

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Matheus Resende Santos

**Aplicação do NSGA-III ao problema de  
escalonamento multiobjetivo de tarefas**

**Uberlândia, Brasil**

**2024**

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Matheus Resende Santos

**Aplicação do NSGA-III ao problema de escalonamento  
multiobjetivo de tarefas**

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Paulo Henrique Ribeiro Gabriel

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2024

# Resumo

Este estudo teve como objetivo ampliar a comparação entre diferentes Algoritmos Evolutivos Multi-objetivo (AEMOs) no escalonamento de tarefas em vários processadores, incluindo o NSGA-III (Non-dominated Sorting Genetic Algorithm) à análise anterior feita por Santos (2023). Ao usar quatro métricas de avaliação – quantidade de soluções em intervalos com bons resultados, média simples, média harmônica e hiper-volume –, a pesquisa descobriu que o NSGA-III superou os outros algoritmos na maioria dos casos (95/96 na produção do melhor indivíduo). No entanto, a vantagem do NSGA-III diminuiu conforme o número de objetivos e processadores também diminuiu, com o AEMMT de média simples e harmônica apresentando resultados semelhantes em cenários menos complexos. Este trabalho concluiu que, o NSGA-III se destaca em problemas altamente complexos, otimizando múltiplos critérios simultaneamente e distribuindo melhor as soluções na Fronteira de Pareto.

**Palavras-chave:** escalonamento de tarefas, sistemas multiprocessados, algoritmos evolutivos, otimização multiobjetivo, problemas com múltiplos objetivos.

# Lista de ilustrações

Figura 1 – Grafo das dependências entre tarefas . . . . .	11
Figura 2 – Fluxograma de um AE . . . . .	14
Figura 3 – Exemplo de soluções não-dominadas. . . . .	17
Figura 4 – Ilustração do Funcionamento do NSGA-III. . . . .	19
Figura 5 – Representação da codificação do indivíduo. . . . .	23
Figura 6 – Exemplo de crossover denominado Crossover Map. . . . .	24
Figura 7 – Exemplo de crossover denominado Order Crossover. . . . .	24
Figura 8 – Exemplo de mutação aplicada ao indivíduo. . . . .	25
Figura 9 – Exemplo do processo de computação das interseções para criação de um hiperplano de três objetivos. . . . .	27
Figura 10 – Pontos de referência mostrados em um hiperplano normalizado de três objetivos com $p = 4$ . . . . .	28
Figura 11 – Ilustração do processo de associação de uma solução com a linha de referência mais próxima. . . . .	28
Figura 12 – Percentuais obtidos por AEMO nos intervalos de valores do <i>Makespan</i> em um DAG de 50 tarefas com execução de 5 objetivos. . . . .	33
Figura 13 – Percentuais obtidos por AEMO nos intervalos de valores do <i>Makespan</i> em um DAG de 50 tarefas com execução de 4 objetivos. . . . .	33
Figura 14 – Percentuais obtidos por AEMO nos intervalos de valores do <i>Makespan</i> em um DAG de 50 tarefas com execução de 3 objetivos. . . . .	34
Figura 15 – Percentuais obtidos por AEMO nos intervalos de valores do <i>Makespan</i> em um DAG de 50 tarefas com execução de 2 objetivos. . . . .	34
Figura 16 – Percentuais obtidos por AEMO nos intervalos de valores do <i>Load Balance</i> em um DAG de 50 tarefas com execução de 5 objetivos. . . . .	35
Figura 17 – Percentuais obtidos por AEMO nos intervalos de valores do <i>Load Balance</i> em um DAG de 50 tarefas com execução de 4 objetivos. . . . .	36
Figura 18 – Percentuais obtidos por AEMO nos intervalos de valores do <i>Load Balance</i> em um DAG de 50 tarefas com execução de 3 objetivos. . . . .	36
Figura 19 – Percentuais obtidos por AEMO nos intervalos de valores do <i>Load Balance</i> em um DAG de 50 tarefas com execução de 2 objetivos. . . . .	37
Figura 20 – Percentuais obtidos por AEMO nos intervalos de valores do <i>Flow Time</i> em um DAG de 50 tarefas com execução de 5 objetivos. . . . .	38
Figura 21 – Percentuais obtidos por AEMO nos intervalos de valores do <i>Flow Time</i> em um DAG de 50 tarefas com execução de 4 objetivos. . . . .	38
Figura 22 – Percentuais obtidos por AEMO nos intervalos de valores do <i>Flow Time</i> em um DAG de 50 tarefas com execução de 3 objetivos. . . . .	39



Figura 23 – Percentuais obtidos por AEMO nos intervalos de valores do <i>Communication Cost</i> em um DAG de 50 tarefas com execução de 5 objetivos. . . . .	40
Figura 24 – Percentuais obtidos por AEMO nos intervalos de valores do <i>Communication Cost</i> em um DAG de 50 tarefas com execução de 4 objetivos. . . . .	40
Figura 25 – Percentuais obtidos por AEMO nos intervalos de valores do <i>Waiting Time</i> em um DAG de 50 tarefas com execução de 5 objetivos. . . . .	41
Figura 26 – Tendência do melhor indivíduo por AEMO no DAG de 50 tarefas com redução de objetivos - Média Simples . . . . .	45
Figura 27 – Tendência do melhor indivíduo por AEMO no DAG de 100 tarefas com redução de objetivos - Média Simples . . . . .	45
Figura 28 – Tendência do melhor indivíduo por AEMO no DAG de 300 tarefas com redução de objetivos - Média Simples . . . . .	46
Figura 29 – Tendência do melhor indivíduo por AEMO no DAG de 50 tarefas com redução de objetivos - Média Harmônica . . . . .	48
Figura 30 – Tendência do melhor indivíduo por AEMO no DAG de 100 tarefas com redução de objetivos - Média Harmônica . . . . .	49
Figura 31 – Tendência do melhor indivíduo por AEMO no DAG de 300 tarefas com redução de objetivos - Média Harmônica . . . . .	49

# Lista de tabelas

Tabela 1 – Parâmetros utilizados na execução dos AEMOs. . . . .	31
Tabela 2 – Intervalo de valores obtidos para cada objetivo no DAG de 50 tarefas. .	32
Tabela 3 – Melhores objetivos encontrados por AEMO em um DAG de 50 tarefas na execução com 5 objetivos. . . . .	42
Tabela 4 – Melhores indivíduos com média simples encontrados por AEMO em um DAG de 50 tarefas. . . . .	43
Tabela 5 – Melhores indivíduos com média simples encontrados por AEMO em um DAG de 100 tarefas. . . . .	44
Tabela 6 – Melhores indivíduos com média simples encontrados por AEMO em um DAG de 300 tarefas. . . . .	44
Tabela 7 – Melhores indivíduos com média harmônica encontrados por AEMO em um DAG de 50 tarefas. . . . .	47
Tabela 8 – Melhores indivíduos com média harmônica encontrados por AEMO em um DAG de 100 tarefas. . . . .	47
Tabela 9 – Melhores indivíduos com média harmônica encontrados por AEMO em um DAG de 300 tarefas. . . . .	48
Tabela 10 – Hiper-Volume das melhores soluções encontradas no DAG de 50 tarefas.	50
Tabela 11 – Hiper-Volume das melhores soluções encontradas no DAG de 100 tarefas.	51
Tabela 12 – Hiper-Volume das melhores soluções encontradas no DAG de 300 tarefas.	51

# Lista de abreviaturas e siglas

ASF	<i>Achievement Scalarization Function</i>
AE	Algoritmos Evolutivo
AEMMD	Algoritmo Evolutivo Multiobjetivo com Múltiplas Dominâncias
AEMMO	Algoritmos Evolutivos Multi-Objetivo
AEMMT	Algoritmo Evolutivo Multi-Objetivo com Múltiplas Tabelas
DTLZ	<i>Deb-Thiele-Laumanns-Zitzler</i>
DTLZ1-4	<i>Deb-Thiele-Laumanns-Zitzler 4</i>
MSE	<i>Matching Scheduling Encoding</i>
NSGA	<i>Non-Dominated Sorting Genetic Algorithm</i>
NSGA-II	<i>Non-Dominated Sorting Genetic Algorithm II</i>
NSGA-III	<i>Non-Dominated Sorting Genetic Algorithm III</i>
OLE	<i>Ordered List Encoding</i>
PAES	<i>Pareto Archived Evolution Strategy</i>
PLE	<i>Processor List Encoding</i>
SPEA-II	<i>Strength Pareto Evolutionary Algorithm II</i>
TLE	<i>Topological List Encoding</i>
WFG	<i>Walking Fish Group</i>
MOEAD/D	<i>Multi-Objective Evolutionary Algorithm Based on Decomposition</i>

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>9</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>11</b>
2.1	Escalonamento de tarefas paralelas	11
2.2	Algoritmos evolutivos	13
2.3	Algoritmos evolutivos multi-objetivo	14
2.3.1	Algoritmo Evolutivo Multiobjetivo com Múltiplas Tabelas	15
2.3.2	Algoritmo Evolutivo Multi-Objetivo com Múltiplas Dominâncias	16
2.3.3	Non-dominated Sorting Genetic Algorithm II	16
2.3.4	Non-dominated Sorting Genetic Algorithm III	18
2.3.5	Diferenças entre o NSGA-II e o NSGA-III	20
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>21</b>
<b>4</b>	<b>DESENVOLVIMENTO</b>	<b>23</b>
4.1	Implementação	23
4.1.1	Representação dos Indivíduos	23
4.1.2	Recombinação e Mutação	24
4.1.3	Classificação dos Indivíduos em Fronteiras Não-Dominadas	25
4.2	Seleção dos Indivíduos	25
4.2.1	Normalização Adaptativa dos Indivíduos	26
4.2.2	<i>Reference Points</i> e Associação	26
4.2.3	Operação de Preservação de Nicho	28
<b>5</b>	<b>RESULTADOS</b>	<b>30</b>
5.1	Ambiente de Experimentos	30
5.2	Experimentos e Avaliação dos Resultados	30
5.2.1	Percentuais por AEMO nos intervalos dos objetivos	31
5.2.2	Melhores valores de objetivos encontrados pelos AEMOs	41
5.2.3	Melhores indivíduos encontrados pelos AEMOs	43
5.2.3.1	Média Simples	43
5.2.3.2	Média Harmônica	46
5.2.3.3	Hiper-Volume	49
<b>6</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b>	<b>53</b>

<b>REFERÊNCIAS</b> .....	<b>56</b>
--------------------------	-----------

# 1 Introdução

A adoção de sistemas de computação distribuída e processadores multi-core tem crescido significativamente nos últimos anos (SECARA, 2020), impulsionada pela demanda por execução eficiente de aplicações de alto desempenho e paralelizáveis. No entanto, a plena utilização destes sistemas requer a solução de desafios como a alocação eficiente de tarefas paralelas entre processadores computacionais em computação de alto desempenho (ROBERT, 2011). Assim, algoritmos de escalonamento são cruciais para otimizar a execução de tarefas.

Este trabalho aborda o problema de escalonamento de tarefas, onde as tarefas são representadas como grafos acíclicos direcionados (DAGs) com pesos, e as comunicações entre tarefas são representadas por arestas (ROBERT, 2011; SILVA, 2020). Consideramos um sistema de computação heterogêneo no qual o custo de execução de uma tarefa depende do processador ao qual ela é atribuída, e os processadores estão interconectados. O escalonamento de tarefas é um problema clássico  $\mathcal{NP}$ -Difícil (ROBERT, 2011). Diversas heurísticas e meta-heurísticas têm sido propostas (SILVA, 2020), geralmente focando em objetivos únicos, como o tempo de conclusão das tarefas (OMARA; ARAFA, 2010). O surgimento de sistemas multiprocessadores introduziu métricas relacionadas à utilização de recursos (SILVA, 2020), levando a conflitos entre os objetivos de escalonamento, como equilibrar custos de comunicação e cargas dos processadores.

Para enfrentar estes desafios, ferramentas de inteligência artificial, como algoritmos genéticos (DEB; KALYANMOY, 2001), têm sido bastante empregadas com o objetivo de encontrar a melhor solução para o problema de escalonamento de tarefas em múltiplos processadores, especialmente no contexto de sistemas distribuídos e computação de alto desempenho. Métodos baseados em dominância de Pareto, como algoritmos evolutivos multiobjetivo (MOEAs), são frequentemente preferidos. Embora eficazes para dois ou três objetivos, os MOEAs enfrentam dificuldades com mais de três (ISHIBUCHI et al., 2016). No entanto, é importante destacar que dificilmente um único algoritmo será superior em todos os cenários, uma vez que resultados diferentes podem surgir em diferentes contextos.

Neste contexto, o trabalho de Santos (2023) envolve a implementação e comparação de algoritmos existentes, mais especificamente, o NSGA-II (*Non-Dominated Sorting Genetic Algorithm II*), o AEMMT (Algoritmo Evolutivo Multi-Objetivo com Múltiplas Tabelas) de média simples e harmônica e o AEMMD (Algoritmo Evolutivo Multi-Objetivo com Múltiplas Dominâncias). Este estudo visa expandir a pesquisa de Santos (2023) com a implementação do algoritmo NSGA-III (*Non-Dominated Sorting Genetic Algorithm III*) e sua comparação com os demais algoritmos presentes no estudo anterior. Ao manter o

mesmo sistema, base de dados, linguagem de programação e parâmetros, é possibilitada uma comparação entre os algoritmos (AEMMT (média simples e harmônica), AEMMD, NSGA-II e NSGA-III), onde a eficácia de produzir a melhor solução será o principal critério de avaliação.

Os cinco objetivos estudados no trabalho também não mudaram, são eles: makespan, balanceamento de carga, flowtime, custo de comunicação e tempo de espera. Makespan refere-se ao tempo total necessário para completar todas as tarefas, enquanto o balanceamento de carga busca distribuir as tarefas de forma equilibrada entre os processadores, evitando sobrecargas. O flowtime é a soma dos tempos de conclusão de todas as tarefas, e o custo de comunicação diz respeito ao overhead gerado pela troca de informações entre os processadores. Finalmente, o tempo de espera avalia o período que as tarefas passam aguardando para serem processadas.

Uma ilustração clara destas variações é a distinção entre o NSGA-II e o NSGA-III. Enquanto um algoritmo pode ter critérios de otimização diversos, muitas vezes estes objetivos estão em conflito, aumentando a complexidade do processamento à medida que um critério é otimizado. O NSGA-III foi desenvolvido para cenários com múltiplos critérios de otimização, enquanto o NSGA-II é mais adequado para problemas com até dois critérios. Portanto, este trabalho visa observar e destacar as diferenças entre o NSGA-III e seu predecessor, assim como o AEMMT e o AEMMD.

Como mencionado anteriormente, este trabalho é uma extensão do estudo anterior de (SANTOS, 2023), que foi totalmente implementado em Java e disponibilizado em um repositório público<sup>1</sup>. Para tal, o desenvolvimento deste TCC foi realizado em três etapas: na primeira, buscou-se compreender os algoritmos já presentes no repositório; na segunda fase, abordou-se o entendimento do NSGA-III e suas diferenças em relação aos outros algoritmos; e, por fim, a terceira fase consistiu na implementação propriamente dita, seguindo o modelo e a arquitetura do código existente. A implementação do NSGA-III foi incorporada ao mesmo repositório, garantindo uma comparação o mais equitativa possível, mantendo constantes variáveis como máquina, linguagem, base de dados e outros.

---

<sup>1</sup> [agscheduling](#)

## 2 Fundamentação Teórica

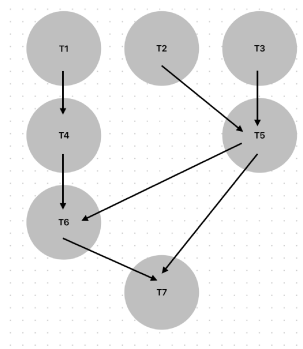
Este capítulo aborda a base teórica do estudo, explorando os algoritmos evolutivos multiobjetivos e o problema de escalonamento de tarefas. São apresentados os principais conceitos destes algoritmos, como NSGA-II e NSGA-III, que permitem otimizar múltiplos objetivos ao mesmo tempo. Além disso, é discutido o escalonamento de tarefas, um problema que busca distribuir tarefas entre múltiplos processadores de forma eficiente.

### 2.1 Escalonamento de tarefas paralelas

Com o propósito de simplificar e otimizar a resolução de problemas computacionais de alta complexidade, é viável fragmentá-los em subproblemas menores que podem ser processados simultaneamente. Neste contexto, o escalonamento de tarefas paralelas refere-se à divisão, distribuição, execução e coordenação de múltiplas tarefas de forma simultânea em um ambiente composto por vários processadores.

No entanto, esta tarefa está longe de ser trivial. A maioria das tarefas apresenta interdependências com uma ou mais partes, estabelecendo uma complexa rede de dependências, como ilustrado na Figura 1. Na ilustração, cada círculo é um processo e as setas são a interdependência entre eles, ou seja, o processo  $T5$  só pode iniciar quando os processos  $T2$  e  $T3$  estiverem concluídos. Além disso, é preciso lidar com tempos de execução variados para cada tarefa, o que aumenta a complexidade da sua organização.

Figura 1 – Grafo das dependências entre tarefas



Fonte: Autoria própria

O escalonamento de tarefas paralelas pode se tornar ainda mais complexo quando pensamos em uma otimização multi-objetivo, que nada mais é do que organizar a execução das tarefas com o intuito de otimizar a execução visando um ou mais critérios. Existem inúmeros critérios, sendo os utilizados neste estudo:



- **Makespan:** Refere-se ao tempo total necessário para completar todas as tarefas. É uma medida importante para avaliar a eficiência do escalonamento em ambientes de múltiplos processadores.
- **Load Balance:** Busca distribuir as tarefas de forma equilibrada entre os processadores, evitando que alguns processadores fiquem sobrecarregados enquanto outros permanecem subutilizados.
- **Flowtime:** É a soma dos tempos de conclusão de todas as tarefas. Este objetivo avalia a rapidez com que as tarefas são processadas em conjunto.
- **Communication Cost:** Diz respeito ao overhead gerado pela troca de informações entre os processadores. Overhead é o custo adicional em termos de tempo ou recursos computacionais necessários para gerenciar e coordenar a comunicação entre os processadores, que pode incluir latências de rede, tempos de sincronização e outros fatores que não contribuem diretamente para o processamento das tarefas.
- **Waiting Time:** Avalia o período que as tarefas passam aguardando para serem processadas. Reduzir o tempo de espera é essencial para aumentar a eficiência do escalonamento.

Há também diversos outros critérios que não foram contemplados neste estudo, a seguir tem-se dois exemplos de objetivos de sustentabilidade:

- **Consumo elétrico:** A velocidade dos processadores está relacionada diretamente com o consumo elétrico.
- **Temperatura:** Assim como o consumo elétrico, essa é uma métrica voltada a sustentabilidade e está diretamente relacionada com a velocidade do processador.

