chamado de Algoritmo Evolutivo Multi-objetivo com Múltiplas Dominâncias(AEMMD).

Foi concluído que o AEMMD teve rendimento significativamente superior ao AEMMT.

Além disso, os dois algoritmos citados previamente superaram o SPEA2 a partir de quatro funções objetivo. Portanto, justifica-se o uso dos algoritmos *many-objective* para resolução de problemas com uma quantidade maior de objetivos como, por exemplo, o *Problema do Roteamento Multicast (PRM)*.

2.8.2 A Comparative Analysis of MOEAs Considering Two Discrete Optimization Problems

Este artigo(FRANÇA et al., 2017) faz um estudo de comportamento de 5 algoritmos sendo que dois deles são multi-objetivo(SPEA2 e NSGA-II), outros dois são considerados *many-objective*(NSGA-III e AEMMT) e um é considerado uma transição entre esses dois tipos de algoritmos(MOEAD).

É feita uma análise do comportamento desses algoritmos sobre o problema da mochila com multi-objetivos, e o *Problema do roteamento Multicast (PRM)*. As métricas utilizadas para essa análise são as mesmas que se fizeram uso nesse estudo(*Error Rate*, *Pareto Subset* e *Generational Distance*).

3 Experimentos e Resultados

Esta seção descreve os experimentos realizados, bem como as métricas de desempenho aplicadas para avaliar os algoritmos evolutivos NSGAII, SPEA2, AEMMT, AEMMD e MOEA/D e os parâmetros utilizados.

Todas as implementações foram analisadas na resolução do problema da mochila. Diferentes instâncias de problemas foram avaliadas, tendo como variáveis o número de itens na mochila e a quantidade de objetivos.

3.1 Especificação do método de avaliação

Foi feita uma média de desempenho de acordo com as métricas de erro, *pareto subset* e *generational distance* considerando 30, 50 e 100 itens. Tais métricas foram aplicadas aos resultados obtidos através das execuções dos algoritmos NSGAII, SPEA2, AEMMT, AEMMD e MOEA/D, os quais estão mostrados na forma de gráfico.

3.1.1 Parâmetros

Para deixar a comparação mais justa, foram estabelecidos alguns parâmetros, retirados de (FRANÇA et al., 2017), para que se tenha chegado a esses resultados. Todos os valores presentes, que serão exibidos em tabelas, foram aplicados a uma mesma instância de problema da mochila.

Além disso, foi determinado um número de 100 gerações para os algoritmos *NSGA-II*, *SPEA2* e *MOEA/D* aplicados a problemas de 30 e 50 itens e 200 gerações para problemas com 100 itens. Foi utilizada uma taxa de *mutação* de 5% e 100% de *crossover* para todos os algoritmos. No que diz respeito aos algoritmos *AEMMD* e *AEMMT* foi determinado uma abordagem de 15000 gerações tendo em vista que cada geração produz apenas 2 filhos. Essa abordagem justifica-se quando se tem como base de comparação a quantidade de filhos inseridos a cada geração que é a mesma quantidade de filhos gerados. Além disso, é importante destacar alguns parâmetros internos específicos de cada algoritmo:

- **Tamanho das vizinhanças no MOEA/D** : 10
- **Tamanho das tabelas no AEMMT** : 50

A população inicial de indivíduos foi gerada a partir de sementes pseudo-aleatórias.

3.2 Fronteiras de Pareto

Os valores associados a cada métrica foram obtidos tendo como comparativo uma fronteira de Pareto para cada instância de problema. Ela foi determinada por meio de repetidas execuções de todos os algoritmos, uma vez que, todos os indivíduos não dominados eram salvos em um arquivo. Essas amostras foram retiradas de (FRANÇA et al., 2017). É importante salientar que não existe garantia que esses arquivos representem a fronteira de Pareto, portanto trata-se de uma estimativa da fronteira de Pareto.

A tabela a seguir apresenta a cardinalidade da fronteira de Pareto obtida em cada instância (I, O) onde I é o número de itens da mochila e O o número de objetivos utilizado no experimento.

	2 Objetivos	3 Objetivos	4 Objetivos	5 Objetivos	6 Objetivos
30 Itens	15	106	425	1765	5800
50 Itens	67	501	986	5213	35760
100 Itens	170	6288	88374	176868	248198

Tabela 1 – Cardinalidade das Fronteiras de Pareto

3.3 Experimento 1: Problema da Mochila com Dois Objetivos

3.3.1 30 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAII	0	15	0
SPEA2	1.5973958137597928	13	14.28571428571429
AEMMT	0	15	0
AEMMD	100	0	26.082510579127163
MOEA/D	0	13	0

Tabela 2 – 2 Objetivos - 30 Itens

Como era esperado, devido a baixa complexidade do problema e do tamanho pequeno da fronteira de pareto, os algoritmos NSGAII, AEMMT e MOEA/D, obtiveram um ótimo desempenho. Embora apresente valores de erro e *generational distance* maiores que 0, o SPEA2 atingiu um desempenho razoável nessas métricas. O algoritmo AEMMD não teve convergência nesse cenário, não sendo capaz de encontrar qualquer solução do Pareto.

3.3.2 50 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAI	26.190476190476186	31	0.6142733838524725
SPEA2	29.799477619186053	27	0.965547344878765
AEMMT	61.290322580645164	16	1.7364219295334833
AEMMD	100	0	35.948560048032452
MOEA/D	53.84615384615385	12	1.5114982925099874

Tabela 3 – 2 Objetivos - 50 Itens

A partir deste cenário, os algoritmos multi-objetivos (SPEA2 e NSGAI) passaram a destacar-se em relação ao AEMMT e MOEA/D em qualquer métrica observada. Fica destacada a grande diferença da taxa de erro entre esses dois tipos de algoritmos bem como a proximidade das respostas encontradas por eles, o que pode ser observado pela métrica de *generational distance*. Novamente, o AEMMD não apresentou convergência ao Pareto.

3.3.3 100 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAI	51.42857142857142	34	0.7190956130385407
SPEA2	55.59350235793263	29	0.877635715428937
AEMMT	100	0	8.470392587545271
AEMMD	100	0	65.03787994752211
MOEA/D	91.66666666666666	2	2.327850432239705

Tabela 4 – 2 Objetivos - 100 Itens

O comportamento dos algoritmos é mantido se comparado com o problema de 2 objetivos para 50 itens, ou seja, os algoritmos multi-objetivos mantêm-se superiores aos demais. Nesse cenário, dentre os *many-objective*, apenas o MOEA/D conseguiu encontrar alguma solução do Pareto. Os algoritmos AEMMT e AEMMD não convergiram.