Conforme observado, uma ampla gama de métricas de otimização está disponível, abrangendo considerações como velocidade, sustentabilidade, confiabilidade e muitas outras. Neste contexto, a otimização multi-objetivo envolve a tentativa de organizar tarefas de forma a aprimorar simultaneamente um ou mais critérios. No entanto, como já mencionado, esta tarefa está longe de ser simples.

Frequentemente, os critérios de otimização entram em conflito, tornando a busca pela solução ideal um desafio de extrema complexidade. Um exemplo notável é a relação inversamente proporcional entre as métricas de velocidade e sustentabilidade: à medida que um código é executado mais rapidamente, ele consome mais energia e gera temperaturas mais elevadas.

Com o propósito de buscar soluções, uma ampla gama de estudos científicos surgiram, categorizados em dois paradigmas (SANTOS, 2023): os determinísticos e os não-determinísticos. A título de ilustração dos métodos determinísticos, podemos mencionar o escalonamento por lista (ARABNEJAD; BARBOSA, 2014).

Por sua vez, no âmbito dos métodos não-determinísticos, houve significativos avanços com a introdução dos algoritmos evolutivos (OMARA; ARAFA, 2010), que atualmente representam a abordagem mais amplamente adotada para resolver este problema (SILVA, 2020).

## 2.2 Algoritmos evolutivos

Algoritmos evolutivos (AEs) são uma classe de algoritmos de otimização que busca simular a evolução natural. Eles foram desenvolvidos com o intuito de otimizar problemas complexos com múltiplos parâmetros, além de serem amplamente utilizados na área de otimização, aprendizado de máquina e modelagem de sistemas complexos. Seu surgimento está associado aos trabalhos de Holland (1962), que é um dos principais responsáveis por popularizar os AEs na década de 1970 (SANTOS, 2023).

Nos AEs cada solução para o problema é representada como um indivíduo, e cada indivíduo possui uma nota de aptidão, que nada mais é do que um número que representa o quão bom é o desempenho da solução se acordo com os parâmetros passados. De maneira similar a evolução natural, a população de indivíduos passa por uma série de variações ao longo das gerações, podendo ser por recombinação, seleção, mutação, migração ou pressão evolutiva.

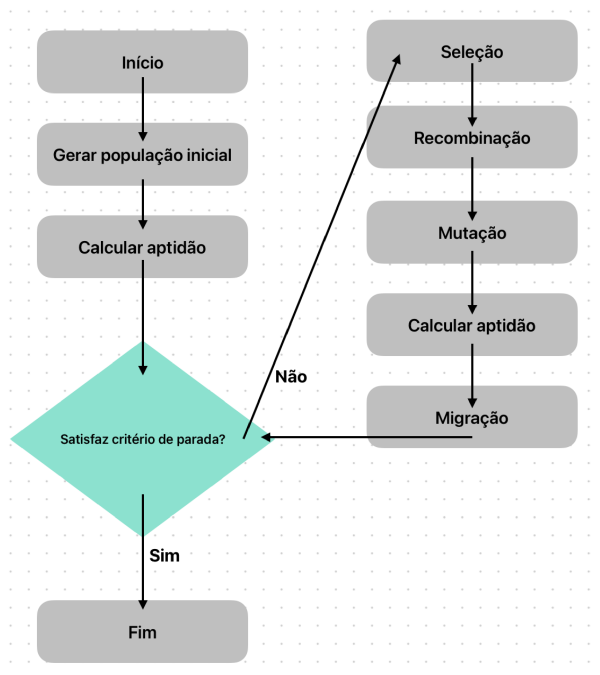
- **Recombinação:** É a mistura do material genético de dois indivíduos (pais). É selecionado um ponto aleatório e o material genético (na maioria das vezes uma cadeia de bits) é misturado a partir daquele ponto.
- **Seleção:** Simulando a seleção natural, indivíduos com maior aptidão terão mais chances de reproduzir e sobreviver, propagando seu material genético para futuras gerações. Além disso, ao juntar o nível de aptidão com uma probabilidade de ser ou não selecionado é possível evitar convergências prematuras nas soluções.
- **Mutação:** Através de uma série de escolhas aleatórias é possível gerar mudanças não enviesadas nas soluções, ajudando a manter a diversidade genética da população. A mutação pode ser uma alteração em um único bit por exemplo, de maneira que não gere caracteres inválidos.
- **Migração:** Com o objetivo de controlar o tamanho da população pode ser empregado diversas estratégias de controle, a mais famosa delas sendo as estratégias

elitistas, nas quais indivíduos com baixa aptidão são substituídos por outros de ranking mais alto.

- **Pressão evolutiva:** É a probabilidade de que indivíduos com rankings mais altos sejam escolhidos. Quanto maior a pressão evolutiva, maior será o elitismo. É importante atenção nesta parte, pois caso a pressão seja muito alta o algoritmo pode chegar em uma solução prematura.

No início do processo, uma população inicial é gerada e avaliada quanto à sua aptidão. Em seguida, o algoritmo opera em um ciclo ao longo de várias gerações. Durante cada iteração, indivíduos são selecionados para recombinação, com seus materiais genéticos sendo combinados de acordo com a estratégia escolhida. As novas soluções resultantes são submetidas à mutação, podendo sofrer alterações aleatórias em seus genomas. Por fim, o AE pode aplicar uma estratégia de migração para determinar quais indivíduos persistirão nas gerações subsequentes. O fluxo completo do processo é ilustrado na Figura 2.

Figura 2 – Fluxograma de um AE



Fonte: Autoria própria

## 2.3 Algoritmos evolutivos multi-objetivo

Os Algoritmos Evolutivos Multi-Objetivo (AEMOs) são uma classe de algoritmos de otimização que se concentram na resolução de problemas de otimização multi-objetivo.

Em vez de encontrar uma única solução ótima, estes algoritmos buscam encontrar um conjunto de soluções que representam um compromisso entre múltiplos objetivos, considerando que estes objetivos muitas vezes estão em conflito (DEB; KALYANMOY, 2001). A frente de Pareto é um conceito fundamental em AEMO, representando o conjunto de soluções não dominadas, ou seja, soluções em que não é possível melhorar um objetivo sem piorar outro.

Os AEMOs têm diversas aplicações em campos como engenharia, economia, logística, planejamento urbano e ambiental, entre outros. Uma das características distintivas dos AEMOs é sua capacidade de lidar com problemas do mundo real que envolvem compromissos complexos e variáveis inter-relacionadas. Eles são particularmente úteis quando os objetivos não podem ser agregados em um único critério de desempenho (DEB; KALYANMOY, 2001).

Na década de 1990, a área de AEMO viu um grande avanço com a introdução do NSGA (*Non-dominated Sorting Genetic Algorithm*) por Srinivas e Deb (1995). Este algoritmo inovou ao apresentar a ideia de ordenação não dominada para classificar soluções candidatas. Além disto, outros algoritmos, como o PAES (*Pareto Archived Evolution Strategy*) (KNOWLES; CORNE, 1999) também surgiram nessa década.

Nos anos 2000, o desenvolvimento de AEMOs continuou a ganhar destaque. Em 2002, Deb e sua equipe apresentaram o NSGA-II (DEB et al., 2002), uma versão aprimorada do NSGA, que introduziu seleção baseada em dominância e o conceito de elitismo. Neste período, o SPEA2 (*Strength Pareto Evolutionary Algorithm 2*) (COELLO; LAMONT; VELDHUIZEN, 2007) também foi desenvolvido, melhorando ainda mais a resolução de problemas multi-objetivo.

O foco deste trabalho se encontra no NSGA-III, uma evolução dos AEMOs que se concentra em otimização de múltiplos objetivo (tipicamente, mais que três). Trata-se de uma extensão direta do NSGA-II e será detalhado na próxima seção.

### 2.3.1 Algoritmo Evolutivo Multiobjetivo com Múltiplas Tabelas

O Algoritmo Evolutivo Multiobjetivo com Múltiplas Tabelas (AEMMT) é um AEMO proposto inicialmente para predição de estrutura de proteínas (BRASIL, 2012; BRASIL; DELBEM; SILVA, 2013), e trata-se de uma adaptação do *Multi-Objective Evolutionary Algorithm in Tables* (MEAT), proposto para problemas de dois objetivos (SANTOS et al., 2010; GABRIEL; MELO; DELBEM, 2012). Ele organiza as soluções em tabelas, cada uma representando combinações específicas dos objetivos do problema. No caso de quatro objetivos, por exemplo, são criadas 15 tabelas mais uma adicional para armazenar indivíduos não-dominados. A população inicial é gerada de modo a preencher as tabelas com indivíduos distintos, e cada solução é avaliada e inserida em uma tabela

apenas se superar a pior solução já presente, considerando a média aritmética dos objetivos. A tabela dos não-dominados tem regras específicas para inclusão de indivíduos e pode ter tamanho diferenciado das demais.

Durante as gerações, o AEMMT seleciona duas tabelas por torneio e realiza cruzamentos para gerar novos descendentes, que passam pelo mesmo processo de avaliação e inserção nas tabelas. A pontuação das tabelas é zerada periodicamente para evitar favoritismo em seleções futuras. Como o algoritmo gera apenas um descendente por geração, ele requer mais iterações para atingir um número equivalente de soluções avaliadas em comparação a outros algoritmos evolutivos. Ao final, o conjunto de soluções não-dominadas é retornado como resultado, representando as melhores soluções para o problema multiobjetivo.

### 2.3.2 Algoritmo Evolutivo Multi-Objetivo com Múltiplas Dominâncias

O Algoritmo Evolutivo Multiobjetivo com Múltiplas Dominâncias (AEMMD) é uma evolução do AEMMT que introduz mudanças na formação e pontuação das tabelas (LAFETÁ *et al.*, 2018). No AEMMD, utiliza-se o conceito de dominância em vez da média aritmética para determinar a entrada de um indivíduo em uma tabela. Isso significa que um indivíduo só é aceito em uma tabela se não for dominado por nenhum outro já presente, o que elimina tabelas dedicadas a objetivos individuais e a tabela específica de não-dominância. Para problemas com quatro objetivos, por exemplo, são criadas 11 tabelas representando combinações de 2 a 4 objetivos. Além disso, as tabelas do AEMMD não possuem tamanho fixo, e sua pontuação é incrementada sempre que recebem um novo indivíduo, baseando-se na convergência em vez de contribuição.

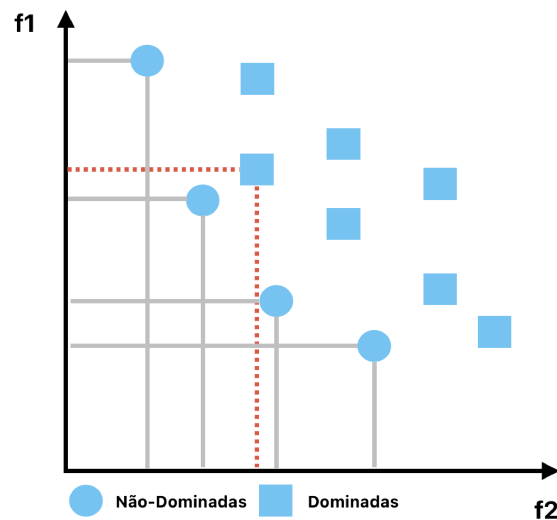
O AEMMD mantém o processo de seleção, cruzamento e geração de descendentes semelhante ao AEMMT. Cada geração seleciona duas tabelas para o cruzamento, e o descendente resultante é avaliado para entrada nas tabelas conforme o critério de dominância. Indivíduos dominados por um novo integrante são removidos. Como as gerações produzem apenas um descendente, o algoritmo requer mais iterações para avaliar um número comparável de soluções em relação a outros AGs. Ao final, o conjunto das melhores soluções é extraído das tabelas, representando os indivíduos não-dominados.

### 2.3.3 Non-dominated Sorting Genetic Algorithm II

O NSGA-II, proposto por Deb *et al.* (2002), é um algoritmo evolutivo multiobjetivo amplamente utilizado devido à sua eficiência e simplicidade na resolução de problemas de otimização com múltiplos objetivos. O algoritmo introduziu várias melhorias em relação ao seu predecessor, o NSGA, tornando-o uma escolha popular para muitos problemas de otimização.

A estrutura básica do NSGA-II envolve três componentes principais: ordenação não dominada, *crowding distance* e um operador elitista. A ordenação não dominada classifica a população em diferentes frentes de Pareto, onde a primeira frente contém as soluções que não são dominadas por nenhuma outra, a segunda frente contém as soluções que são dominadas apenas pelas da primeira frente, e assim por diante. Como demonstrado na Figura 3.

Figura 3 – Exemplo de soluções não-dominadas.



Fonte: Autoria própria

O *crowding distance* é um mecanismo que assegura a diversidade das soluções dentro de cada frente. Ele calcula uma medida de densidade de soluções ao redor de cada ponto, priorizando soluções que estão em regiões menos povoadas do espaço de soluções. Este processo evita que o algoritmo acabe convergindo prematuramente para uma única solução, mantendo uma diversidade adequada.

Por fim, o operador elitista garante que as melhores soluções encontradas até o momento sejam preservadas na próxima geração. Isso é feito combinando a população atual com a nova geração gerada, e então aplicando a ordenação não dominada e o *crowding distance* para selecionar as melhores soluções.

O NSGA-II se destaca por seu desempenho em problemas com dois ou três objetivos, apresentando uma boa combinação de diversidade e convergência. No entanto, conforme o número de objetivos aumenta, o algoritmo enfrenta dificuldades em manter a diversidade das soluções, levando ao desenvolvimento de abordagens mais avançadas como o NSGA-III.

### 2.3.4 Non-dominated Sorting Genetic Algorithm III

O NSGA-III é uma extensão da família de algoritmos evolutivos multi-objetivo, desenvolvida para lidar especificamente com problemas que envolvem múltiplos critérios de otimização. Este algoritmo, proposto por [Deb e Jain \(2014\)](#), representa uma evolução do NSGA-II, buscando superar limitações específicas associadas à otimização em cenários com mais de dois objetivos.

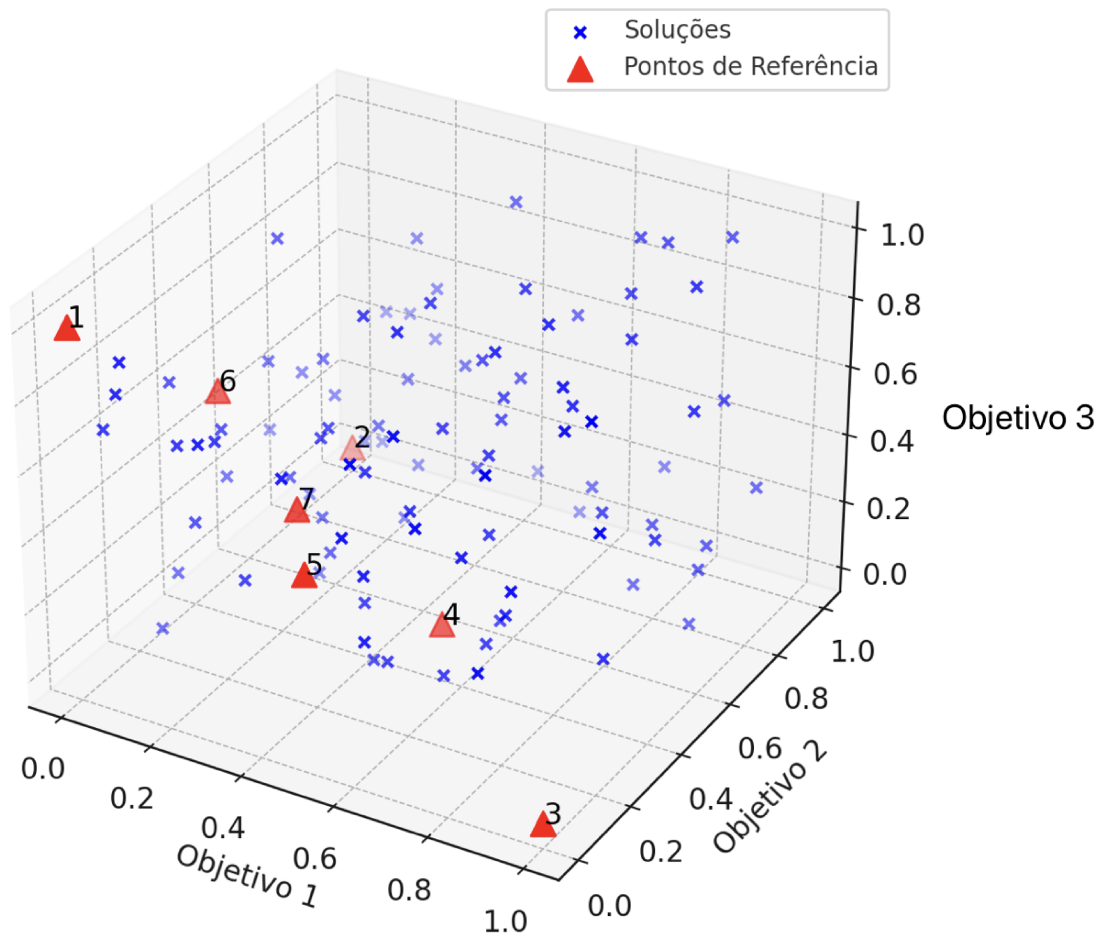
A principal motivação por trás do desenvolvimento do NSGA-III foi a necessidade de proporcionar uma solução mais eficaz para problemas com múltiplos critérios, nos quais a otimização de um objetivo muitas vezes entra em conflito com a otimização de outros. Enquanto o NSGA-II é eficiente para problemas bidimensionais, o NSGA-III se destaca em situações mais complexas que envolvem uma diversidade maior de objetivos.

A abordagem fundamental do NSGA-III reside no aprimoramento da técnica de ordenação não dominada (*non-dominated sorting*), um componente crítico em algoritmos evolutivos multi-objetivo. A ordenação não dominada é essencial para classificar as soluções candidatas em frentes de Pareto, onde uma solução não domina outra se não for pior em nenhum critério e for melhor em pelo menos um critério. O NSGA-III aprimora este processo, permitindo uma classificação mais eficiente e uma diversificação aprimorada das soluções.

Um dos diferenciais notáveis do NSGA-III é sua capacidade de lidar com um número variável de objetivos de maneira eficaz. Enquanto muitos algoritmos multi-objetivo enfrentam dificuldades quando o número de objetivos aumenta, o NSGA-III incorpora estratégias inovadoras de seleção e ordenação para garantir a adaptabilidade em cenários de otimização com muitos objetivos.

Para alcançar isto, o NSGA-III introduz o conceito de pontos de referência. Estes pontos são distribuídos uniformemente ao longo de uma hiperplano, representando as direções preferenciais de otimização. Cada solução na população é então associada a um destes pontos de referência, permitindo uma diversificação controlada das soluções ao longo das várias direções de otimização. Esta associação é crítica para manter uma distribuição equilibrada de soluções em cenários com muitos objetivos.

Figura 4 – Ilustração do Funcionamento do NSGA-III.



Fonte: Autoria Própria

Na Figura 4, as soluções (representadas por pontos azuis) são as diversas soluções candidatas para um problema multi-objetivo. Os pontos de referência (representados por triângulos vermelhos) são usados pelo NSGA-III para garantir a diversidade das soluções ao longo das direções preferenciais de otimização. Cada solução é associada a um desses pontos de referência para promover uma distribuição equilibrada e eficaz das soluções no espaço de objetivos. A imagem de exemplo representa um problema de três objetivos.