3.4 Experimento 2: Problema da Mochila com Três Objetivos

3.4.1 30 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAII	11.627047178707048	64.7	0.5246493008288773
SPEA2	19.999999999999996	16	1.6159616504737397
AEMMT	14.200751063516911	60.233333333333334	0.6151037168286724
AEMMD	99.666666666666674	0.33333	17.74050844142331
MOEA/D	21.838207971387472	28.033333333333335	1.1626839576827275

Tabela 5 – 3 Objetivos - 30 Itens

Com três objetivos, os algoritmos *many-objective* começam a apresentar melhorias na convergência ao Pareto. O AEMMT quase alcança o NSGAII em termos de *pareto subset* e *erro*. O MOEA/D começa a apresentar melhoras significativas em seus resultados. Com exceção do AEMMD, os demais *many-objective* ultrapassaram o SPEA2 no número de soluções encontradas no pareto e apresentam menor *generational distance*. O AEMMD apresentou uma convergência mínima ao Pareto.

3.4.2 50 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAII	42.02855656410715	72.366666666666666	0.7237890756497811
SPEA2	18.181818181818176	9	1.741596789795353
AEMMT	74.85539178385926	25.333333333333332	1.4317770134526877
AEMMD	100	0	33.93938967876152
MOEA/D	40.18678933430121	36.633333333333333	1.0881888503609494

Tabela 6 – 3 Objetivos - 50 Itens

Os métodos AEMMT e AEMMD têm um retrocesso se comparados ao cenário anterior (de 3 objetivos e 30 itens) pois suas soluções ficam mais distantes do Pareto. Apesar de ter o erro bem menor do que todos os outros algoritmos, o SPEA2 perde no número de soluções encontradas no Pareto.

3.4.3 100 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAI	95.17241379310344	4.233333333333333	1.6163521901021496
SPEA2	70.96774193548387	9	2.101365448017808
AEMMT	98.82112565876153	2.133333333333333	3.1801380647189834
AEMMD	100	0	56.79830167389674
MOEA/D	81.38327546948582	69.26666666666667	0.6723672326386526

Tabela 7 – 3 Objetivos - 100 Itens

Com exceção do MOEA/D, os demais algoritmos apresentaram um *pareto subset* baixo. Novamente, o SPEA2 tem a menor taxa de erro mas não consegue igualar o número de soluções encontradas no Pareto. O AEMMT apresenta uma convergência mínima ao Pareto e o AEMMD continua nulo.

3.5 Experimento 3: Problema da Mochila com Quatro Objetivos

3.5.1 30 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAI	22.867242426607145	102.46666666666667	0.7812762403144072
SPEA2	6.666666666666665	42	0.8083695945599921
AEMMT	2.228334883668298	203.8	0.17763581959609548
AEMMD	66.9220138334823	14.8	3.49653106386687
MOEA/D	13.01393658239908	83.26666666666667	0.667978402130763

Tabela 8 – 4 Objetivos - 30 Itens

Com quatro objetivos, percebe-se que os algoritmos *many-objective* começam a sobressair. Especialmente o AEMMT, que supera os demais em todos os quesitos de avaliação. O NSGAI apresenta um desempenho superior ao MOEA/D em termos de *pareto-subset* apesar de perder nas métricas de erro e *generational distance*. Nesse cenário, o AEMMD apresenta pela primeira vez uma convergência mínima ao Pareto, embora ainda seja superado por todos os outros métodos.

3.5.2 50 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAI	38.85106129182859	87.8	0.9327278777125526
SPEA2	29.411764705882348	12	2.093620206143562
AEMMT	45.87353247422563	103.9793528739	0.8817867254280334
AEMMD	100	0	30.97115941378762
MOEA/D	21.088867786515678	113.8	0.5677031566699826

Tabela 9 – 4 Objetivos - 50 Itens

No cenário com 50 itens, o MOEA/D apresenta melhor desempenho nas 3 métricas avaliadas. Também se torna evidente o predomínio dos algoritmos AEMMT e MOEA/D sobre os multi-objetivo na métrica de *pareto subset*. Por outro lado, em relação ao erro, o AEMMT perde para todos com exceção do AEMMD (que não apresenta convergência). Com relação a *generational distance*, NSGAI, AEMMT e MOEA/D apresentam resultados similares, ao passo que SPEA2 fica um pouco atrás mas ainda à frente do AEMMD que piorou significativamente com relação ao problema anterior.

3.5.3 100 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAI	99.77414088061226	0.3333333333	2.747133418928517
SPEA2	100	0	5.861938168949844
AEMMT	86.2258953168044	100	0.5676423942741415
AEMMD	100	0	31.91994756480328
MOEA/D	70.40428388192954	192.83333333333334	0.47209051652923756

Tabela 10 – 4 Objetivos - 100 Itens

No cenário com 100 itens, todos os métodos apresentam baixa convergência, mas os *many-objective* AEMMT e MOEA/D são superiores ao NSGAI em todos os critérios. Os multi-objetivos já não são capazes de encontrar dados significativos. O AEMMD e SPEA2 não convergiram para o Pareto.

3.6 Experimento 4: Problema da Mochila com Cinco Objetivos

3.6.1 30 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAII	33.156134151133394	94.8	1.154631444141741
SPEA2	18.6046511627907	70	1.0889804929513092
AEMMT	1.9738970767078134	511.96666666666664	0.1010195527233637
AEMMD	35.35517993847319	250.13333333333333	0.8130986437189467
MOEA/D	6.900387354155226	294.4	0.3087243389139499

Tabela 11 – 5 Objetivos - 30 Itens

Para essa instância de problema da mochila, o AEMMD teve um bom resultado em termos de *pareto subset*. No entanto, ele ainda apresenta a taxa mais elevada de erro de todos os algoritmos. O AEMMT é o algoritmo que apresentou o melhor desempenho nas 3 métricas.

No que diz respeito aos multi-objetivos, a diferença no *pareto subset* do NSGA-II e do SPEA2 em relação aos demais métodos é acentuada. Com relação as outras duas métricas, o SPEA2 foi superior ao seu concorrente multi-objetivo.

3.6.2 50 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAII	65.3061224489796	51	1.4437967069351239
SPEA2	51.02040816326531	24	1.9316998085973258
AEMMT	23.50451784566922	281.7	0.2578257994789109
AEMMD	94.90192848126931	13.666666666666666	0.415182614511281
MOEA/D	14.054054054054054	159	0.47483181028002724

Tabela 12 – 5 Objetivos - 50 Itens

Os algoritmos *many-objective* MOEA/D e AEMMT obtiveram desempenho superior a todos os outros algoritmos com destaque para a métrica de *pareto-subset* e erro. O AEMMD retorna os piores resultados no que diz respeito ao *pareto-subset* e erro. No entanto, ele obteve melhores resultados em *generational-distance* se comparado com os algoritmos multi-objetivo. Os métodos NSGAII e SPEA2 retornam resultados inferiores ao AEMMT e MOEA/D.