Além disso, o NSGA-III incorpora um operador de nicho ambiental, que promove uma distribuição mais uniforme das soluções não dominadas. Este operador de nicho utiliza a associação aos pontos de referência para identificar áreas menos exploradas do espaço de soluções e direcionar a busca para essas regiões. Isso evita a convergência prematura para soluções específicas e favorece a busca por soluções mais diversas.



### 2.3.5 Diferenças entre o NSGA-II e o NSGA-III

As principais diferenças entre o NSGA-II e o NSGA-III residem nas suas capacidades de lidar com problemas de otimização com múltiplos objetivos e nas técnicas utilizadas para manter a diversidade das soluções. O NSGA-II é eficiente para problemas com dois ou três objetivos, mas enfrenta dificuldades quando o número de objetivos aumenta devido à perda de diversidade nas soluções. Em contraste, o NSGA-III foi projetado especificamente para lidar com um grande número de objetivos, utilizando pontos de referência para manter a diversidade das soluções.

Uma distinção significativa entre os dois algoritmos é a utilização de pontos de referência no NSGA-III. Enquanto o NSGA-II não utiliza estes pontos, mantendo a diversidade principalmente através do *crowding distance*, o NSGA-III introduz pontos de referência para representar direções preferenciais de otimização. As soluções são associadas a estes pontos, promovendo uma distribuição equilibrada das soluções em cenários com muitos objetivos.

Além disto, o NSGA-III incorpora um operador de nicho ambiental que promove uma distribuição mais uniforme das soluções não dominadas. Este operador de nicho utiliza a associação aos pontos de referência para identificar áreas menos exploradas do espaço de soluções e direcionar a busca para essas regiões. Isto evita a convergência prematura para soluções específicas e favorece a diversidade. Em contraste, o NSGA-II não incorpora um operador de nicho ambiental específico.

No que diz respeito à técnica de ordenação não dominada, ambos os algoritmos a utilizam para classificar as soluções em frentes de Pareto. No entanto, o NSGA-III aperfeiçoa essa técnica, permitindo uma classificação mais eficiente e uma diversificação aprimorada das soluções em cenários com muitos objetivos. Isto garante que o NSGA-III possa lidar melhor com a complexidade adicional associada a problemas com um grande número de objetivos.

Em resumo, enquanto o NSGA-II é uma ferramenta poderosa para problemas de otimização de até três objetivos, o NSGA-III oferece avanços significativos para cenários de otimização mais complexos com múltiplos objetivos. As técnicas inovadoras do NSGA-III para manter a diversidade e a eficiência das soluções o tornam uma escolha superior para problemas de otimização de alta dimensionalidade.

## 3 Trabalhos Relacionados

Este TCC é uma extensão do trabalho de mestrado de Santos (2023), cujo objetivo é avaliar o desempenho dos algoritmos evolutivos na resolução do problema de escalonamento de tarefas. Especificamente, buscou-se analisar como os algoritmos NSGA-II, AEMMT e AEMMD se comportam diante de diferentes configurações de problemas, considerando variados números de tarefas, processadores e objetivos.

Os algoritmos foram adaptados para o problema de escalonamento de tarefas, levando em consideração as particularidades do contexto. Utilizando os mesmos parâmetros e dados, foram realizadas simulações em um conjunto de problemas variados. A avaliação comparativa baseou-se em métricas relevantes, analisando o desempenho de cada algoritmo diante das diferentes condições experimentais.

Os resultados revelaram que, entre os algoritmos testados, o AEMMT apresentou os melhores resultados, seguido pelo AEMMD e, por fim, pelo NSGA-II. Essas conclusões fornecem compreensões valiosas sobre a eficácia desses algoritmos em cenários específicos de escalonamento de tarefas.

Este trabalho estabeleceu uma base sólida para a implementação de outros algoritmos, como o NSGA-III. Utilizando os mesmos parâmetros e dados, a inclusão do NSGA-III permite uma análise abrangente e comparativa entre os quatro algoritmos evolutivos. Desta forma, é possível entender como o NSGA-III se destaca ou se diferencia em relação aos algoritmos previamente implementados, contribuindo para uma compreensão mais profunda das melhores abordagens para o problema de escalonamento de tarefas em sistemas computacionais.

Outro trabalho relevante no desenvolvimento deste TCC é o de Silva (2020), que propõe uma revisão sistemática da literatura focada nas características dos algoritmos genéticos aplicados ao desafio do escalonamento de tarefas em sistemas multiprocessados. O objetivo principal consiste em investigar e sintetizar os avanços nesse campo ao longo das décadas, abrangendo o período de 1990 a 2018. Para atingir tal propósito, foi desenvolvido um protocolo de pesquisa, delineando as fases de execução da revisão e estabelecendo critérios claros para a inclusão ou exclusão de estudos.

Ao concluir a revisão, identificou-se 13 representações de cromossomos utilizadas nos algoritmos genéticos estudados. Além disso, foram catalogados 78 operadores genéticos distintos, proporcionando uma visão abrangente das abordagens utilizadas nesse contexto. A análise incluiu também a sumarização das funções objetivo empregadas, acrescentando camadas de compreensão à complexidade do problema.

Os resultados mais marcantes emergem da comparação entre quatro codificações diferentes (TLE, MSE, OLE e PLE) por meio de métricas cruciais como makespan, flow-time e load balance. A análise de dispersão revelou um desempenho superior na representação OLE, enquanto TLE, MSE e PLE demonstraram equivalência sob as métricas consideradas. Estes resultados não apenas contribuem para uma compreensão mais profunda do problema de escalonamento de tarefas em sistemas multiprocessados, mas também oferecem revelações valiosas que podem ajudar na tomada de decisões e orientar novas pesquisas na área.

Relacionando-se a este TCC, o trabalho de [Silva \(2020\)](#) oferece uma base sólida para a compreensão das estratégias e desafios associados aos algoritmos genéticos no contexto específico do escalonamento de tarefas em sistemas multiprocessados.

Finalmente, o trabalho de [Ishibuchi et al. \(2016\)](#) visa realizar uma avaliação comparativa de desempenho entre os algoritmos genéticos NSGA-II e NSGA-III. A relevância desse estudo reside no papel fundamental do NSGA-III como algoritmo de referência para a avaliação de novos algoritmos evolutivos voltados para a otimização de múltiplos objetivos. Contudo, devido à indisponibilidade do código-fonte do NSGA-III, diferentes implementações são frequentemente utilizadas em diversos estudos, gerando uma lacuna indesejada. Além disso, a comparação convencional concentra-se em problemas específicos, como DTLZ e WFG (problemas projetados para testar algoritmos de otimização multi-objetivo em cenários complexos), deixando a eficácia do NSGA-III em uma variedade mais ampla de problemas multi-objetivo ainda incerta.

O método adotado pelos autores compreende a análise de desempenho do NSGA-III em comparação com o NSGA-II, utilizando quatro tipos distintos de problemas multi-objetivo, cada um com 3-10 objetivos: problemas DTLZ1-4, suas variantes de maximização, problemas de minimização de distância e problemas de mochila com objetivos correlacionados. Duas implementações diferentes de NSGA-III são utilizadas para avaliar a consistência dos resultados entre elas ([ISHIBUCHI et al., 2016](#)).

Os resultados destacam que o NSGA-III não supera consistentemente o NSGA-II, mesmo em problemas com até dez objetivos. A comparação revela que o desempenho desses algoritmos não depende apenas do número de objetivos, mas também do tipo de problema. Essa informação é particularmente valiosa para o este TCC, que se concentra na implementação do NSGA-III. O estudo indica que o desempenho relativo entre o NSGA-II e o NSGA-III é sensível à natureza específica do problema, fornecendo uma perspectiva crucial para a compreensão dos fatores que influenciam o desempenho destes algoritmos em cenários de otimização multi-objetivo. Estes resultados orientarão a abordagem e as expectativas durante a implementação do NSGA-III no projeto.

## 4 Desenvolvimento

Este estudo tem como objetivo implementar o NSGA-III que é o sucessor direto do NSGA-II ao conjunto de AEMOs presentes no trabalho de Santos (2023). Existe grande semelhança entre os dois algoritmos, basicamente toda a parte de combinação e mutação se mantém inalterada, sua diferença começa na seleção dos indivíduos.

Enquanto o NSGA-II emprega um método de classificação baseado na distância de aglomeração (crowding distance) para manter a diversidade genética na população, o NSGA-III utiliza um método de pontos de referência e nicho para manter a diversidade nas fronteiras de Pareto, o que o torna mais adequado para problemas com muitos objetivos. Enquanto o NSGA-II foi feito pra tratar problemas multi-objetivos de até 2 critérios de otimização, o NSGA-III consegue lidar com um número indeterminado de objetivos. Neste trabalho foi abordado os objetivos, *makespan*, *load balance*, *flowtime*, *communication cost* e *waiting time*.

A implementação do NSGA-III foi baseada no código da biblioteca JMetal (NEBRO; DURILLO; VERGNE, 2015), escrita em Java. Essa biblioteca é um *framework* para otimização multi-objetivos com meta-heurísticas, projetada e desenvolvida por pesquisadores para uso em suas respectivas áreas.

### 4.1 Implementação

#### 4.1.1 Representação dos Indivíduos

A estrutura dos indivíduos foi mantida inalterada do trabalho base (SANTOS, 2023). Cada membro da população é representado pelo sequenciamento e pelo mapeamento das tarefas de uma grafo acíclico dirigido (DAG). O sequenciamento representa a ordem em que as tarefas serão processadas e o mapeamento representa o índice dos processadores em que as tarefas são atribuídas a ele, sempre respeitando suas relações de precedência.

Figura 5 – Representação da codificação do indivíduo.

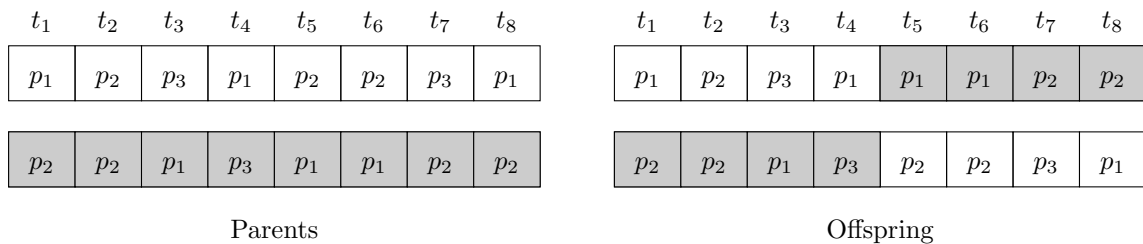
Sequenciamento								Mapeamento							
1	2	3	4	5	6	7	8	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$
$t_2$	$t_1$	$t_3$	$t_5$	$t_4$	$t_7$	$t_6$	$t_8$	$p_1$	$p_2$	$p_1$	$p_1$	$p_2$	$p_2$	$p_1$	$p_2$

### 4.1.2 Recombinação e Mutação

Como citado anteriormente, o processo de recombinação e mutação do NSGA-III se mantém o mesmo do NSGA-II. Na recombinação, os pais e o método de recombinação são selecionados aleatoriamente. Os métodos podem ser o *Crossover Map* e o *Order Crossover*.

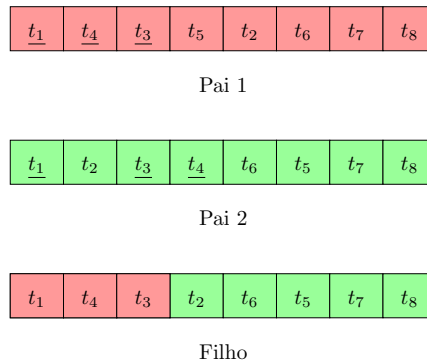
- **Crossover Map:** Neste método é gerado um ponto de corte aleatório (no caso do exemplo  $pc = 4$ ). Esse ponto de corte altera exclusivamente o vetor de mapeamento do indivíduo, gerando dois novos filhos. O primeiro filho mantém o vetor de sequenciamento do pai um e o segundo filho do pai dois, demonstrado na Figura 6.
- **Order Crossover:** Também gera-se um ponto de corte aleatório, porém agora alterando apenas o vetor de sequenciamento do indivíduo, mudando assim a ordem na qual as tarefas serão executadas no processador. Nesse método de recombinação apenas um indivíduo é gerado a partir dos dois pais, como exemplificado na Figura 7.

Figura 6 – Exemplo de crossover denominado Crossover Map.



Fonte: Santos (2023)

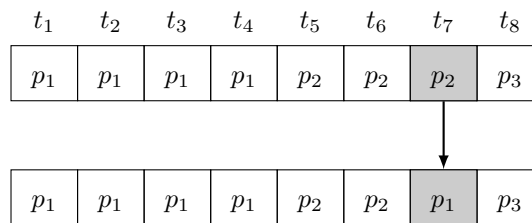
Figura 7 – Exemplo de crossover denominado Order Crossover.



Fonte: Santos (2023)

No processo de mutação a mudança ocorre apenas no vetor de mapeamento, e tem como objetivo alterar o índice em que uma tarefa é executado. Esta alteração é feita para gerar uma gama maior de diversidade nos indivíduos, e é feita de modo a não gerar dados inconsistentes, sempre respeitando a ordem de precedência das tarefas. A mutação de um indivíduo é representada na Figura 8.

Figura 8 – Exemplo de mutação aplicada ao indivíduo.



Fonte: Santos (2023)

### 4.1.3 Classificação dos Indivíduos em Fronteiras Não-Dominadas

Após todo esse processo, a população de pais e filhos são combinadas a fim de serem classificadas de acordo com as Frentes de Pareto, também conhecidas como Fronteiras de Pareto. Aqui está presente a primeira diferença entre os algoritmos do NSGA-II e NSGA-III. Por mais que o NSGA-II também faça uma classificação dos indivíduos em fronteiras não-dominadas, a classificação feita pelo NSGA-III é mais robusta, dividindo a população em mais de doze Fronteiras, enquanto o NSGA-II chega em apenas cinco.

Esta divisão em uma quantidade maior de Fronteiras de Pareto é crucial para o funcionamento do NSGA-III, e com isso evitar uma convergência prematura da população em uma solução sub-ótima. Ao associar cada indivíduo a uma fronteira, gera-se um *Rank* que é atribuído ao indivíduo, de modo que aqueles presentes nas fronteiras não-dominadas recebem o *Rank* 1.

## 4.2 Seleção dos Indivíduos

Após a união e classificação da população de pais e filhos em Fronteiras de Pareto, é iniciado o processo de seleção, o qual irá determinar quais indivíduos irão permanecer na próxima geração. Nesse momento, a população consiste nos pais (população no início da geração) mais os filhos (gerados no processo de recombinação), essa união resulta em uma população duas vezes maior que a população inicial. Sendo assim, se faz necessário selecionar uma quantidade de indivíduos que seja igual a quantidade inicial.

Para isso, selecionamos todas as fronteiras até que a quantidade atinja ou ultrapasse a quantidade de indivíduos no início do problema. Se a quantidade for a mesma, todos os indivíduos foram selecionados e inicia-se a próxima geração. Caso contrário, é aplicado um processo baseado na associação dos indivíduos a pontos de referência pré-definidos em um hiperplano. A população selecionada até o momento passa por um processo de normalização, em que seus objetivos são convertidos em pontos em um hiperplano  $M$  dimensional, em que  $M$  é a quantidade de objetivos do problema.

### 4.2.1 Normalização Adaptativa dos Indivíduos

Primeiramente, os *ideal points* são determinados encontrando o menor valor de cada um dos objetivos do problema. Após isso, todos os valores são traduzidos, ou seja, subtrai-se o respectivo *ideal point* de cada um dos objetivos, criando um vetor no qual o menor valor é sempre zero.

Tendo todos os valores traduzidos, é preciso calcular os *extreme points* para cada objetivo. Esses pontos nada mais são que uma solução que minimiza a função ASF (*achievement scalarization function*) dada por:

$$\text{ASF}(\mathbf{x}, \mathbf{w}) = \max_{i=1, \dots, M} \frac{f'_i(\mathbf{x})}{w_i}, \forall x \in S_t \quad (4.1)$$

Sendo  $i$  o objetivo,  $w$  a direção do eixo (zeros serão substituídos por  $10^{-6}$ ),  $f'_i(\mathbf{x})$  o valor traduzido do respectivo objetivo do indivíduo,  $x$  a solução e  $S_t$  a população. Isso irá resultar em um vetor de pontos extremos do objetivo. Esses  $M$  vetores são então usados para construir um hiperplano linear de  $M$  dimensões, aonde  $M$  é a quantidade de objetivos. Só então as interseções dos eixos dos objetivos e o hiperplano podem ser calculados como mostra a Figura 9.

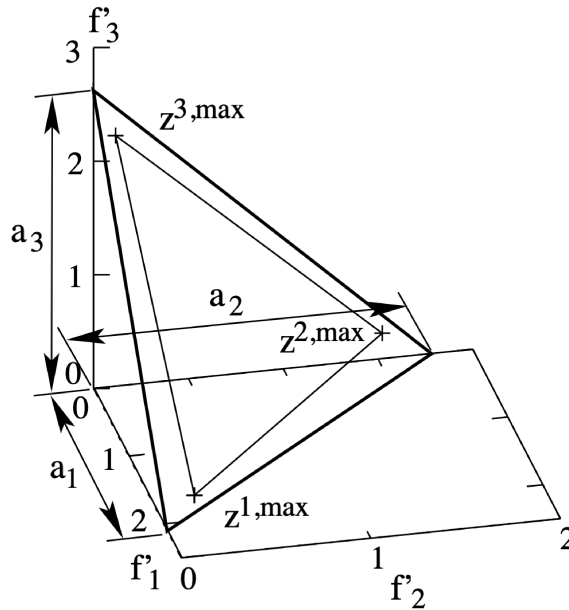
Tendo os valores dos objetivos traduzidos, os *extreme points* e as interseções nos eixos dos objetivos, é possível calcular os valores normalizados das soluções através da equação 4.2. Após normalizadas, elas irão constituir um ponto no hiperplano para serem associadas a um *reference point*.

$$f_i^n(\mathbf{x}) = \frac{f'_i(\mathbf{x})}{a_i - z_i^{\min}} = \frac{f_i(\mathbf{x}) - z_i^{\min}}{a_i - z_i^{\min}}, \forall i \in 1, 2, \dots, M \quad (4.2)$$

### 4.2.2 Reference Points e Associação

Como comentado anteriormente, o NSGA-III implementado usa pontos de referência pré-definidos para garantir a diversidade dos indivíduos da população. Além disso os pontos de referência também permitem uma melhor distribuição das soluções pela Fronteira de Pareto.

Figura 9 – Exemplo do processo de computação das interseções para criação de um hiperplano de três objetivos.



Fonte: Deb e Jain (2014)

Qualquer estrutura de pontos de referência pode ser usada. Neste trabalho escolheu-se um método que coloca os pontos em um hiperplano normalizado, o qual é igualmente inclinado em todos os eixos dos objetivos e intercepta cada um deles em um. Além disso, a quantidade de pontos de referência pode ser calculado através do coeficiente binomial dado pela equação 4.3, onde  $H$  é a quantidade de pontos gerados. Um exemplo dos pontos de referência em um hiperplano pode ser observado na Figura 10.

O algoritmo pode se tornar computacionalmente intensivo, especialmente conforme aumenta o número de objetivos e pontos de referência, mas é mais eficaz para distribuir soluções em problemas de alta dimensionalidade. Neste estudo optou-se por usar  $p = 6$ , gerando sete pontos nos problemas de dois objetivos e indo até duzentos e dez pontos nos problemas de cinco objetivos.

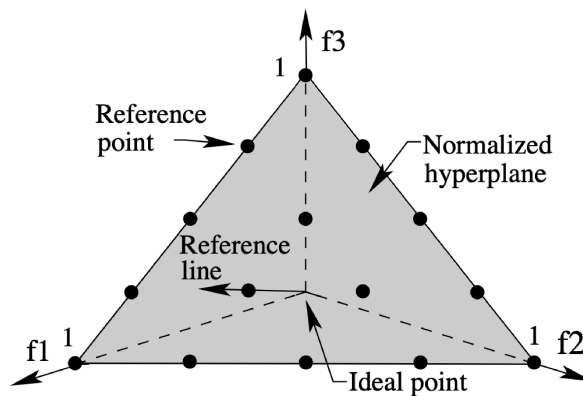
$$H = \binom{M + p - 1}{p} \quad (4.3)$$

Após ter em mãos os pontos de referência no hiperplano e as soluções normalizadas presentes nesse mesmo espaço, são criadas as linhas de referência. Essas linhas são obtidas após juntar um ponto de referência com a origem, gerando uma linha que atravessa o hiperplano.

A partir desse momento, é calculado a distância perpendicular entre as soluções e as linhas de referência geradas. A linha que possuir a menor distância é então associada



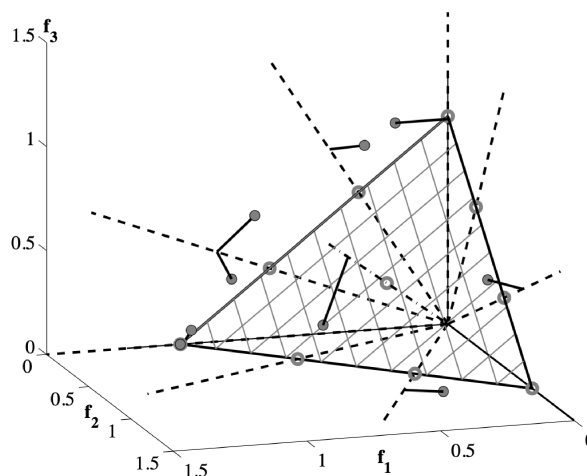
Figura 10 – Pontos de referência mostrados em um hiperplano normalizado de três objetivos com  $p = 4$



Fonte: Deb e Jain (2014)

ao respectivo indivíduo da população. Uma demonstração do processo de associação pode ser observado na Figura 11.

Figura 11 – Ilustração do processo de associação de uma solução com a linha de referência mais próxima.



Fonte: Deb e Jain (2014)

### 4.2.3 Operação de Preservação de Nicho

A operação de preservação de nicho é uma técnica usada para manter a diversidade genética dentro da população. Implementou-se um mecanismo onde, em cada geração, as soluções são escolhidas não apenas com base em sua proximidade aos pontos de referência, mas também considerando sua contribuição à diversidade no espaço objetivo.

Caso a seleção levasse em consideração apenas a distância perpendicular da solução ao ponto de referência, o algoritmo acabaria por selecionar várias soluções de um mesmo

ponto, ocasionando em uma baixa diversidade. Deste modo, a operação de preservação de nicho tenta dividir as soluções de uma maneira mais igualitária entre todos os pontos. Além disso, também aplica-se randomizações em suas escolhas, aumentando ainda mais a diversidade e evitando uma convergência prematura da população em uma solução sub-ótima.

## 5 Resultados

Neste capítulo, serão apresentados os resultados obtidos com o algoritmo NSGA-III em comparação com os outros AEMOs.

### 5.1 Ambiente de Experimentos

Este estudo é a continuação do trabalho de Santos (2023). Deste modo, foi adicionado no mesmo código o NSGA-III, a fim de criar uma comparação um pra um entre todos os AEMOs. O algoritmo foi desenvolvido em Java e o processador utilizado foi o Apple M2. Tanto o código fonte quanto os resultados gerados podem ser obtidos no seguinte repositório [agscheduling](#).

A fim de manter a comparação mais próxima possível do trabalho anterior, as *seeds* utilizadas foram as mesmas, assim como o número de objetivos (2, 3, 4 ou 5) e o número de processadores (2, 4, 8 ou 16). As *seeds* nada mais são que valores pré-definidos usados para gerar sempre os mesmos valores nos métodos randômicos do Java. Ao todo, foram realizados 120 casos de testes para cada um dos algoritmos (AEMMT de média simples, AEMMT de média harmônica, AEMMD, NSGA-II e NSGA-III), totalizando 600 execuções dos algoritmos.

Como mencionado, os objetivos observados nos algoritmos se mantêm inalterados. São eles: o *makespan*, que é o tempo total necessário para completar todas as tarefas; o *load balance*, que busca a distribuição equilibrada das tarefas entre os processadores; o *flowtime*, que representa a soma dos tempos de conclusão de todas as tarefas; o *communication cost*, que se refere ao custo adicional gerado pela troca de informações entre os processadores, conhecido como overhead; e o *waiting time*, que avalia o período que as tarefas passam aguardando para serem processadas.

Além disso os DAGs utilizados também foram os mesmos, podendo ter 50, 100 ou 300 tarefas. Os DAGs também contêm as restrições de precedência, o custo computacional e o custo de comunicação do problema específico.

### 5.2 Experimentos e Avaliação dos Resultados

Como mencionado, o NSGA-III herda grande parte do código do seu predecessor, tendo alterações apenas no processo de seleção. Assim, com exceção da seleção, os parâmetros de ambos possuem grande semelhança, como observado na Tabela 1.

Tabela 1 – Parâmetros utilizados na execução dos AEMOs.

Parâmetro	AEMMT	AEMMD	NSGA-II	NSGA-III
Tamanho da população	480	480	200	200
Número de gerações	15000	15000	500	500
Avaliações de <i>Fitness</i>	15480	15480	100000	100000
Método de seleção	Torneio Simples ( <i>tour</i> = 2)	Torneio Simples ( <i>tour</i> = 2)	Torneio Simples ( <i>tour</i> = 2)	Torneio Simples ( <i>tour</i> = 2)
Taxa de <i>crossover</i>	-	-	100%	100%
Taxa de mutação	a cada 500 gerações	a cada 500 gerações	2%	2%
Tamanho das tabelas	30	-	-	-
Tamanho da tabela de dominância	100	-	-	-
Método para reinserção	-	-	Reinserção Ordenada	Pontos de Referência

Fonte: Adaptado de Santos (2023)

### 5.2.1 Percentuais por AEMO nos intervalos dos objetivos

Para manter a mesma magnitude das comparações entre os dois trabalhos, os intervalos dos resultados foram mantidos iguais, apenas adicionou-se os resultados do NSGA-III ao conjunto. Os valores dos intervalos para cada um dos objetivos podem ser observados na Tabela 2, aonde, *M*, *L*, *F*, *C* e *W* representam a letra inicial de cada uma das métricas estudadas.

Inicia-se com a análise dos resultados do *makespan*. Através das figuras a seguir, o NSGA-III mostrou-se superior em relação aos outros AEMOs. No entanto, conforme o número de objetivos diminui, sua capacidade de encontrar indivíduos nos intervalos superiores também diminui, sendo que possui seus piores resultados nas execuções de dois objetivos, como demonstrado na Figura 15.

Tabela 2 – Intervalo de valores obtidos para cada objetivo no DAG de 50 tarefas.

Obj.	Inter.	$m = 2$		$m = 4$		$m = 8$		$m = 16$	
		Inf.	Sup.	Inf.	Sup.	Inf.	Sup.	Inf.	Sup.
<i>M</i>	90–100	<b>133,0</b>	148,4	<b>85,0</b>	100,7	<b>74,0</b>	88,7	<b>76,0</b>	85,5
	80–90	148,4	163,8	100,7	116,4	88,7	103,4	85,5	95,0
	70–80	163,8	179,2	116,4	132,1	103,4	118,1	95,0	104,5
	60–70	179,2	194,6	132,1	147,8	118,1	132,8	104,5	114,0
	50–60	194,6	210,0	147,8	163,5	132,8	147,5	114,0	123,5
	40–50	210,0	225,4	163,5	179,2	147,5	162,2	123,5	133,0
	30–40	225,4	240,8	179,2	194,9	162,2	176,9	133,0	142,5
	20–30	240,8	256,2	194,9	210,6	176,9	191,6	142,5	152,0
	10–20	256,2	271,6	210,6	226,3	191,6	206,3	152,0	161,5
	0–10	271,6	<b>287,0</b>	226,3	<b>242,0</b>	206,3	<b>221,0</b>	161,5	<b>171,0</b>
<i>L</i>	90–100	<b>1,00000</b>	1,03233	<b>1,00146</b>	1,04827	<b>1,04092</b>	1,10721	<b>1,18876</b>	1,28219
	80–90	1,03233	1,06465	1,04827	1,09508	1,10721	1,17350	1,28219	1,37563
	70–80	1,06465	1,09698	1,09508	1,14189	1,17350	1,23979	1,37563	1,46907
	60–70	1,09698	1,12931	1,14189	1,18871	1,23979	1,30608	1,46907	1,56251
	50–60	1,12931	1,16163	1,18871	1,23552	1,30608	1,37238	1,56251	1,65595
	40–50	1,16163	1,19396	1,23552	1,28233	1,37238	1,43867	1,65595	1,74939
	30–40	1,19396	1,22628	1,28233	1,32915	1,43867	1,50496	1,74939	1,84283
	20–30	1,22628	1,25861	1,32915	1,37596	1,50496	1,57125	1,84283	1,93626
	10–20	1,25861	1,29094	1,37596	1,42277	1,57125	1,63754	1,93626	2,02970
	0–10	1,29094	<b>1,32326</b>	1,42277	<b>1,46959</b>	1,63754	<b>1,70383</b>	2,02970	<b>2,12314</b>
<i>F</i>	90–100	<b>3263,0</b>	3681,1	<b>2071,0</b>	2475,3	<b>1679,0</b>	2066,8	<b>1672,0</b>	1918,0
	80–90	3681,1	4099,2	2475,3	2879,6	2066,8	2454,6	1918,0	2164,0
	70–80	4099,2	4517,3	2879,6	3283,9	2454,6	2842,4	2164,0	2410,0
	60–70	4517,3	4935,4	3283,9	3688,2	2842,4	3230,2	2410,0	2656,0
	50–60	4935,4	5353,5	3688,2	4092,5	3230,2	3618,0	2656,0	2902,0
	40–50	5353,5	5771,6	4092,5	4496,8	3618,0	4005,8	2902,0	3148,0
	30–40	5771,6	6189,7	4496,8	4901,1	4005,8	4393,6	3148,0	3394,0
	20–30	6189,7	6607,8	4901,1	5305,4	4393,6	4781,4	3394,0	3640,0
	10–20	6607,8	7025,9	5305,4	5709,7	4781,4	5169,2	3640,0	3886,0
	0–10	7025,9	<b>7444,0</b>	5709,7	<b>6114,0</b>	5169,2	<b>5557,0</b>	3886,0	<b>4132,0</b>
<i>C</i>	90–100	<b>127,0</b>	191,2	<b>276,0</b>	345,6	<b>396,0</b>	470,0	<b>467,0</b>	542,5
	80–90	191,2	255,4	345,6	415,2	470,0	544,0	542,5	618,0
	70–80	255,4	319,6	415,2	484,8	544,0	618,0	618,0	693,5
	60–70	319,6	383,8	484,8	554,4	618,0	692,0	693,5	769,0
	50–60	383,8	448,0	554,4	624,0	692,0	766,0	769,0	844,5
	40–50	448,0	512,2	624,0	693,6	766,0	840,0	844,5	920,0
	30–40	512,2	576,4	693,6	763,2	840,0	914,0	920,0	995,5
	20–30	576,4	640,6	763,2	832,8	914,0	988,0	995,5	1071,0
	10–20	640,6	704,8	832,8	902,4	988,0	1062,0	1071,0	1146,5
	0–10	704,8	<b>769,0</b>	902,4	<b>972,0</b>	1062,0	<b>1136,0</b>	1146,5	<b>1222,0</b>
<i>W</i>	90–100	<b>366,0</b>	521,4	<b>262,0</b>	383,6	<b>219,0</b>	314,1	<b>201,0</b>	269,7
	80–90	521,4	676,8	383,6	505,2	314,1	409,2	269,7	338,4
	70–80	676,8	832,2	505,2	626,8	409,2	504,3	338,4	407,1
	60–70	832,2	987,6	626,8	748,4	504,3	599,4	407,1	475,8
	50–60	987,6	1143,0	748,4	870,0	599,4	694,5	475,8	544,5
	40–50	1143,0	1298,4	870,0	991,6	694,5	789,6	544,5	613,2
	30–40	1298,4	1453,8	991,6	1113,2	789,6	884,7	613,2	681,9
	20–30	1453,8	1609,2	1113,2	1234,8	884,7	979,8	681,9	750,6
	10–20	1609,2	1764,6	1234,8	1356,4	979,8	1074,9	750,6	819,3
	0–10	1764,6	<b>1920,0</b>	1356,4	<b>1478,0</b>	1074,9	<b>1170,0</b>	819,3	<b>888,0</b>

Fonte: Adaptado de Santos (2023)

Figura 12 – Percentuais obtidos por AEMO nos intervalos de valores do *Makespan* em um DAG de 50 tarefas com execução de 5 objetivos.

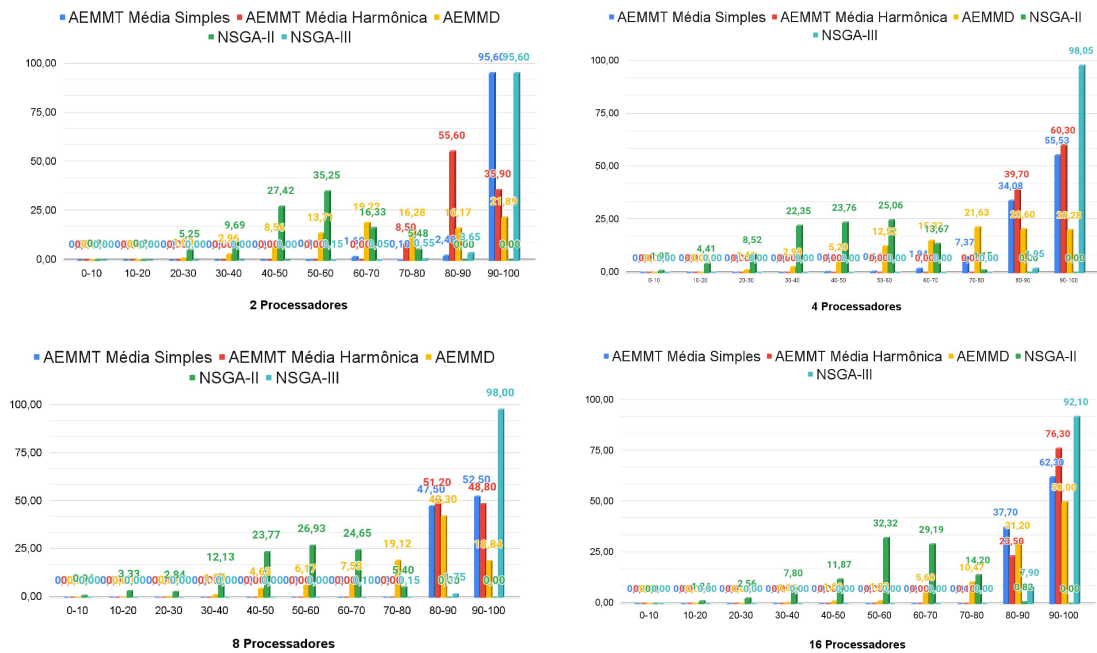


Figura 13 – Percentuais obtidos por AEMO nos intervalos de valores do *Makespan* em um DAG de 50 tarefas com execução de 4 objetivos.

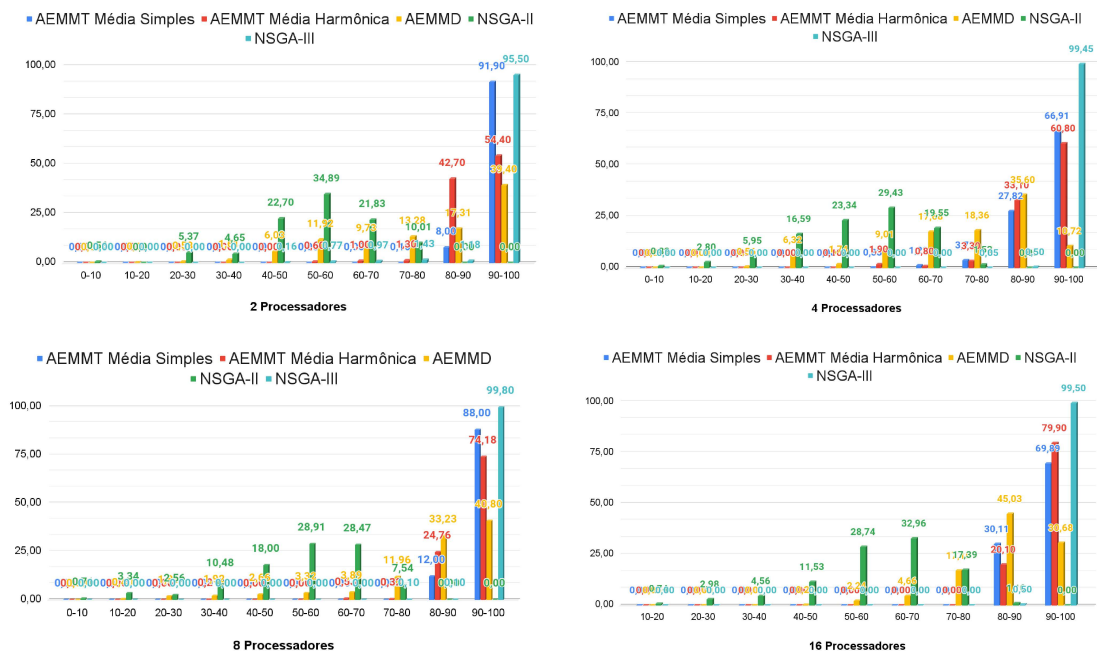


Figura 14 – Percentuais obtidos por AEMO nos intervalos de valores do *Makespan* em um DAG de 50 tarefas com execução de 3 objetivos.