3.6.3 100 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAI	99.18957898576623	1.2	3.509219783253927
SPEA2	100	0	3.816885342414409
AEMMT	83.30354238427502	110	0.8071767282182432
AEMMD	100	0	14.079761237384515
MOEA/D	58.22765884028162	541.9	0.33984459092125924

Tabela 13 – 5 Objetivos - 100 Itens

Com 100 itens, os algoritmos MOEA/D e AEMMT foram os únicos que obtiveram resultado significativo em relação ao número de indivíduos encontrados que estão no pareto. Mesmo assim eles obtiveram uma taxa de erro bem elevada, o que pode ser atribuído a alta quantidade de itens na mochila.

No entanto, é possível notar que existe um decréscimo nessa taxa para os algoritmos citados se comparados com os problemas da mochila de 100 itens anteriores. Isso é um outro indicativo de que os *many-objective* tem uma performance melhor à medida que a complexidade dos problemas aumentam em número de objetivos e quantidade de itens.

3.7 Experimento 5: Problema da Mochila com Seis Objetivos

3.7.1 30 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAII	46.72045375244235	77.23333333333333	1.3944777796731538
SPEA2	44.230769230769226	29	1.9978251753813463
AEMMT	2.664789696315751	620.0333333333333	0.12369501602979797
AEMMD	11.85022519041143	1219.5333333333333	0.175371781666211
MOEA/D	21.77597797231459	265.03333333333336	0.5401044834165805

Tabela 14 – 6 Objetivos, 30 Itens

No cenário com 30 itens e 6 objetivos, fica evidente a superioridade dos algoritmos *many-objective* sobre os multi-objetivo, sendo que o AEMMT retornou a melhor convergência (erro e *generational distance*) enquanto o AEMMD retornou o maior *pareto subset*. Nessa instância de problema, o AEMMD apresentou seu melhor desempenho de todos os cenários analisados com uma taxa de erro relativamente baixa e uma quantidade muito alta de indivíduos presentes no Pareto na solução final (quase o dobro em relação ao segundo colocado).

3.7.2 50 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAII	86.01621690192724	20.8	2.887668271412389
SPEA2	22.44897959183674	38	1.3843171300797987
AEMMT	10.17423565659337	830.2	0.23672412841777846
AEMMD	60.23224737889814	692.6333333333333	0.6807642244246186
MOEA/D	27.458258068689982	554.0333333333333	0.3916463182058507

Tabela 15 – 6 Objetivos - 50 Itens

Para essa instância de problema, todos os *many-objective* foram superiores aos multi-objetivos com relação ao *pareto-subset* e à *generational distance*. O SPEA2, apesar de achar poucos indivíduos no Pareto, teve uma taxa de erro baixa se comparada com os outros algoritmos para esse problema, perdendo apenas para o AEMMT.

3.7.3 100 Itens

	Error (er)	Pareto Subset (ps)	Generational Distance (gd)
NSGAI	94.82228591108537	7.7	4.004529203300145
SPEA2	90.9090909090909	3	3.69881234383409
AEMMT	43.81578947368420	427	0.48909601798822727
AEMMD	100	0	7.601018734874814
MOEA/D	51.02667632815321	775.7333333333333	0.3541664017007407

Tabela 16 – 6 Objetivos - 100 Itens

O AEMMD, SPEA2, e NSGAI apresentaram um desempenho ruim, praticamente sem convergência ao Pareto. O destaque fica com o MOEA/D que obteve a melhor solução no *pareto-subset* e *generational-distance*. O AEMMT também teve uma boa performance com o menor erro dentre todos os algoritmos.

3.8 Análise comparativa por métrica

Após as análises de cada cenário individual apresentado nas seções anteriores, foi feita uma outra análise em forma de gráfico que demonstra o comportamento dos algoritmos para cada métrica já descrita. Primeiramente, serão exibidos cinco gráficos referentes a cada métrica de erro, onde cada um deles representa uma quantidade diferente de objetivos, sendo que, em todos, a exibição dos dados é feita considerando variações da quantidade de itens da mochila (30, 50 e 100). Após isso, serão exibidos mais três gráficos referentes à média de cada métrica dos gráficos anteriores e um referente à média de tempo de execução desses algoritmos.

3.8.1 Erro

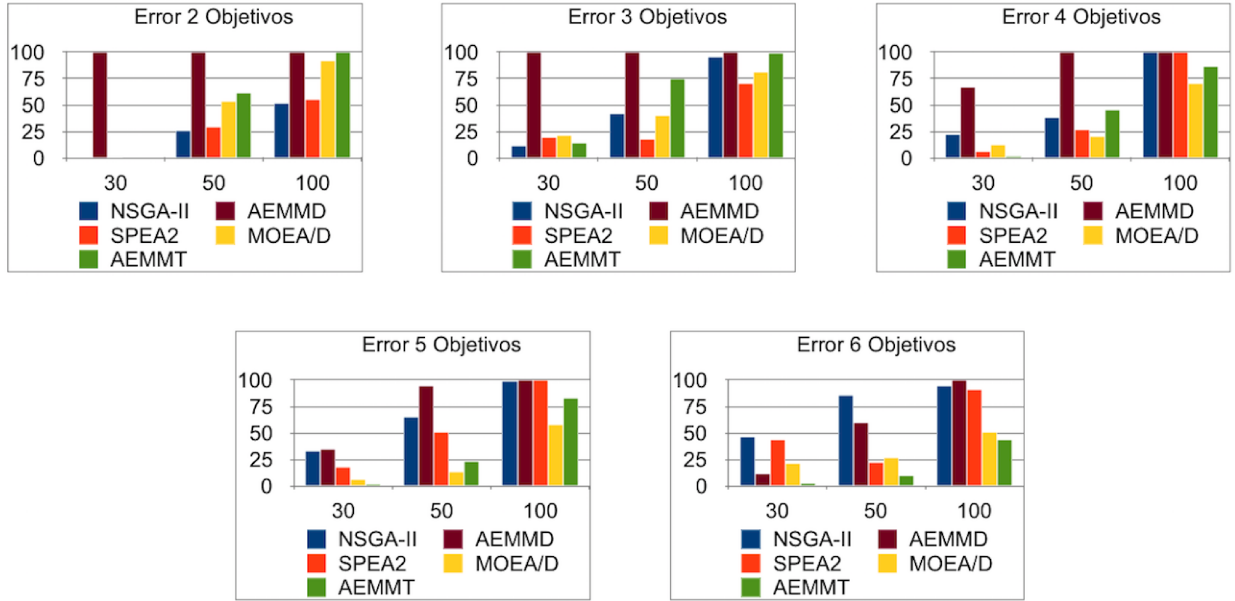


Figura 2 – Comparativo Detalhado de Erro

Observa-se pelos gráficos da Figura 2 que o desempenho, em termos de erro, dos algoritmos multi-objetivos piora à medida que o número de objetivos no problema da mochila aumenta. Por outro lado, ocorre o inverso para os algoritmos AEMMT e MOEA/D, ou seja, o desempenho na métrica erro melhora à medida que o número de objetivos é ampliado.

O AEMMD também segue esse comportamento, embora a mudança seja menos significativa já que ele apresenta altas taxas de erro para 4, 5 e 6 objetivos se comparado com o restante dos algoritmos *many-objective* e também demora a mostrar resultados significativos em relação aos outros algoritmos.

Também fica evidente que a adição do número de itens na mochila é um critério impactante para o aumento da taxa de erro em todos os algoritmos. Vale ressaltar algumas exceções:

- para 3 e 6 objetivos, o SPEA2 não seguiu esse padrão;
- o AEMMD apenas começa a mostrar esse tipo de comportamento a partir de 4 objetivos, uma vez que nas instâncias anteriores a 4 objetivos de problema da mochila, o AEMMD não apresentou convergência ao Pareto, resultando em erro 100% independente do número de itens.

3.8.2 Pareto Subset

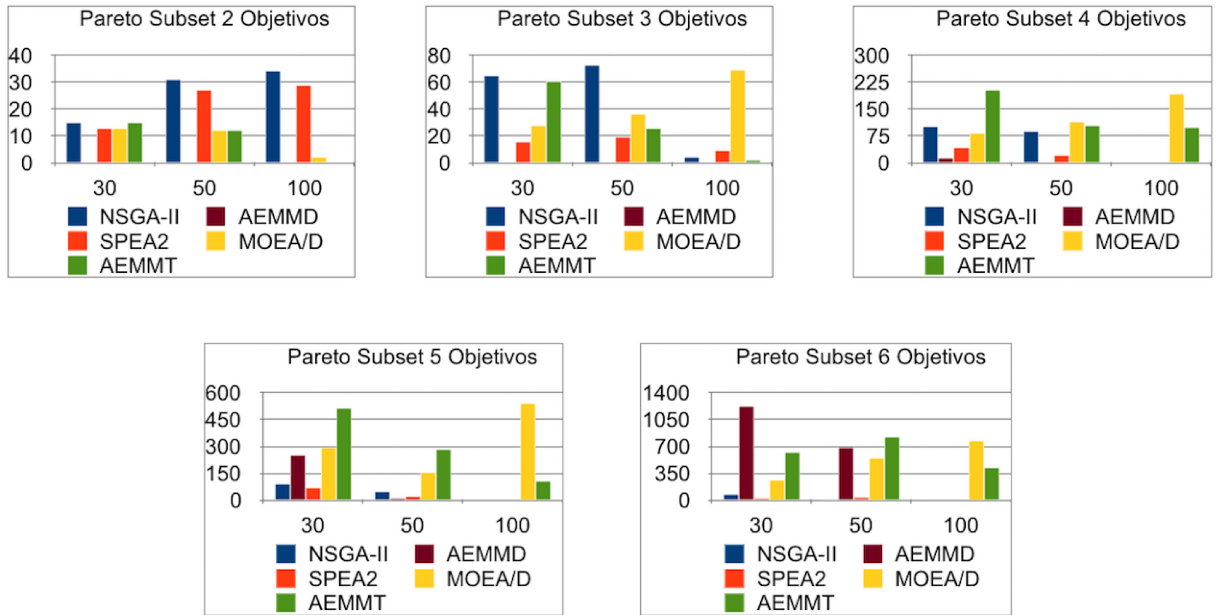


Figura 3 – Comparativo Detalhado de *Pareto Subset*

Conforme a Figura 3, para 2 objetivos e com 30 itens todos os algoritmos (com exceção do AEMMD) retornam resultados satisfatórios devido à baixa complexidade do problema. No entanto fica evidente a superioridade dos multi-objetivos aos *many-objectives* quando o número de itens é aumentado para 50 ou 100. Esse comportamento começa a mudar a partir de 3 objetivos. Para 30 e 50 itens o SPEA2 é superado pelo MOEA/D e AEMMT. No entanto, o NSGA-II retorna os maiores valores de *pareto subset* dentre os 5 algoritmos. Com 100 itens, o comportamento altera-se bastante e o MOEA/D supera todos os demais. Em segundo lugar, fica o SPEA2.

A partir de 4 objetivos, conforme o número de itens aumenta, percebe-se que os multi-objetivos tornam-se cada vez menos eficazes. De forma geral, o MOEA/D apresenta uma performance cada vez mais elevada à medida que a quantidade de objetivos e itens aumentam. Isso já não pode ser observado para o AEMMD e AEMMT que, apesar do *pareto subset* retornado ter aumentado com o acréscimo do número de objetivos, ele diminui com o aumento no número de itens.

Assim como foi observado na métrica de erro, o AEMMD demora mais para apresentar resultados significativos do que os demais *many-objectives* e apenas torna-se competitivo com 5 objetivos e para 30 itens. É importante destacar que ele obteve um resultado muito superior aos demais com 6 objetivos e 30 itens e o segundo melhor desempenho com 50 itens. Com 6 objetivos e 100 itens, novamente o AEMMD não apresenta convergência.

De forma geral, com 5 e 6 objetivos, o AEMMT e o MOEA/D retornam o melhor desempenho na métrica *pareto subset*.

3.8.3 Generational Distance

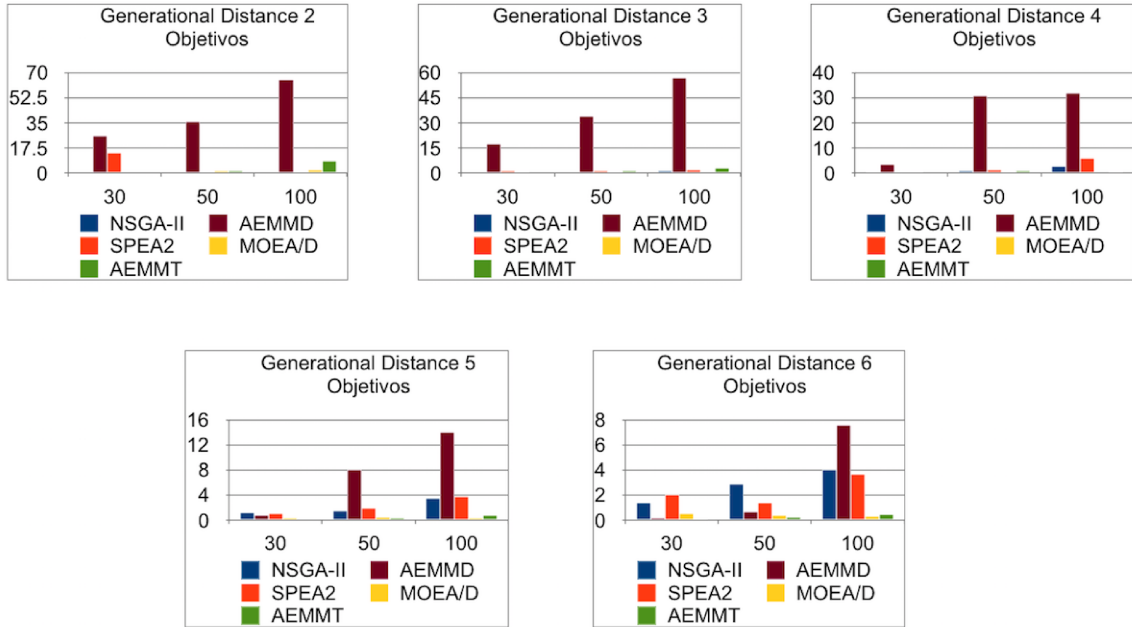


Figura 4 – Comparativo Detalhado de *Generational Distance*

Os gráficos referentes à métrica de *generational distance* apresentados na Figura 4 ficam bem distorcidos por conta do AEMMD que, na média, encontra soluções muito distantes do Pareto. Assim, os gráficos foram refeitos excluindo-se o AEMMD e são apresentados na Figura 5.