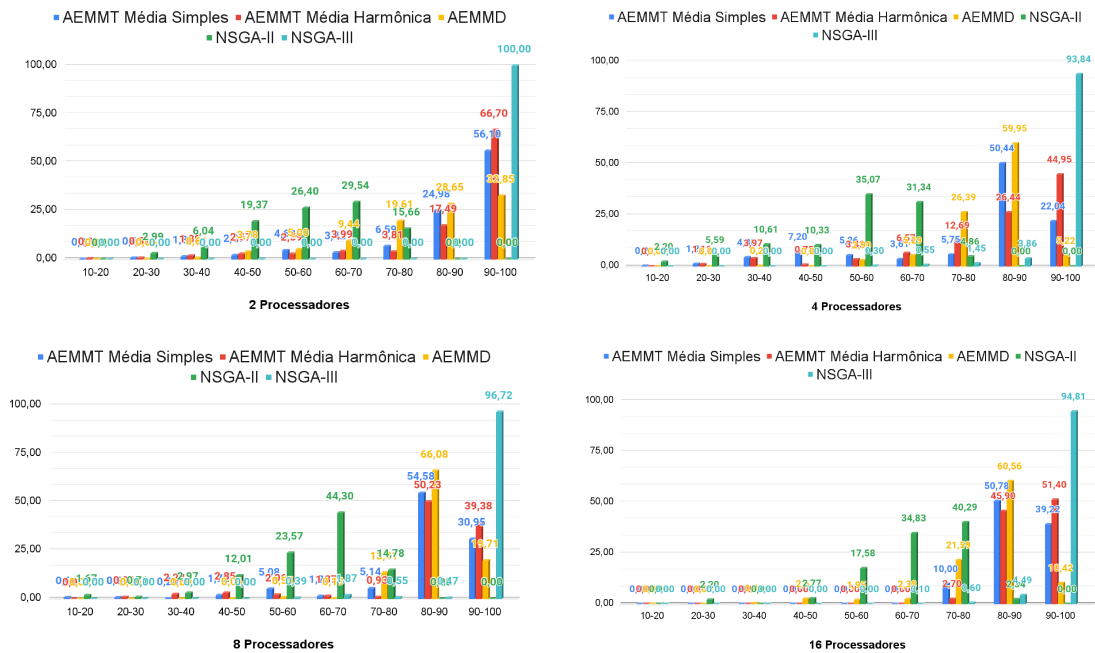
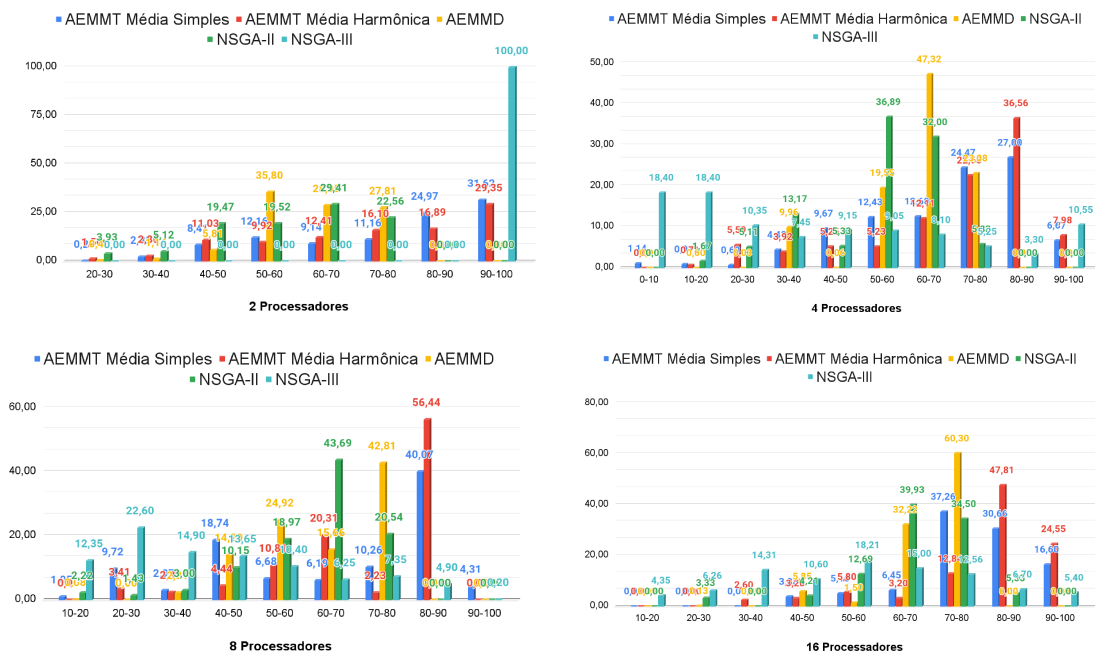


Figura 15 – Percentuais obtidos por AEMO nos intervalos de valores do *Makespan* em um DAG de 50 tarefas com execução de 2 objetivos.



A seguir, tem-se os resultados do *load balance*. Nesse caso, os indivíduos produzidos pelo NSGA-III ficaram atrás em quase todas as execuções, sobressaindo apenas nas práticas de quatro, oito e dezesseis processadores e dois objetivos, observado na Figura 19.

Há pontos interessantes a serem notados nos resultados a seguir, em várias ocasiões o algoritmo AEMMT (Algoritmo Evolutivo Multi-Objetivo com Múltiplas Tabelas) de média simples apresentou a maior quantidade de soluções nos intervalos superiores entre todos os AEMOs, como é o exemplo na Figura 16. Com exceção de dezesseis processadores, isso demonstra a capacidade do AEMMT de encontrar bons resultados de um objetivo específico mesmo considerando vários simultaneamente.

Outra observação importante, são os resultados das práticas em dois processadores do *load balance*. As execuções do NSGA-III apresentaram resultados muito semelhantes aos do NSGA-II, um algoritmo muito menos complexo. Isso reforça ainda mais a recomendação de uso do NSGA-II, o qual é aconselhado para problemas de até dois objetivos.

Além disso, também vale notar que o AEMMT de média harmônica também se sobressaiu, o qual apresentou melhores resultados na execução de cinco objetivos e dezesseis processadores, como demonstrado na Figura 16.

Figura 16 – Percentuais obtidos por AEMO nos intervalos de valores do *Load Balance* em um DAG de 50 tarefas com execução de 5 objetivos.

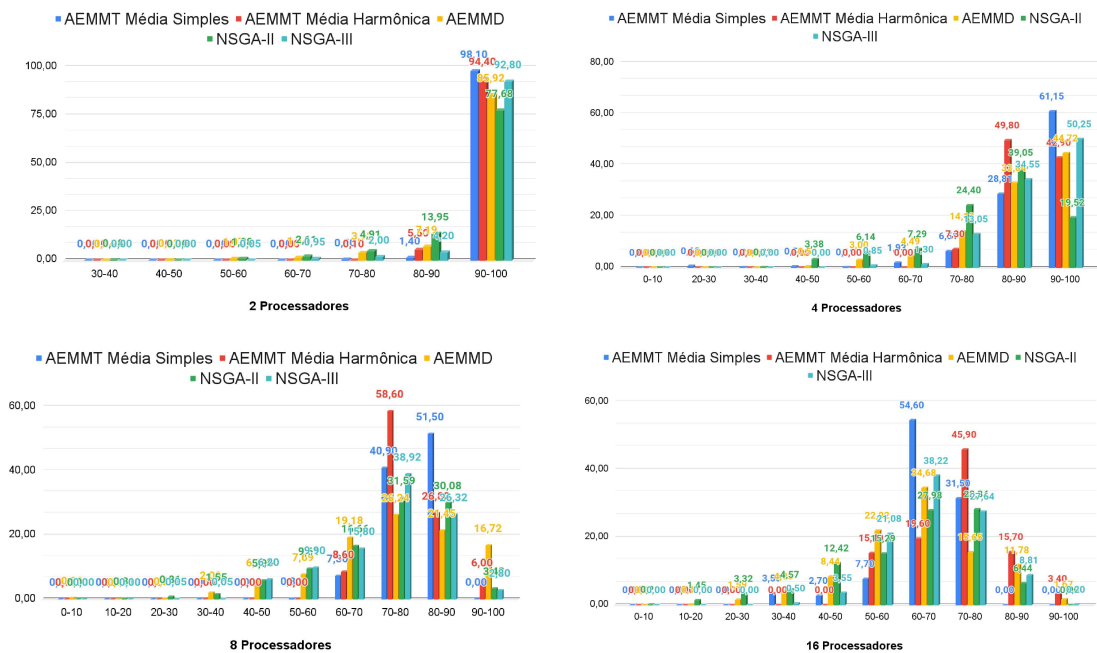




Figura 17 – Percentuais obtidos por AEMO nos intervalos de valores do *Load Balance* em um DAG de 50 tarefas com execução de 4 objetivos.

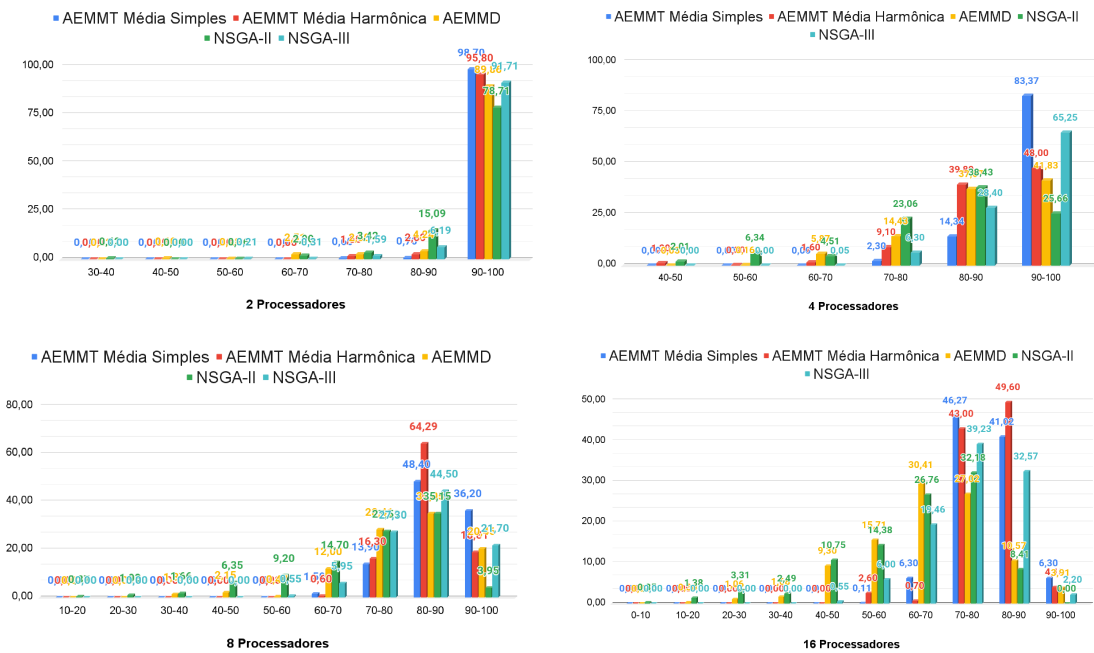


Figura 18 – Percentuais obtidos por AEMO nos intervalos de valores do *Load Balance* em um DAG de 50 tarefas com execução de 3 objetivos.

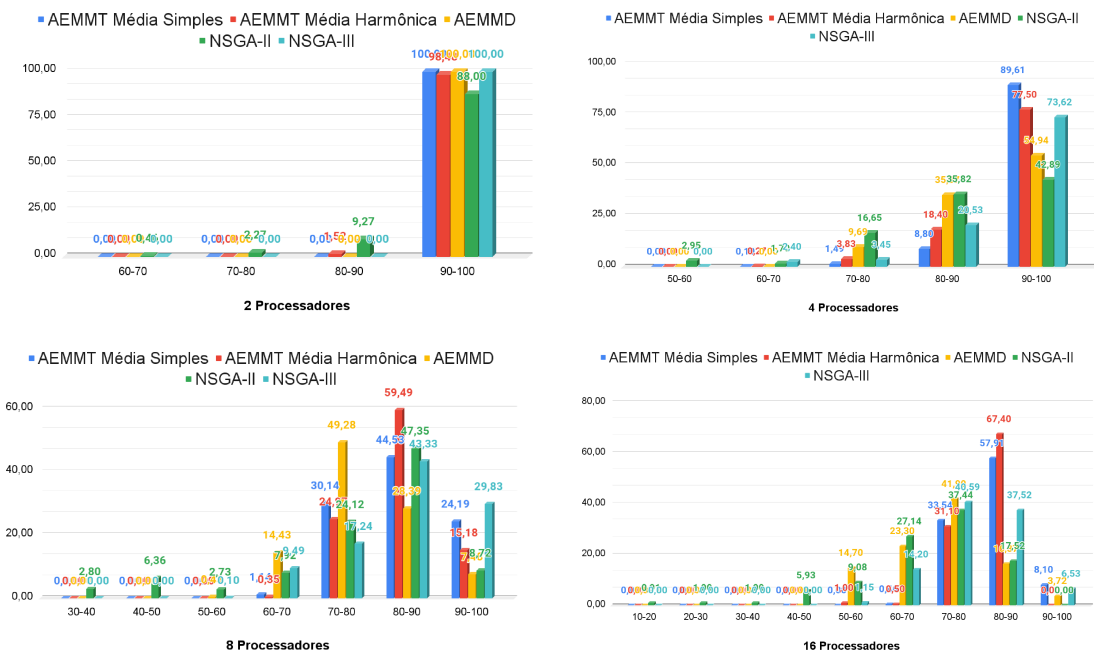
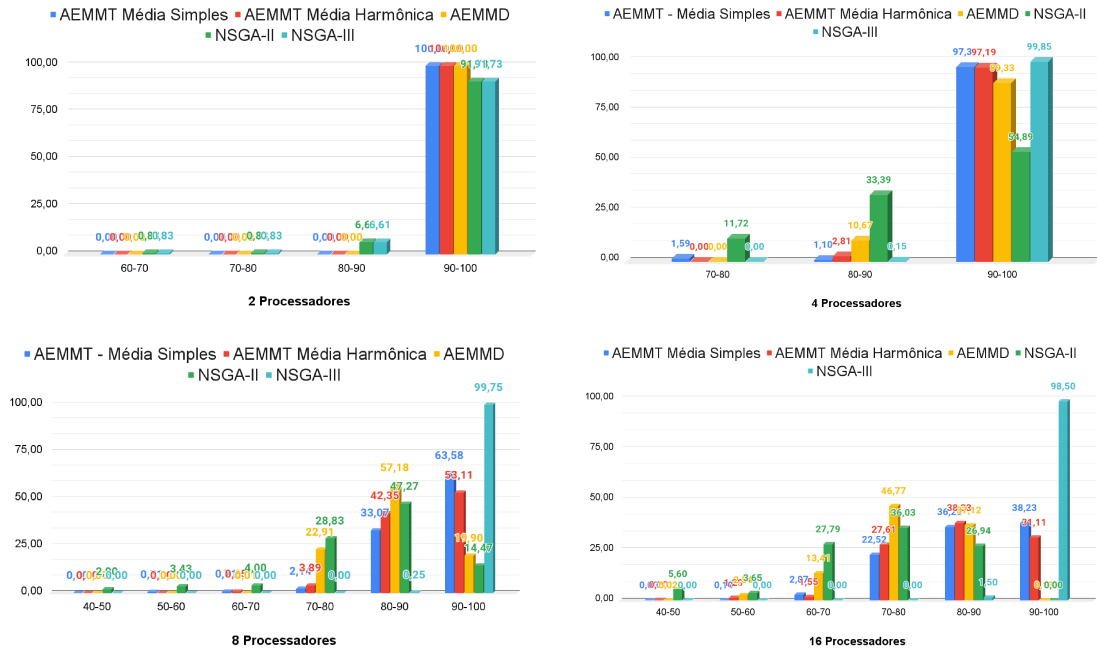


Figura 19 – Percentuais obtidos por AEMO nos intervalos de valores do *Load Balance* em um DAG de 50 tarefas com execução de 2 objetivos.



O *flow time* apresentou resultados diferentes. Na maioria das vezes o NSGA-III foi superior em produzir uma quantidade maior de indivíduos nos limites superiores. Perdendo apenas para o AEMMT de média simples, na execução de cinco objetivos e dezesseis processadores, como demonstrado na Figura 20.

No entanto, vale notar que, mesmo tendo saído na frente na quantidade de soluções nas execuções do *flow time*, o AEMMT de média simples também produziu ótimos resultados. Ficou na maioria das vezes em segundo lugar, perdendo essa posição apenas em três objetivos, no qual foi ultrapassado pelo AEMMT de média harmônica.

Figura 20 – Percentuais obtidos por AEMO nos intervalos de valores do *Flow Time* em um DAG de 50 tarefas com execução de 5 objetivos.

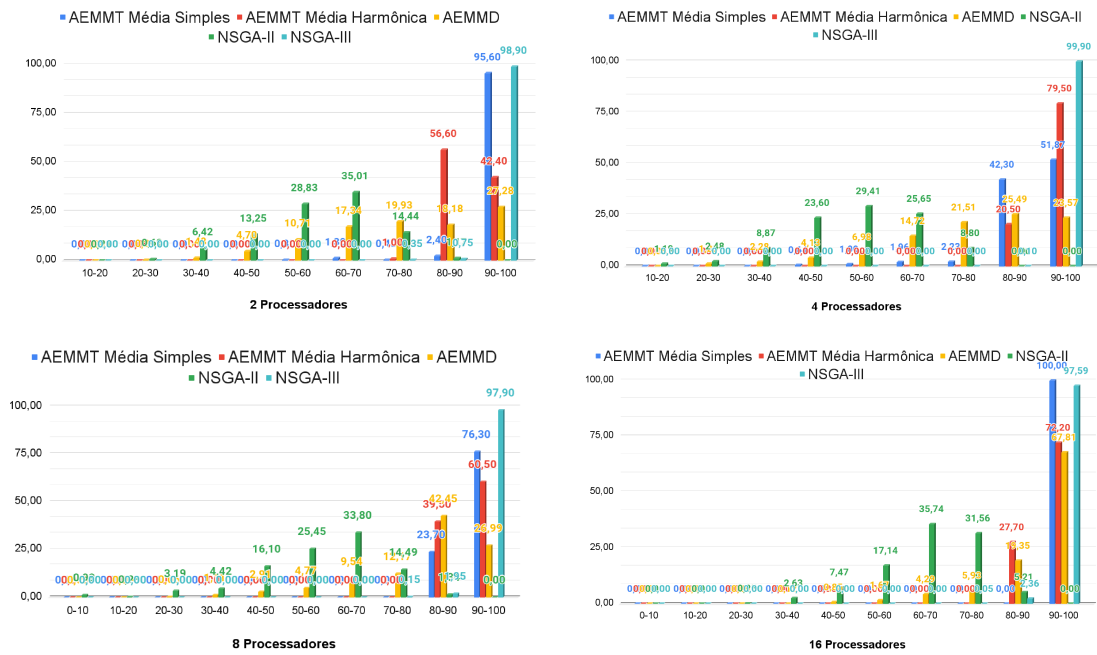


Figura 21 – Percentuais obtidos por AEMO nos intervalos de valores do *Flow Time* em um DAG de 50 tarefas com execução de 4 objetivos.