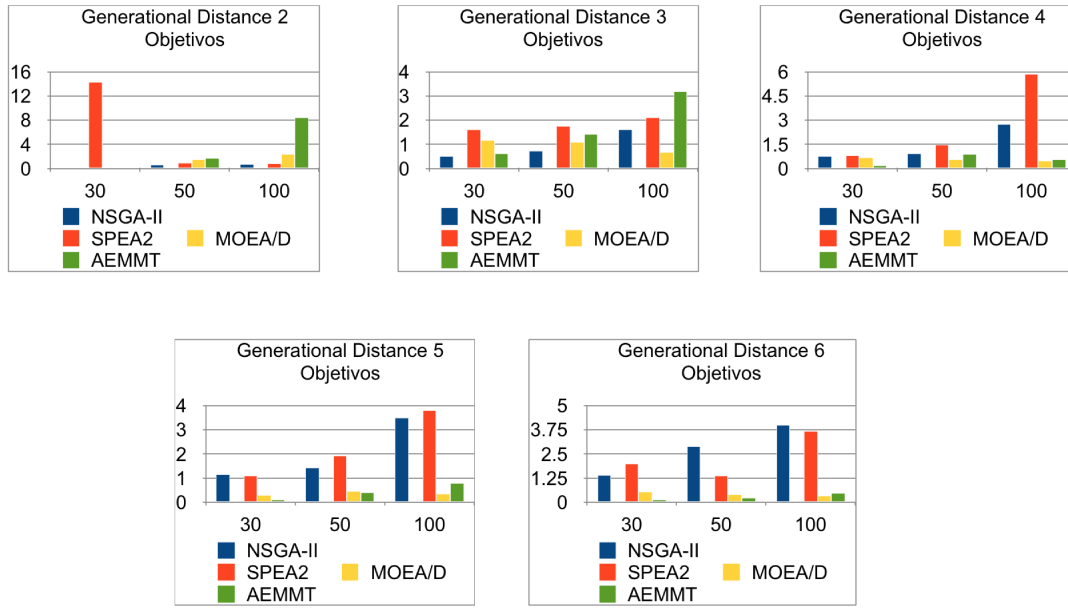


Figura 5 – Comparativo Detalhado de *Generational Distance* sem o AEMMD

De acordo com a Figura 5, para 2 objetivos e 30 itens percebe-se que o SPEA2 apresenta um valor elevado mas diminui com o aumento de itens. Nota-se um aumento significativo no algoritmo AEMMT para 100 itens.

Em um cenário com 3 objetivos, nota-se que o SPEA2 mantém-se praticamente estável independentemente do número de itens. O NSGAII, para 30 e 50 itens, apresenta os melhores valores mas perde para o MOEA/D com 100 itens. Diferentemente do AEMMT, o MOEA/D melhora seus valores de *generational distance* à medida que o número de itens aumenta.

Para 4 e 5 objetivos, o comportamento dos algoritmos multi-objetivos NSGAII e SPEA2 é de um incremento em seus valores conforme o número de itens aumenta. Eles apresentam, para 50 e 100 itens, valores mais elevados se comparados com os algoritmos *many-objective* AEMMT e MOEA/D. Esses dois últimos mantêm-se praticamente estáveis de forma independente do acréscimo de itens na mochila.

Com 6 Objetivos o comportamento do algoritmo SPEA2 apresentou uma diferença com relação aos cenários de 4 e 5 objetivos pois obteve um decréscimo no valor retornado de 30 para 50 itens. O algoritmo MOEA/D seguiu o comportamento dos cenário de 3 e 4 objetivos, diminuindo conforme o número de itens aumentava. O AEMMT mostrou um pequeno aumento se considerado o cenário de 30 itens, no qual o valor retornado é quase insignificante com relação aos demais algoritmos, para o cenário de 100 itens.

3.9 Análise comparativa da média das métricas

As próximas seções apresentam uma análise comparativa, considerando-se a média dos gráficos anteriores com relação ao número de itens da mochila, resultando em um gráfico para cada número de objetivos.

3.9.1 Erro Médio

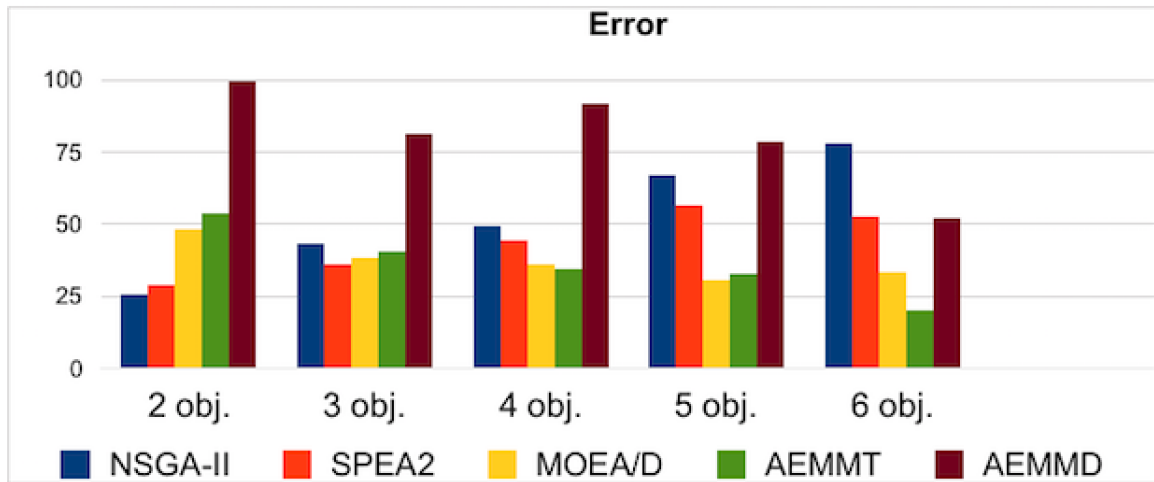


Figura 6 – Comparativo de Erro

A média obtida (calculada considerando-se 30, 50 e 100 itens) exibida em forma de gráfico na Figura 6 para cada formulação de objetivos deixa evidente que o desempenho dos algoritmos multi-objetivos pioram à medida que o número de objetivos aumenta. Por outro lado, percebe-se um decréscimo na taxa de erro nos algoritmos MOEA/D e AEMMT, sendo que esse último teve uma redução na sua taxa de erro em mais de 50% se compararmos o desempenho de 2 para 6 objetivos. Também observa-se uma queda, menos acentuada do que nos dois últimos algoritmos, no AEMMD que tem sua taxa de erro praticamente igual ao SPEA2 e muito menor do que o NSGAII na formulação com 6 objetivos.

Os dados desse gráfico permitem concluir também que, à medida que a complexidade do problema, em número de objetivos, aumenta, a taxa de erro dos algoritmos *many-objective* diminui.

3.9.2 Pareto Subset Médio

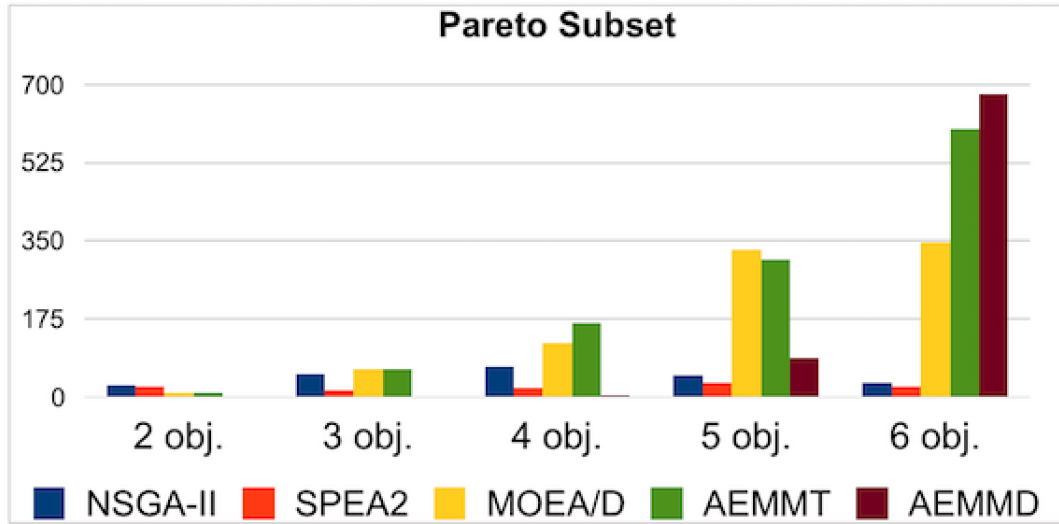


Figura 7 – Comparativo de *Pareto Subset*

Para 2 objetivos, pela análise dos gráficos na figura 7, percebe-se que os algoritmos multi-objetivos NSGAII e SPEA2 são competitivos em relação aos demais. O NSGAII, inclusive, mantém-se competitivo com 3 objetivos. Com 4 ou mais objetivos, a superioridade dos *many-objective* AEMMT e MOEA/D fica evidente. Por outro lado, a partir de 5 objetivos o *many-objective* AEMMD torna-se melhor nessa métrica que o SPEA2 e NSGAII.

De acordo com a média para 6 objetivos a conclusão é de que o AEMMD foi o melhor algoritmo mesmo não encontrando nenhum indivíduo no Pareto para 100 itens. Isso se justifica pois ele apresentou desempenho muito superior para 30 itens e um bom desempenho para 50 itens.

3.9.3 *Generational Distance* Médio

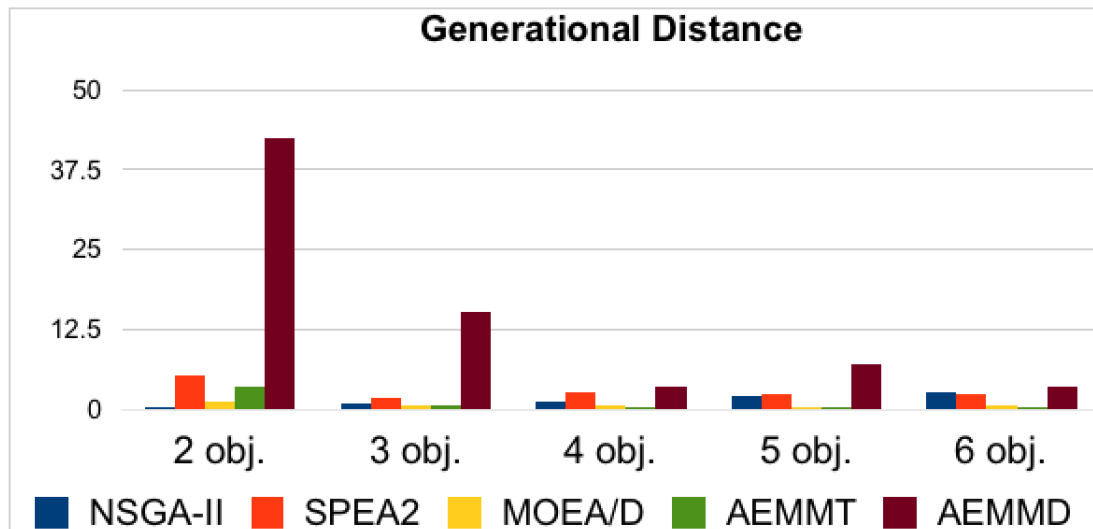


Figura 8 – Comparativo de *Generational Distance*

Analisando-se a Figura 8, percebe-se que o AEMMD distorce muito o gráfico, pois seus valores para *generational distance* são muito elevados. Por isso, será realizada uma análise sem o AEMMD.

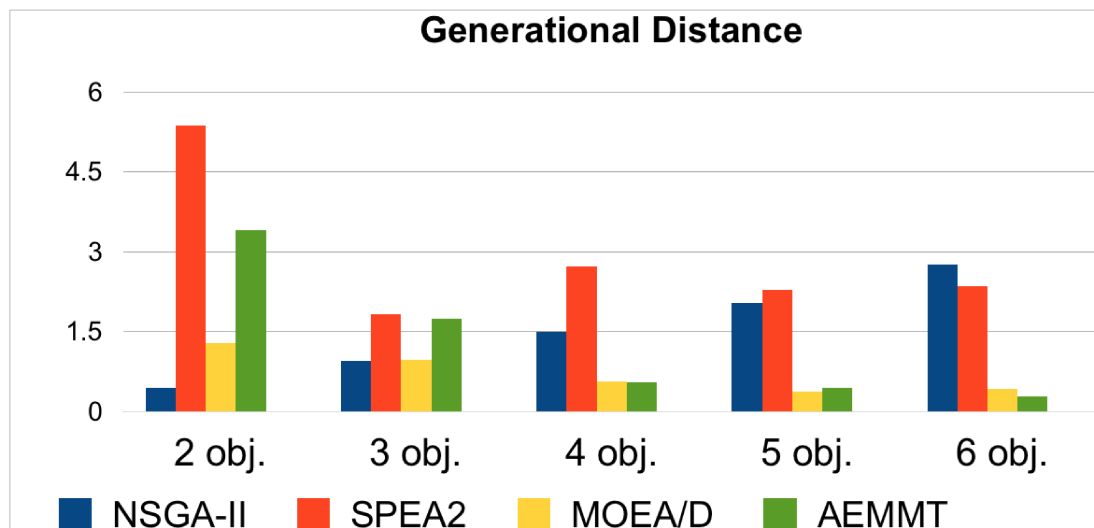


Figura 9 – Comparativo de *Generational Distance* sem o AEMMD

Analisando-se a Figura 9 percebe-se que o NSGAII obteve um acréscimo gradativo nessa métrica conforme o número de objetivos aumentou. O SPEA2, para 2 objetivos, foi

o algoritmo com o maior valor retornado, muito superior a qualquer um dos outros. Ele apresentou uma melhora significativa para 3 objetivos e um aumento no cenário de 4 objetivos, mantendo-se estável para os demais.