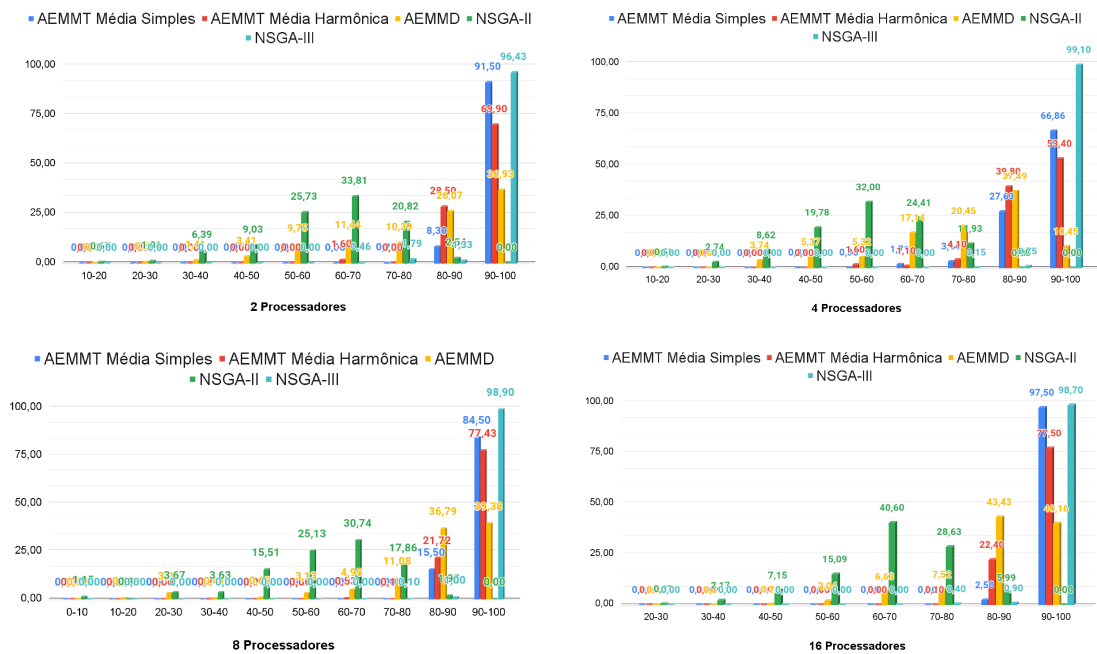
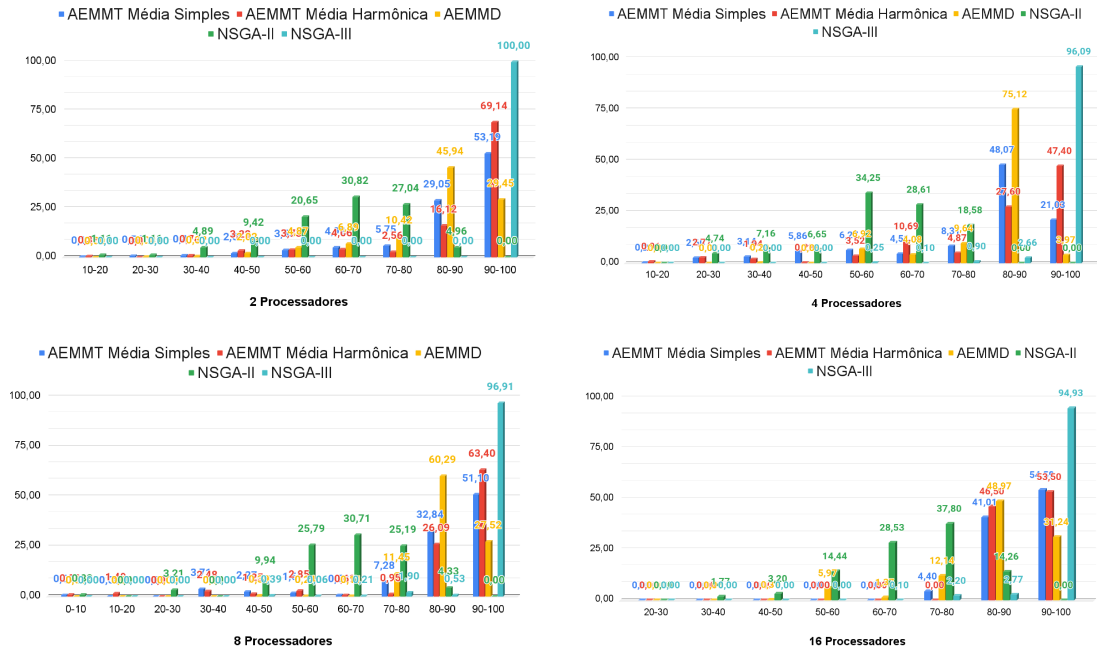


Figura 22 – Percentuais obtidos por AEMO nos intervalos de valores do *Flow Time* em um DAG de 50 tarefas com execução de 3 objetivos.



No objetivo *communication cost* continua-se a ver os bons resultados do NSGA-III na produção de boas soluções nos intervalos superiores. Superando todos os outros AEMOs, perdeu apenas para o AEMMD no intervalo de 90-100 na execução de quatro objetivos e quatro processadores.

Vale observar também que nessa parte nota-se o início das melhoras nos algoritmos AEMMD e AEMMT de média harmônica em relação ao AEMMT de média simples. O qual até o momento manteve-se majoritariamente à frente dos AEMOs, com exceção do NSGA-III, que prevaleceu em grande parte dos cenários observados.

Figura 23 – Percentuais obtidos por AEMO nos intervalos de valores do *Communication Cost* em um DAG de 50 tarefas com execução de 5 objetivos.

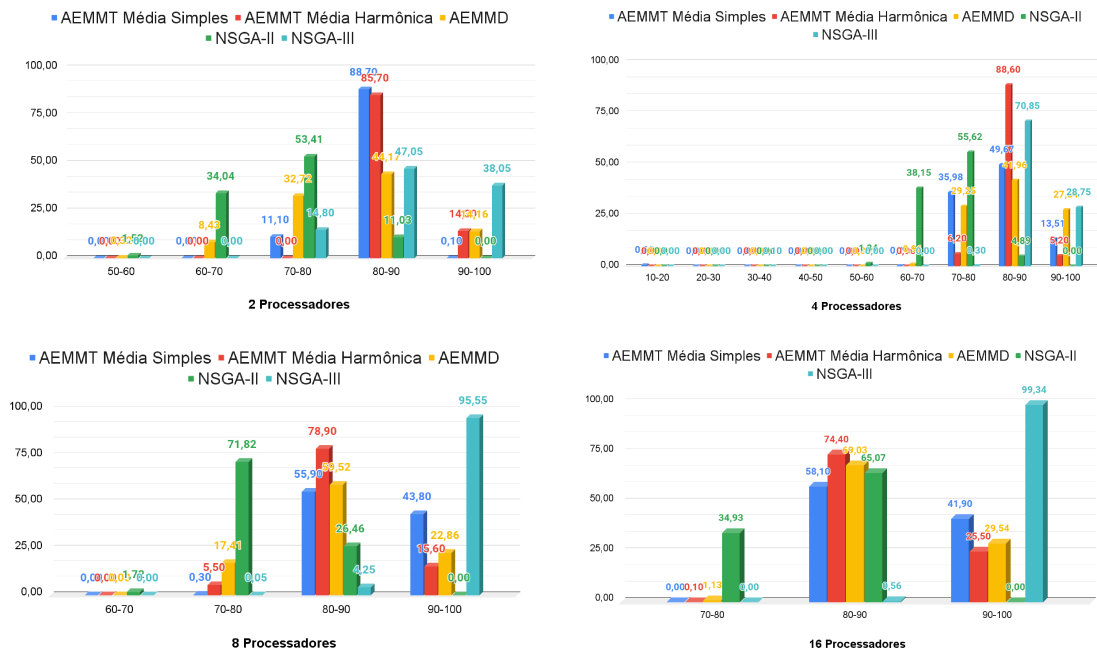
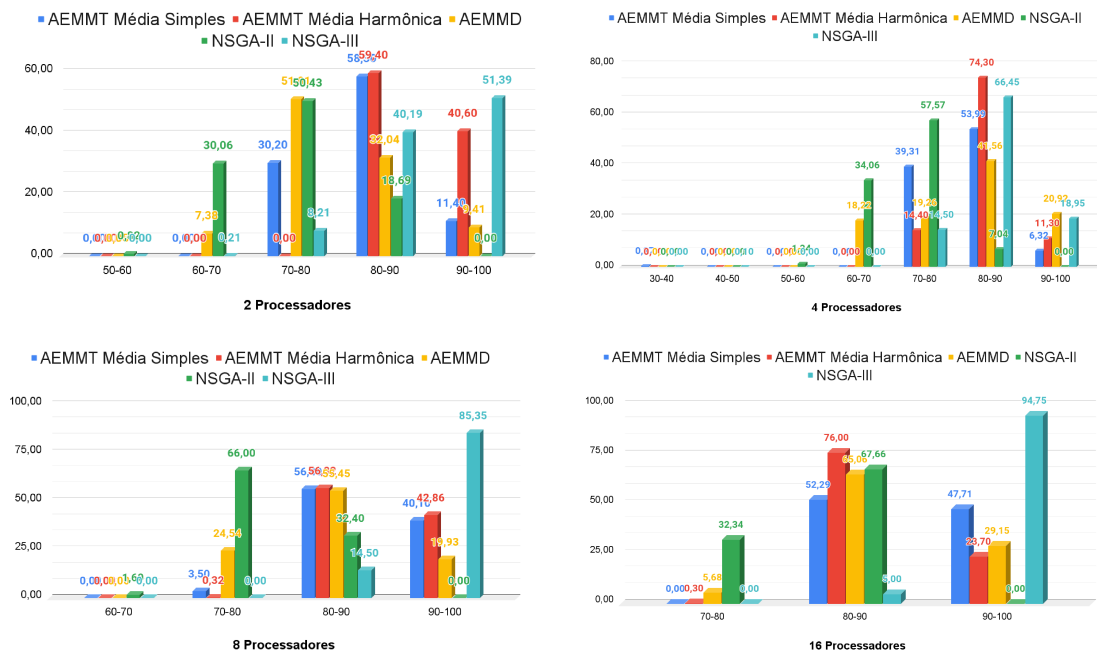
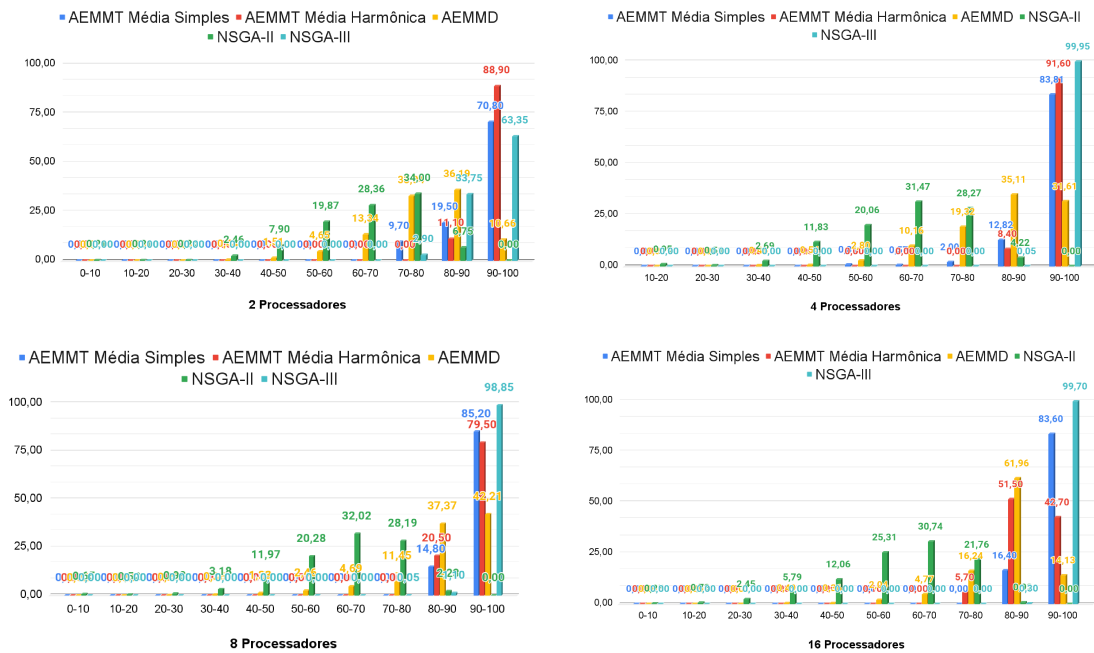


Figura 24 – Percentuais obtidos por AEMO nos intervalos de valores do *Communication Cost* em um DAG de 50 tarefas com execução de 4 objetivos.



Por último, tem-se o *waiting time*, no qual o NSGA-III também prevaleceu, perdendo apenas na prática de dois processadores, como observado na Figura 25. Aqui também nota-se o AEMMT de média harmônica, o qual também teve bons resultados.

Figura 25 – Percentuais obtidos por AEMO nos intervalos de valores do *Waiting Time* em um DAG de 50 tarefas com execução de 5 objetivos.



Através das diversas execuções em variados cenários, pode-se observar que quanto mais complexo o problema foi (quantidade de processadores e objetivos) melhor foi seu desempenho em relação aos outros AEMOs. Como citado no estudo de [Ishibuchi et al. \(2016\)](#), o NSGA-III tem seu maior destaque conforme o número de objetivos presentes aumenta, obtendo os melhores resultados a partir de dez objetivos.

Vale lembrar também, que estes resultados se referem à quantidade de soluções presentes nos intervalos superiores de cada objetivo individualmente, e não da solução como um todo. A análise para encontrar o melhor resultado do objetivo e o melhor indivíduo será feita nos capítulos a seguir.

### 5.2.2 Melhores valores de objetivos encontrados pelos AEMOs

A seguir, será apresentado os melhores valores para cada um dos objetivos, analisados individualmente em um DAG de 50 tarefas em variados processadores (2, 4, 8 e 16). Aqui, observam-se ótimos resultados do NSGA-III, o qual obteve o melhor resultado 80% das vezes. O AEMMD veio em segundo lugar, sobressaindo-se 15% das vezes.

Empatados em terceiro lugar vem o AEMMT de média simples (AEMMT1) e o AEMMT de média harmônica (AEMMT2) com 10%. Já o NSGA-II ficou em último, encontrando o melhor valor apenas uma vez, ou 5% das vezes. Todos esses valores estão demonstrado na Tabela 3.

Tabela 3 – Melhores objetivos encontrados por AEMO em um DAG de 50 tarefas na execução com 5 objetivos.

$m$	Obj.	AEMMT1		AEMMT2		AEMMD		NSGA-II		NSGA-III	
		melhor	média	melhor	média	melhor	média	melhor	média	melhor	média
2	$M$	133,000	134,600±1,955	133,000	142,900±5,587	133,000	135,200±2,251	167,000	173,500±5,603	<b>132,000</b>	132,500±0,527
	$L$	<b>1,00000</b>	1,00036±0,00114	<b>1,00000</b>	1,00136±0,00325	<b>1,00000</b>	1,00000±0,00000	<b>1,00000</b>	1,00022±0,00069	<b>1,00000</b>	1,00034±0,00109
	$F$	3258,000	3350,900±70,693	3300,000	3537,700±132,766	3299,000	3371,800±77,740	3917,000	4122,900±139,155	<b>3208,000</b>	3245,600±30,486
	$C$	191,000	212,000±18,791	146,000	190,100±20,388	127,000	160,000±24,545	208,000	226,200±9,647	<b>121,000</b>	173,100±47,774
	$W$	382,000	450,000±73,224	<b>374,000</b>	436,100±30,377	385,000	439,300±31,833	561,000	621,500±32,153	376,000	433,700±58,805
4	$M$	82,000	95,400±8,746	88,000	95,700±4,111	85,000	94,900±6,226	127,000	137,600±6,736	<b>78,000</b>	82,100±3,247
	$L$	<b>1,00386</b>	1,01254±0,00808	1,01064	1,03198±0,01526	1,00146	1,00698±0,00443	1,01155	1,02356±0,00692	1,00923	1,01576±0,00588
	$F$	2022,000	2318,500±191,513	2107,000	2290,800±123,105	2136,000	2267,500±104,138	3000,000	3128,200±117,795	<b>1896,0000</b>	1983,800±63,676
	$C$	316,000	366,600±39,822	327,000	365,900±24,062	305,000	332,000±16,425	353,000	404,300±19,883	<b>272,000</b>	325,900±30,005
	$W$	236,000	298,700±33,247	267,000	318,800±31,365	262,000	305,500±27,273	437,000	496,300±32,121	<b>210,000</b>	245,200±22,987
8	$M$	80,000	87,300±4,398	80,000	89,000±5,981	75,000	87,000±6,864	108,000	114,000±4,372	<b>67,000</b>	71,300±3,622
	$L$	1,11515	1,14781±0,02160	1,07533	1,14477±0,03766	<b>1,04092</b>	1,06321±0,01075	1,07648	1,10041±0,01619	1,05846	1,11588±0,03545
	$F$	1803,000	1926,800±95,211	1837,000	1974,100±107,410	1718,000	1963,300±140,580	2330,000	2478,100±74,536	<b>1513,000</b>	1574,000±51,502
	$C$	433,000	463,400±21,930	436,000	472,700±21,562	418,000	448,300±16,647	479,000	505,700±13,073	<b>344,000</b>	397,400±23,363
	$W$	230,000	264,700±20,955	225,000	258,400±24,254	227,000	266,300±24,143	389,000	406,600±11,325	<b>152,000</b>	165,700±11,235
16	$M$	74,000	82,900±5,109	78,000	84,000±3,496	78,000	82,700±3,057	96,000	101,100±3,814	<b>69,000</b>	73,200±3,910
	$L$	1,36864	1,43259±0,04959	1,24731	1,33584±0,04967	<b>1,18876</b>	1,22430±0,01854	1,26796	1,33138±0,03055	1,21725	1,33355±0,06859
	$F$	1612,000	1735,300±63,732	1747,000	1829,200±80,377	1767,000	1799,300±24,734	2071,000	2205,000±89,800	<b>1537,000</b>	1579,900±31,942
	$C$	507,000	531,800±20,054	491,000	541,100±22,610	489,000	506,000±11,324	546,000	561,900±10,898	<b>429,000</b>	455,800±13,028
	$W$	219,000	237,800±15,915	233,000	256,100±15,351	221,000	249,900±14,433	317,000	345,300±17,218	<b>164,000</b>	174,300±7,196

### 5.2.3 Melhores indivíduos encontrados pelos AEMOs

Nesta seção, serão demonstrados os melhores indivíduos encontrados em cada uma das soluções com base na média simples, média harmônica e hiper-volume. Para comparação das médias, os dados foram normalizados através de uma função escalar para calcular a aptidão do indivíduo. O processo de normalização não foi alterado do estudo de Santos (2023), tão pouco os parâmetros usados, com o objetivo de manter a mesma magnitude de comparação entre os dois trabalhos. Por esse motivo, existem valores de aptidão que ultrapassam 1,0000.