Com relação aos algoritmos *many-objective* AEMMT e MOEA/D, para um cenário de 4 objetivos em diante, ambos mantêm-se praticamente inalterados e em um mesmo valor. Com 2 e 3 objetivos o MOEA/D é melhor do que o AEMMT e até mostra-se competitivo com o NSGAII para 3 objetivos.

Fica evidente a superioridade do AEMMT e MOEA/D em relação aos demais algoritmos para os cenários de 4, 5 e 6 objetivos.

3.10 Tempo de execução

Após a análise das métricas multi-objetivos erro, *pareto subset* e *generational distance*, realizou-se uma análise final sobre o tempo de processamento.

Os valores do tempo de execução foram obtidos em um computador com processador 2.7 GHz Intel Core i5, 8GB de memória RAM e com memória APPLE SSD SM0256G. A linguagem de programação utilizada para implementar esses algoritmos foi JAVA.

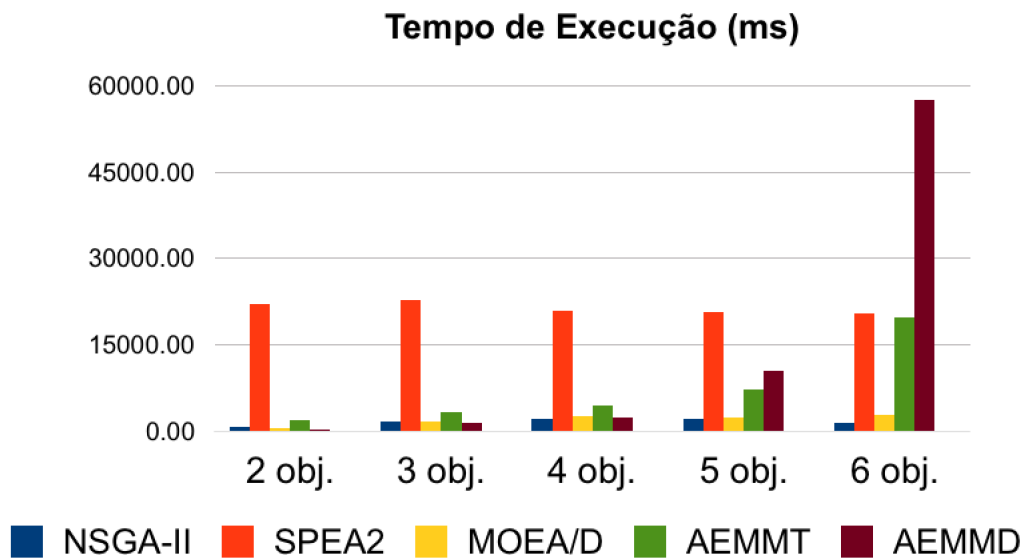


Figura 10 – Comparativo de *Tempo de Execução*

O tempo de execução do SPEA2 não apresentou muita alteração conforme o número de objetivos aumentou. Ele obteve o maior valor no gráfico em quase todos os cenários de testes, perdendo apenas para o AEMMD com 6 objetivos.

Percebe-se o mesmo comportamento para os algoritmos NSGA-II e MOEA/D. Os dois algoritmos tem tempo de execução similar apesar do *many-objective* ter uma performance muito superior acima de 4 objetivos nas outras métricas avaliadas. Esses dois

algoritmos não tem um aumento significativo de complexidade a medida que o número de objetivos aumenta.

No entanto, os algoritmos AEMMT e AEMMD apresentaram um aumento significativo nos seus respectivos tempos de execução, uma vez que o número de tabelas utilizado para resolver o problema está diretamente relacionado a quantidade de objetivos dele. Logo, quanto maior a quantidade de objetivos para esses algoritmos, maior será o tempo de execução deles. Entretanto, o AEMMT até 6 objetivos, é competitivo em relação ao SPEA2, mesmo com esse aumento de tempo.

4 Conclusão

Neste trabalho, foram implementados os algoritmos evolutivos multi-objetivos NSGAII e SPEA2 e os *many-objective* como AEMMT, AEMMD e MOEA/D. Foram utilizadas métricas de desempenho geralmente empregadas em problemas com múltiplos objetivos como erro, *pareto subset* e *generational distance*, para avaliar os algoritmos citados. Além disso, o tempo de execução foi avaliado.

No que diz respeito ao tempo de execução, e em se tratando de algoritmos *many-objective*, apenas o MOEA/D obteve tempo relativamente igual em todos os cenários de testes. Os multi-objetivos, também apresentaram essa característica. O AEMMT e AEMMD demonstraram tempo de execução crescente com a complexidade do problema, ou seja, número de objetivos utilizados.

Os algoritmos evolutivos multi-objetivos testados apresentaram dados superiores aos *many-objective* para 2 e 3 objetivos. Em alguns casos de testes, até chegaram a ser competitivos com 4 objetivos. No entanto, conforme a complexidade do problema, em termos de número de objetivos e itens na mochila aumentava, o desempenho desses algoritmos diminuiu de acordo com as métricas analisadas.

O oposto pôde ser observado no que diz respeito aos algoritmos *many-objective*, mais especificamente para o AEMMT e o MOEA/D que apresentaram uma melhora significativa nos seus dados a partir de 4 objetivos. O AEMMD apenas mostra seu potencial com 5 objetivos ou mais e para problemas de 30 e 50 itens. No entanto, mesmo para essas instâncias, ele apresenta um alto valor em *generational distance* o que significa que as soluções fora do Pareto estão muito distantes dele.

Esse trabalho foi proposto com o intuito de mostrar a força dos algoritmos *many-objective* em um problema de otimização discreta em vários objetivos se comparado com os multi-objetivo. Espera-se também, que esse trabalho sirva de base para consulta e escolha da melhor abordagem de resolução de um problema utilizando-se um dos algoritmos estudados.

A partir dos dados observados, para 2 objetivos, os algoritmos multi-objetivo apresentam desempenho superior em relação aos *many-objective* em quase todas as métricas. A única exceção foi o SPEA2 que obteve um tempo de execução e *generational distance* altos. O NSGAII é o algoritmo mais indicado para essa quantidade de objetivos.

Já com 3 objetivos, o MOEA/D e AEMMT começam a ficar competitivos em relação ao NSGAII pois apresentam *generational distance*, *pareto subset* e erro semelhantes. No que diz respeito ao tempo de execução, o AEMMT é pior que o MOEA/D e o NSGAII.

Para essa instância de problema da mochila o MOEA/D ou o NSGAII seriam a melhor indicação.

Com 4 objetivos percebe-se que o *generational distance* dos algoritmos multi-objetivos começa a aumentar. O NSGAII obtém um resultado interessante no *pareto subset* mas fica bem atrás do MOEA/D e do AEMMT. O mesmo ocorre com o erro. Em termos de tempo de execução o MOEA/D sobressaiu-se em relação aos outros mas perde para o AEMMT em relação ao número de indivíduos encontrados que estão na fronteira de Pareto.

Somente com 5 objetivos percebe-se uma nítida diferença entre os multi e *many-objective*. O NSGAII não apresenta bom desempenho em nenhuma métrica com exceção do tempo de execução. No entanto, isso não consiste em uma justificativa para o seu uso nessa instância de problema. O AEMMD começa a mostrar seu potencial mas ainda fica atrás do AEMMT e MOEA/D. Os dois apresentam resultados semelhantes sendo que o MOEA/D torna-se superior na métrica de *pareto subset*.

Com 6 objetivos, apesar do tempo de execução alto, o AEMMD é o mais indicado para 30 e 50 itens. Para 100 itens, o AEMMT obteve um melhor resultado. No entanto caso o fator tempo de execução for impactante o MOEA/D é o mais indicado.

De acordo com a análise feita, esse trabalho cumpre com seu objetivo ao mostrar a eficácia dos algoritmos *many-objective* em relação aos multi-objetivo a partir de 4 objetivos e para o problema da mochila em um cenário de 30, 50 e 100 itens.

4.1 Dificuldades Encontradas

Uma grande dificuldade foi assimilar os conceitos relacionados aos algoritmos evolutivos como fronteira de Pareto e os conceitos de dominância. Além disso, compreender como aplicar o problema da mochila nos algoritmos estudados também demandou tempo.

A escassez de documentação para implementação de como determinar os vizinhos próximos do SPEA2 foi algo impactante em termos de tempo pois o artigo original não explicita como essas distâncias são estabelecidas. Além disso, o SPEA2 se mostrou o algoritmo mais complicado de se entender e foi o que mais demandou tempo de implementação.

A implementação do NSGA-II também provou-se de dificuldade elevada por conta de ter sido o primeiro algoritmo a ser implementado. Os conceitos de fronteira de Pareto e dominância ainda não estavam bem estabelecidos. O método de seleção de *crowding distance* demorou a ser implementado por conta disso.

No que diz respeito aos *many-objective* o AEMMT e AEMMD foram relativamente simples de se implementar com relação ao MOEA/D. O conceito de vizinhança desse

algoritmo demorou a ser assimilado.

4.2 Trabalhos Futuros

Como proposta de trabalho futuro, um algoritmo *many-objective* considerado estado da arte que foi citado nesse trabalho e não foi implementado é o NSGA-III. Seria interessante que os resultados desse algoritmo fossem comparados com os que já foram analisados nesse trabalho. Além disso, seria interessante dar continuidade a essa investigação aplicando os algoritmos evolutivos em novos problemas aplicando os algoritmos evolutivos com múltiplos objetivos, tais como, o problema do caixeiro viajante e o problema do roteamento *multicast*. Uma outra sugestão seria o uso de métricas que não necessitam do conhecimento prévio da fronteira de Pareto.

Referências

- BADER, J.; ZITZLER, E. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation*, MIT Press, v. 19, n. 1, p. 45–76, 2011. Citado na página 19.
- BRASIL, C. R. S.; DELBEM, A. C. B.; SILVA, F. L. B. da. Multiobjective evolutionary algorithm with many tables for purely ab initio protein structure prediction. *Journal of computational chemistry*, Wiley Online Library, v. 34, n. 20, p. 1719–1734, 2013. Citado 5 vezes nas páginas 3, 9, 15, 16 e 20.
- BRINGMANN, K. et al. Approximation-guided evolutionary multi-objective optimization. In: *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*. [S.l.: s.n.], 2011. v. 22, n. 1, p. 1198. Citado na página 19.
- BUENO, M. L. d. P. Heurísticas e algoritmos evolutivos para formulações mono e multiobjetivo do problema do roteamento multicast. 2010. Citado na página 8.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182–197, Apr 2002. ISSN 1089-778X. Citado 3 vezes nas páginas 8, 9 e 12.
- FRANÇA, T. P. et al. A comparative analysis of moeas considering two discrete optimization problems. In: . [S.l.: s.n.], 2017. Citado 3 vezes nas páginas 21, 22 e 23.
- HE, Z.; YEN, G. G.; ZHANG, J. Fuzzy-based pareto optimality for many-objective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 18, n. 2, p. 269–285, 2014. Citado na página 19.
- LAFETÁ, T. F. de Q. Algoritmos evolutivos many objectives aplicados ao problema de roteamento multicast com qualidade de serviço. 2016. Citado 3 vezes nas páginas 9, 17 e 20.
- LI, M.; YANG, S.; LIU, X. Bi-goal evolution for many-objective optimization problems. *Artificial Intelligence*, v. 228, p. 45 – 65, 2015. ISSN 0004-3702. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0004370215000995>>. Citado na página 18.
- SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, MIT Press, v. 2, n. 3, p. 221–248, 1994. Citado na página 8.
- TICONA, W. G. C.; DELBÉM, A. C. B. Algoritmos evolutivos para otimização multi-objetivo. Aug 2008. Citado na página 14.
- VELDHUIZEN, D. A. V.; LAMONT, G. B. Evolutionary computation and convergence to a pareto front. In: CITESEER. *Late breaking papers at the genetic programming 1998 conference*. [S.l.], 1998. p. 221–228. Citado 2 vezes nas páginas 8 e 12.

- YUAN, Y.; XU, H.; WANG, B. An improved nsga-iii procedure for evolutionary many-objective optimization. In: ACM. *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. [S.l.], 2014. p. 661–668. Citado na página 18.
- ZHANG, Q.; LI, H. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, IEEE, v. 11, n. 6, p. 712–731, 2007. Citado 2 vezes nas páginas 9 e 18.
- ZITZLER, E. et al. Spea2: Improving the strength pareto evolutionary algorithm. In: *Eurogen*. [S.l.: s.n.], 2001. v. 3242, n. 103, p. 95–100. Citado 3 vezes nas páginas 8, 9 e 14.
- ZITZLER, E.; THIELE, L. An evolutionary algorithm for multiobjective optimization: The strength pareto approach. 1998. Citado na página 8.