#### 5.2.3.1 Média Simples

Nos capítulos anteriores, o NSGA-III ficou parelho com os outros AEMOs na quantidade de soluções produzidas nos intervalos superiores. No entanto, agora será analisado a qualidade dos indivíduos em si. Através das próximas tabelas (Tabela 4, Tabela 5, Tabela 6) nota-se a superioridade do NSGA-III, o qual produziu as melhores soluções 97,91% das vezes, perdendo apenas uma vez para o AEMMT de média simples, demonstrado na Tabela 5 em dois objetivos e quatro processadores.

O AEMMT1 (AEMMT de média simples) ficou em segundo lugar na grande maioria das vezes, e em alguns momentos produzindo soluções quase tão boas quanto as do NSGA-III. Em terceiro lugar veio o AEMMD seguido do NSGA-II, o qual produziu resultados muito piores que os de seus concorrentes sempre que saia do escopo de dois objetivos e dois processadores.

Tabela 4 – Melhores indivíduos com média simples encontrados por AEMO em um DAG de 50 tarefas.

Obj.	$m$	AEMMT1		AEMMD		NSGA-II		NSGA-III	
		melhor	média	melhor	média	melhor	média	melhor	média
5	2	0,8813	0,8178±0,0323	0,8416	0,7871±0,0366	0,5838	0,5487±0,0237	<b>0,9149</b>	0,8603±0,0387
	4	0,9016	0,7865±0,0652	0,8809	0,7549±0,0550	0,5183	0,4686±0,0239	<b>0,9852</b>	0,9278±0,0227
	8	0,7959	0,7563±0,0354	0,8210	0,7198±0,0556	0,5356	0,5015±0,0189	<b>1,0676</b>	1,0039±0,0480
	16	0,8498	0,7739±0,0389	0,7710	0,7385±0,0212	0,5691	0,5350±0,0259	<b>1,0093</b>	0,9532±0,0316
4	2	0,8970	0,8355±0,0288	0,8280	0,8025±0,0172	0,6275	0,5982±0,0247	<b>0,9590</b>	0,9051±0,0316
	4	0,8959	0,8138±0,0454	0,7533	0,7020±0,0419	0,5503	0,5093±0,0211	<b>0,9840</b>	0,9347±0,0381
	8	0,8523	0,7849±0,0462	0,8128	0,7502±0,0429	0,5750	0,5450±0,0235	<b>0,9913</b>	0,9360±0,0504
	16	0,8268	0,7857±0,0285	0,8135	0,7138±0,0500	0,6335	0,5727±0,0314	<b>0,9872</b>	0,9159±0,0316
3	2	0,9757	0,9285±0,0436	0,9464	0,8763±0,0716	0,7546	0,6838±0,0383	<b>1,0160</b>	0,9862±0,0162
	4	0,8629	0,8074±0,0467	0,8122	0,7331±0,0494	0,6067	0,5428±0,0339	<b>1,0466</b>	1,0091±0,0247
	8	0,8632	0,7652±0,0512	0,8729	0,6982±0,0732	0,5858	0,5392±0,0333	<b>1,0387</b>	0,9774±0,0357
	16	0,8040	0,7396±0,0473	0,7814	0,6978±0,0556	0,6231	0,5553±0,0347	<b>1,0264</b>	0,9409±0,0499
2	2	0,9728	0,9404±0,0266	0,8222	0,7915±0,0323	0,7998	0,7530±0,0309	<b>1,0000</b>	0,9877±0,0134
	4	0,8827	0,8049±0,0393	0,7123	0,6723±0,0219	0,6633	0,6158±0,0294	<b>0,9360</b>	0,8521±0,0455
	8	0,7711	0,7424±0,0213	0,7303	0,6418±0,0576	0,6231	0,5855±0,0259	<b>0,8945</b>	0,8164±0,0546
	16	0,8045	0,7359±0,0398	0,6642	0,6132±0,0363	0,6403	0,5719±0,0294	<b>0,9365</b>	0,8550±0,0553



Tabela 5 – Melhores indivíduos com média simples encontrados por AEMO em um DAG de 100 tarefas.

Obj.	$m$	AEMMT1		AEMMD		NSGA-II		NSGA-III	
		melhor	média	melhor	média	melhor	média	melhor	média
5	2	0,8644	0,8395±0,0171	0,8446	0,8075±0,0263	0,6348	0,5804±0,0295	<b>0,9118</b>	0,8702±0,0271
	4	0,8641	0,8070±0,0316	0,8689	0,7880±0,0392	0,5896	0,5494±0,0234	<b>1,0508</b>	0,9849±0,0410
	8	0,9065	0,8335±0,0397	0,8963	0,8270±0,0406	0,5748	0,5528±0,0189	<b>1,0994</b>	1,0298±0,0521
	16	0,8861	0,8633±0,0198	0,8874	0,8513±0,0205	0,6654	0,6174±0,0256	<b>1,0390</b>	1,0126±0,0178
4	2	0,9264	0,8914±0,0203	0,8925	0,8394±0,0309	0,6486	0,6192±0,0242	<b>0,9461</b>	0,9300±0,0159
	4	0,9158	0,8672±0,0476	0,8550	0,8068±0,0357	0,6293	0,5876±0,0207	<b>1,0445</b>	1,0154±0,0201
	8	0,8759	0,8526±0,0168	0,8509	0,8200±0,0212	0,6295	0,5996±0,0209	<b>1,0573</b>	1,0249±0,0203
	16	0,9063	0,8741±0,0155	0,8841	0,8452±0,0296	0,6969	0,6568±0,0277	<b>1,0509</b>	1,0152±0,0265
3	2	0,9746	0,9189±0,0331	0,9410	0,8739±0,0511	0,6977	0,6545±0,0306	<b>1,0104</b>	0,9937±0,0107
	4	0,9207	0,8142±0,0515	0,7654	0,7031±0,0526	0,5839	0,5460±0,0239	<b>1,1252</b>	1,0555±0,0488
	8	0,8400	0,8001±0,0296	0,7766	0,7092±0,0389	0,5728	0,5378±0,0232	<b>1,0869</b>	1,0066±0,0543
	16	0,8881	0,8428±0,0374	0,8430	0,7634±0,0603	0,6444	0,5900±0,0370	<b>1,0364</b>	0,9935±0,0331
2	2	0,9791	0,9008±0,0410	0,8159	0,7710±0,0341	0,7448	0,7110±0,0308	<b>1,0000</b>	0,9833±0,0186
	4	<b>0,9069</b>	0,8195±0,0512	0,7675	0,6955±0,0345	0,6619	0,6222±0,0221	0,8988	0,8474±0,0249
	8	0,8421	0,7844±0,0347	0,6713	0,6297±0,0319	0,6209	0,6004±0,0134	<b>0,8640</b>	0,8128±0,0407
	16	0,8698	0,8128±0,0360	0,7913	0,6879±0,0591	0,6897	0,6155±0,0326	<b>0,9080</b>	0,8615±0,0391

Tabela 6 – Melhores indivíduos com média simples encontrados por AEMO em um DAG de 300 tarefas.

Obj.	$m$	AEMMT1		AEMMD		NSGA-II		NSGA-III	
		melhor	média	melhor	média	melhor	média	melhor	média
5	2	0,8908	0,8483±0,0320	0,8551	0,8184±0,0354	0,5491	0,5315±0,0172	<b>1,0429</b>	0,9693±0,0464
	4	0,8612	0,7947±0,0427	0,8187	0,7109±0,0560	0,4986	0,4891±0,0086	<b>0,9901</b>	0,9473±0,0461
	8	0,8739	0,8170±0,0531	0,8680	0,8384±0,0249	0,5425	0,5149±0,0167	<b>1,1316</b>	1,0585±0,0670
	16	0,8729	0,7792±0,0455	0,8431	0,7683±0,0411	0,5132	0,4953±0,0134	<b>1,1406</b>	1,0561±0,0503
4	2	0,9734	0,8834±0,0584	0,8605	0,7947±0,0403	0,5924	0,5539±0,0200	<b>1,0654</b>	1,0051±0,0400
	4	0,9064	0,8565±0,0401	0,8219	0,7610±0,0375	0,5406	0,5290±0,0074	<b>1,1150</b>	0,9824±0,0897
	8	0,8595	0,8215±0,0257	0,8741	0,7873±0,0449	0,5583	0,5307±0,0114	<b>1,1930</b>	1,0192±0,0815
	16	0,9117	0,8389±0,0433	0,8251	0,7840±0,0293	0,5569	0,5392±0,0119	<b>1,1703</b>	1,0678±0,0427
3	2	0,8931	0,8501±0,0334	0,8638	0,7342±0,0585	0,5924	0,5372±0,0286	<b>1,1313</b>	1,0833±0,0236
	4	0,8695	0,7715±0,0462	0,7288	0,6198±0,0542	0,4851	0,4662±0,0085	<b>1,0138</b>	0,9499±0,0439
	8	0,8200	0,7383±0,0706	0,7147	0,6128±0,0712	0,4778	0,4329±0,0205	<b>1,0864</b>	0,9214±0,0914
	16	0,8248	0,7489±0,0492	0,6731	0,6074±0,0588	0,4553	0,4316±0,0173	<b>1,1264</b>	0,9453±0,0985
2	2	0,9155	0,8563±0,0466	0,7410	0,6684±0,0426	0,6633	0,6297±0,0177	<b>1,1124</b>	1,0057±0,0651
	4	0,8017	0,7458±0,0356	0,6592	0,6296±0,0243	0,5863	0,5735±0,0100	<b>0,8780</b>	0,7966±0,0381
	8	0,7757	0,7264±0,0393	0,6224	0,5848±0,0234	0,5672	0,5408±0,0175	<b>0,9030</b>	0,7693±0,0697
	16	0,7782	0,7327±0,0340	0,6680	0,5995±0,0377	0,5669	0,5282±0,0211	<b>0,9199</b>	0,8034±0,0609

Os gráficos a seguir demonstram a tendência da melhor solução por AEMOs conforme o número de objetivos diminui. Nota-se que a capacidade do NSGA-III de encontrar um bom indivíduo tende a diminuir com a redução dos objetivos, como observado em quase todas as imagens a seguir.

Figura 26 – Tendência do melhor indivíduo por AEMO no DAG de 50 tarefas com redução de objetivos - Média Simples

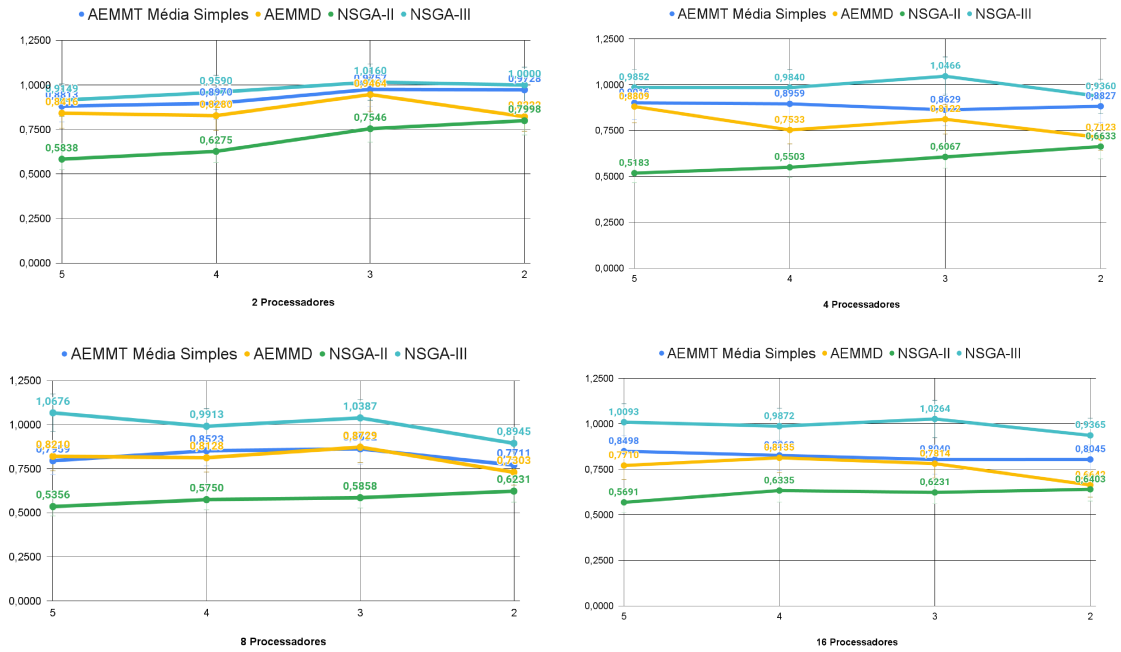


Figura 27 – Tendência do melhor indivíduo por AEMO no DAG de 100 tarefas com redução de objetivos - Média Simples

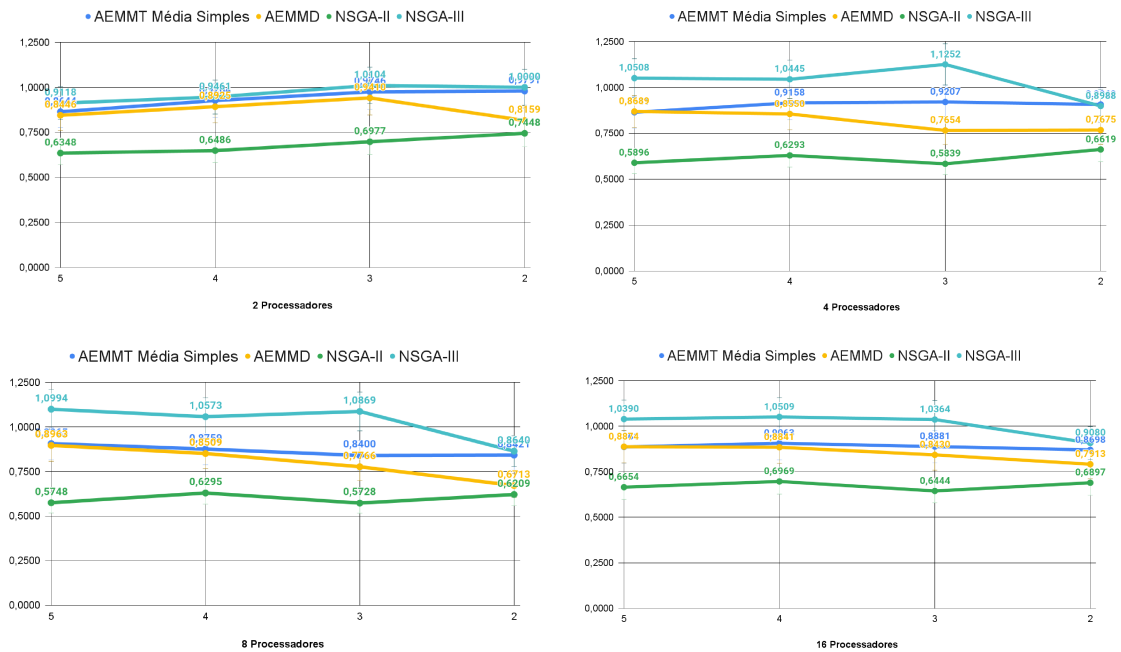
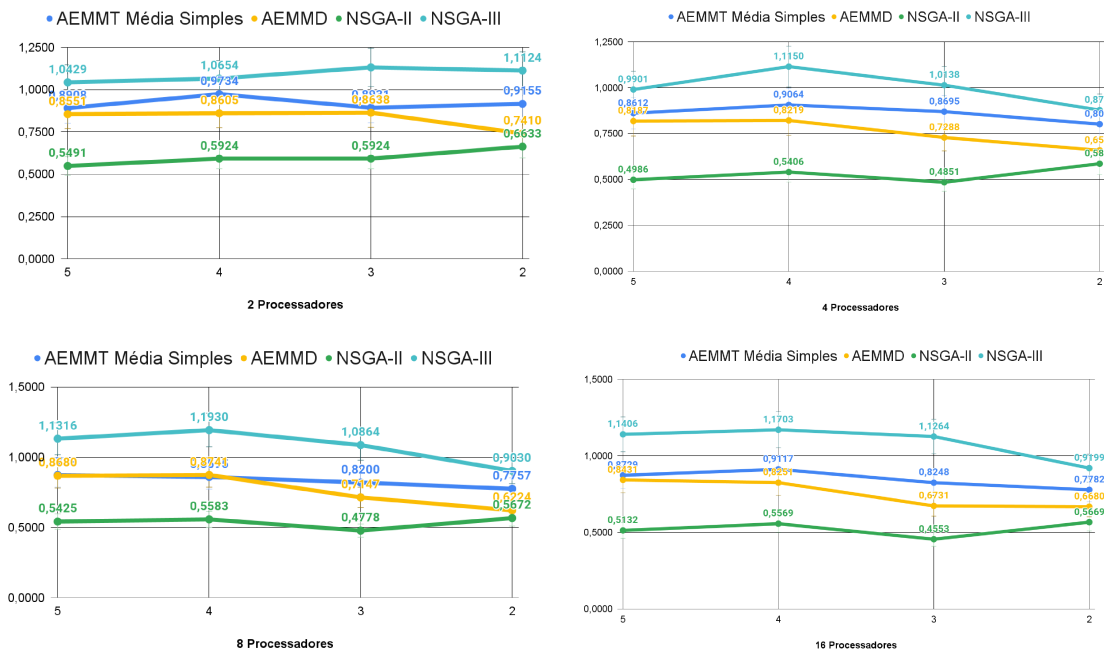


Figura 28 – Tendência do melhor indivíduo por AEMO no DAG de 300 tarefas com redução de objetivos - Média Simples



### 5.2.3.2 Média Harmônica

Observa-se nas tabelas a seguir, que a média harmônica tem resultados bem semelhantes à média simples. O NSGA-III produziu melhores soluções 100% das vezes, no entanto vale notar que, o AEMMT de média harmônica também teve ótimos resultados, praticamente empatando com o NSGA-III em alguns casos, como demonstrado na Tabela 7 em dois objetivos e dois processadores, e na Tabela 8 em dois objetivos e quase todos os processadores.

Tais resultados do NSGA-III são fruto do uso de uma grande quantidade de pontos de referência, os quais fazem com que as soluções fiquem muito bem distribuídas nas Fronteiras de Pareto, como citado anteriormente no capítulo de desenvolvimento.

Tabela 7 – Melhores indivíduos com média harmônica encontrados por AEMO em um DAG de 50 tarefas.

Obj.	$m$	AEMMT2		AEMMD		NSGA-II		NSGA-III	
		melhor	média	melhor	média	melhor	média	melhor	média
5	2	0,8436	0,7446±0,0379	0,7764	0,7241±0,0389	0,5838	0,5487±0,0237	<b>0,9091</b>	0,8233±0,0697
	4	0,8055	0,7602±0,0373	0,8615	0,7324±0,0564	0,5183	0,4686±0,0239	<b>0,9772</b>	0,9040±0,0296
	8	0,7674	0,7282±0,0354	0,7950	0,7098±0,0527	0,5356	0,5015±0,0189	<b>1,0045</b>	0,9594±0,0352
	16	0,7884	0,7311±0,0399	0,7424	0,7153±0,0227	0,5531	0,5158±0,0279	<b>0,9755</b>	0,9027±0,0335
4	2	0,8445	0,7888±0,0342	0,7638	0,7276±0,0309	0,5805	0,5286±0,0389	<b>0,9578</b>	0,8845±0,0482
	4	0,8171	0,7695±0,0331	0,7256	0,6681±0,0477	0,5050	0,4682±0,0232	<b>0,9731</b>	0,8984±0,0466
	8	0,8382	0,7817±0,0419	0,8124	0,7442±0,0463	0,5672	0,5244±0,0337	<b>0,9658</b>	0,9238±0,0466
	16	0,8570	0,7813±0,0480	0,7880	0,7056±0,0447	0,6279	0,5664±0,0322	<b>0,9420</b>	0,9009±0,0224
3	2	0,9777	0,9480±0,0275	0,9457	0,8673±0,0804	0,7225	0,6534±0,0406	<b>1,0158</b>	0,9858±0,0166
	4	0,9314	0,8324±0,0748	0,8025	0,7198±0,0514	0,5651	0,4991±0,0395	<b>1,0440</b>	1,0046±0,0229
	8	0,8257	0,7698±0,0485	0,8688	0,6904±0,0804	0,5759	0,5149±0,0432	<b>1,0236</b>	0,9706±0,0338
	16	0,8026	0,7505±0,0440	0,7681	0,6926±0,0537	0,6165	0,5502±0,0349	<b>0,9925</b>	0,9225±0,0450
2	2	0,9930	0,9443±0,0276	0,7890	0,7439±0,0450	0,7556	0,6910±0,0494	<b>1,0000</b>	0,9874±0,0139
	4	0,8662	0,7927±0,0473	0,6578	0,6160±0,0355	0,6094	0,5341±0,0445	<b>0,9310</b>	0,8416±0,0517
	8	0,8096	0,7376±0,0349	0,6932	0,6137±0,0575	0,5939	0,5484±0,0365	<b>0,8908</b>	0,8059±0,0543
	16	0,8159	0,7469±0,0383	0,6613	0,5945±0,0371	0,6402	0,5600±0,0370	<b>0,9223</b>	0,8433±0,0585

Tabela 8 – Melhores indivíduos com média harmônica encontrados por AEMO em um DAG de 100 tarefas.

Obj.	$m$	AEMMT2		AEMMD		NSGA-II		NSGA-III	
		melhor	média	melhor	média	melhor	média	melhor	média
5	2	0,8827	0,8272±0,0296	0,8119	0,7709±0,0272	0,6348	0,5804±0,0295	<b>0,8983</b>	0,8363±0,0476
	4	0,8844	0,8139±0,0403	0,8606	0,7730±0,0431	0,5896	0,5494±0,0234	<b>1,0388</b>	0,9598±0,0476
	8	0,8706	0,8090±0,0453	0,8864	0,8204±0,0411	0,5748	0,5528±0,0189	<b>1,0801</b>	1,0127±0,0485
	16	0,9034	0,8662±0,0231	0,8803	0,8473±0,0196	0,6654	0,6174±0,0256	<b>1,0237</b>	0,9942±0,0177
4	2	0,9086	0,8542±0,0227	0,8548	0,8017±0,0312	0,6151	0,5823±0,0261	<b>0,9369</b>	0,9118±0,0251
	4	0,8959	0,8482±0,0336	0,8502	0,7988±0,0398	0,5892	0,5325±0,0263	<b>1,0272</b>	1,0013±0,0201
	8	0,8861	0,8365±0,0267	0,8506	0,8170±0,0238	0,5972	0,5527±0,0338	<b>1,0468</b>	1,0147±0,0195
	16	0,9028	0,8696±0,0224	0,8834	0,8433±0,0305	0,6744	0,6309±0,0307	<b>1,0416</b>	1,0075±0,0246
3	2	0,9727	0,9258±0,0400	0,9396	0,8683±0,0549	0,6655	0,6097±0,0364	<b>1,0104</b>	0,9936±0,0108
	4	0,8477	0,7911±0,0583	0,7508	0,6740±0,0651	0,5472	0,4911±0,0281	<b>1,1174</b>	1,0500±0,0464
	8	0,8703	0,8077±0,0469	0,7753	0,7003±0,0444	0,5435	0,5032±0,0358	<b>1,0717</b>	1,0021±0,0517
	16	0,8785	0,8486±0,0252	0,8409	0,7614±0,0617	0,6358	0,5804±0,0350	<b>1,0259</b>	0,9858±0,0306
2	2	0,9690	0,9102±0,0391	0,7892	0,7098±0,0582	0,7014	0,6455±0,0379	<b>1,0000</b>	0,9827±0,0196
	4	0,8741	0,7862±0,0595	0,7565	0,6177±0,0626	0,5831	0,5303±0,0283	<b>0,8901</b>	0,8282±0,0319
	8	0,8423	0,7751±0,0400	0,6379	0,5533±0,0492	0,5770	0,5277±0,0381	<b>0,8492</b>	0,8002±0,0423
	16	0,8440	0,8149±0,0282	0,7904	0,6644±0,0735	0,6725	0,5961±0,0388	<b>0,9078</b>	0,8595±0,0398

Tabela 9 – Melhores indivíduos com média harmônica encontrados por AEMO em um DAG de 300 tarefas.

Obj.	m	AEMMT2		AEMMD		NSGA-II		NSGA-III	
		melhor	média	melhor	média	melhor	média	melhor	média
5	2	0,9622	0,8220±0,0574	0,8499	0,7956±0,0472	0,5491	0,5315±0,0172	<b>1,0389</b>	0,9419±0,0581
	4	0,8021	0,7375±0,0461	0,7959	0,6806±0,0625	0,4986	0,4891±0,0086	<b>0,9866</b>	0,9344±0,0478
	8	0,8725	0,8095±0,0365	0,8587	0,8304±0,0279	0,5425	0,5149±0,0167	<b>1,1065</b>	1,0359±0,0597
	16	0,8791	0,7674±0,0559	0,8320	0,7515±0,0445	0,5132	0,4953±0,0134	<b>1,0645</b>	1,0122±0,0423
4	2	0,8750	0,8369±0,0292	0,8508	0,7823±0,0417	0,5255	0,4728±0,0334	<b>1,0637</b>	0,9950±0,0441
	4	0,8792	0,8229±0,0402	0,8200	0,7396±0,0456	0,4244	0,3778±0,0207	<b>1,0916</b>	0,9721±0,0827
	8	0,9189	0,8349±0,0567	0,8657	0,7744±0,0522	0,4546	0,3719±0,0315	<b>1,1434</b>	0,9988±0,0715
	16	0,8768	0,8162±0,0309	0,8158	0,7746±0,0309	0,4414	0,4251±0,0121	<b>1,0999</b>	1,0348±0,0343
3	2	0,8882	0,8658±0,0156	0,8536	0,6994±0,0757	0,5064	0,4438±0,0393	<b>1,1232</b>	1,0785±0,0215
	4	0,8565	0,7392±0,0483	0,7005	0,5527±0,0714	0,3742	0,3266±0,0206	<b>1,0074</b>	0,9420±0,0425
	8	0,8676	0,7512±0,0641	0,6947	0,5762±0,0881	0,3974	0,3146±0,0311	<b>1,0773</b>	0,9123±0,0898
	16	0,8566	0,7481±0,0529	0,6700	0,5962±0,0667	0,3792	0,3638±0,0117	<b>1,1033</b>	0,9286±0,0944
2	2	0,9460	0,8400±0,0619	0,6674	0,5293±0,0805	0,5398	0,4814±0,0329	<b>1,1010</b>	1,0025±0,0635
	4	0,8166	0,7174±0,0634	0,5975	0,4708±0,0671	0,3999	0,3632±0,0250	<b>0,8772</b>	0,7607±0,0528
	8	0,8331	0,7210±0,0783	0,5443	0,4469±0,0704	0,4503	0,3481±0,0372	<b>0,9010</b>	0,7425±0,0795
	16	0,7837	0,7366±0,0353	0,6077	0,5339±0,0639	0,4513	0,4171±0,0209	<b>0,9152</b>	0,7957±0,0668

Nos gráficos de abaixo, pode-se observar novamente a tendência do NSGA-III de piorar suas soluções conforme o número de objetivos diminui. Porém, mesmo os resultados enfatizando que há uma piora em seus indivíduos em poucos objetivos, o NSGA-III ainda conseguiu produzir melhores valores que os outros AEMOs, o que demonstra sua eficácia.

Figura 29 – Tendência do melhor indivíduo por AEMO no DAG de 50 tarefas com redução de objetivos - Média Harmônica

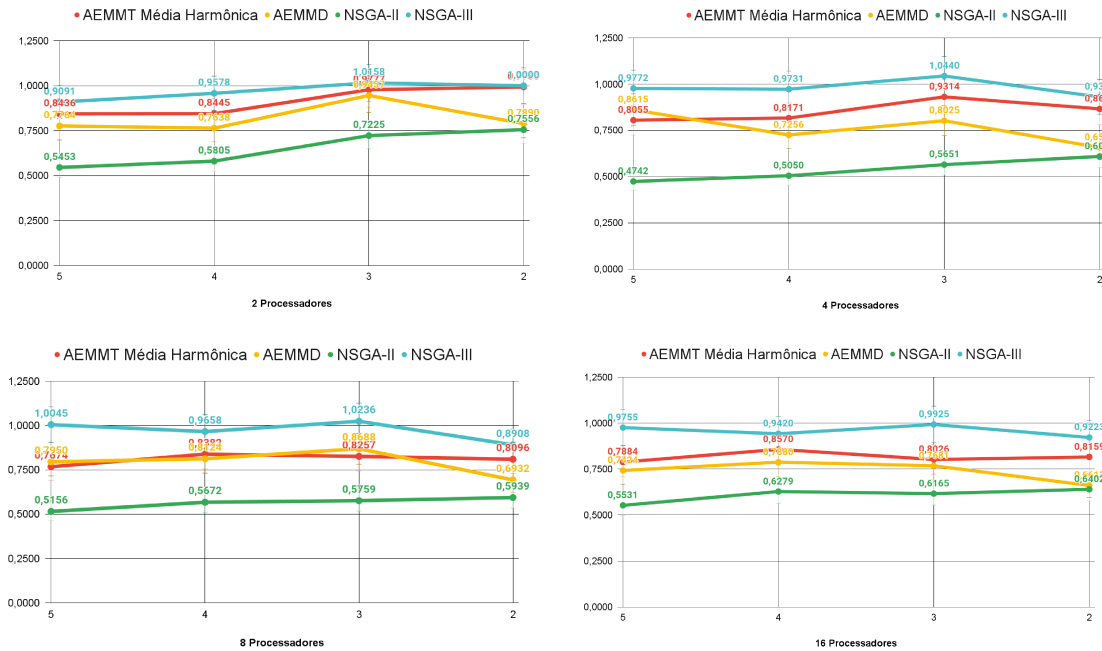


Figura 30 – Tendência do melhor indivíduo por AEMO no DAG de 100 tarefas com redução de objetivos - Média Harmônica

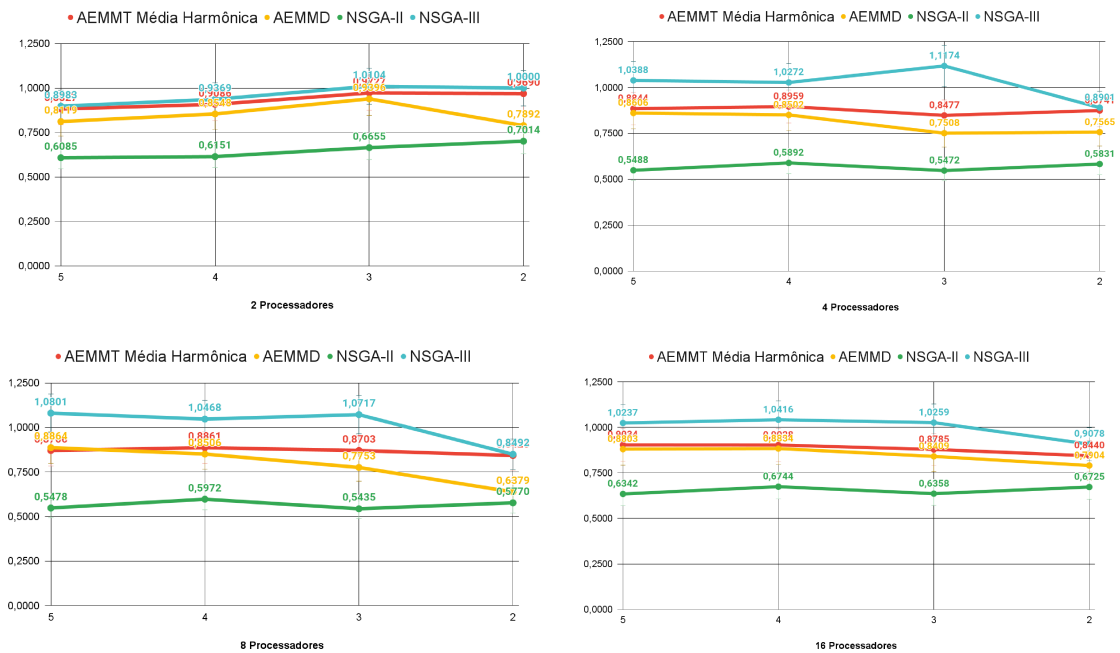
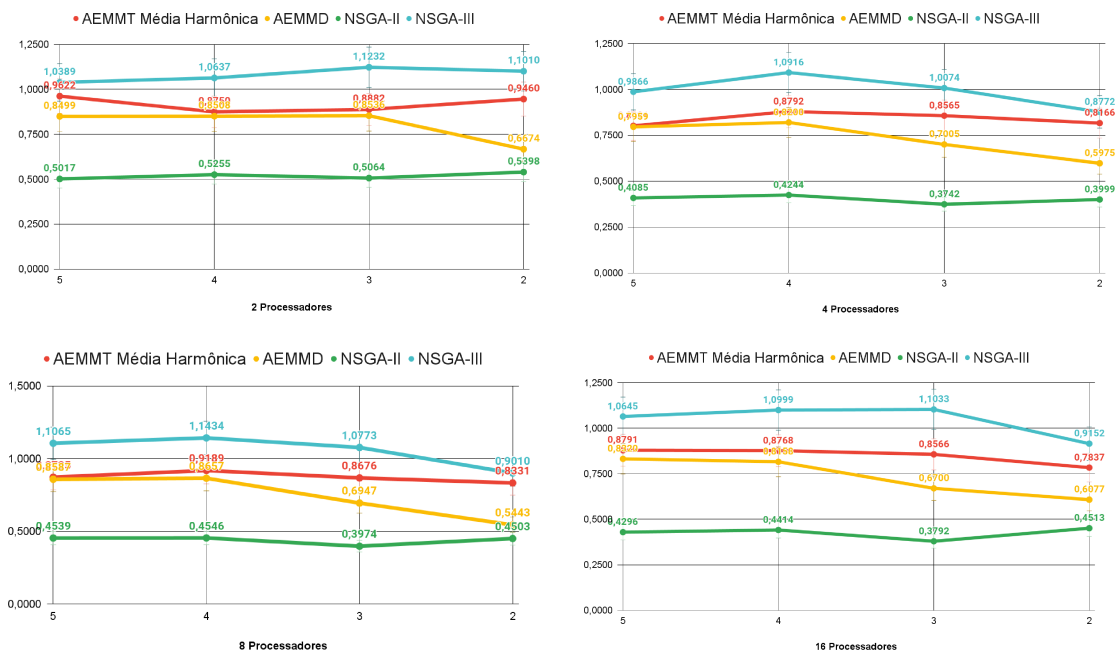


Figura 31 – Tendência do melhor indivíduo por AEMO no DAG de 300 tarefas com redução de objetivos - Média Harmônica



### 5.2.3.3 Hiper-Volume

Complementando o trabalho de Santos (2023), o hiper-volume foi adicionado ao trabalho. Uma métrica que mede o volume ocupado pelos indivíduos no espaço objetivo do problema, utilizada para avaliar a qualidade de um AEMO.

Normalmente, o hiper-volume é calculado para um conjunto de soluções, no entanto, visto a diferença na quantidade de indivíduos não-dominados entre cada um dos algoritmos, adotou-se uma abordagem diferente. Foi calculado o hiper-volume para cada uma das soluções individualmente, baseado nos valores dos objetivos.

Para obtenção dos valores, foi utilizado as seguintes *seeds* como parâmetros - 1995383100, -1989726638 e -1949695351, diferentemente do restante do trabalho que utiliza 10 *seeds* para a execução dos algoritmos. A lógica é dada por: *seed* x (*algoritmo* x *DAG* x *objetivo* x *processador*), e foi selecionado o melhor valor entre todas as soluções geradas. A disposição dos dados para o DAG de 50 tarefas pode ser observado na Tabela 10.

Tabela 10 – Hiper-Volume das melhores soluções encontradas no DAG de 50 tarefas.

Obj.	<i>m</i>	AEMMT1 melhor	AEMMT2 melhor	AEMMD melhor	NSGA-II melhor	NSGA-III melhor
5	2	<b>6.411,32</b>	6.117,32	6.157,32	5.278,27	6.384,29
	4	5.925,43	5.903,40	5.784,40	4.561,36	<b>6.133,36</b>
	8	5.421,49	5.213,44	5.355,34	4.498,41	<b>5.862,42</b>
	16	4.486,58	4.483,50	4.447,57	3.916,27	<b>4.876,47</b>
4	2	4.834,32	4.634,32	4.707,31	4.088,27	<b>4.939,32</b>
	4	4.784,42	4.538,38	4.437,38	3.682,31	<b>4.899,38</b>
	8	4.588,45	4.565,55	4.484,44	3.790,41	<b>4.843,44</b>
	16	3.807,77	3.949,69	3.886,24	3.450,27	<b>4.078,60</b>
3	2	4.220,30	4.200,32	4.129,30	3.573,27	<b>4.327,32</b>
	4	3.827,41	4.050,42	3.671,38	3.208,31	<b>4.329,41</b>
	8	3.926,52	3.822,50	3.764,45	3.209,41	<b>4.128,51</b>
	16	3.037,66	3.176,64	3.111,56	2.828,27	<b>3.394,64</b>
2	2	149,32	<b>153,32</b>	122,32	118,27	150,32
	4	138,45	142,45	122,38	115,36	<b>152,44</b>
	8	<b>137,53</b>	125,57	114,54	107,56	130,59
	16	104,75	103,75	80,63	86,55	<b>107,85</b>

O NSGA-III prevaleceu aproximadamente 81% da vezes, seguido pelo AEMMT de média simples (AEMMT1). Vale notar também, que o AEMMT1 e o AEMMT2 (média harmônica) apresentaram valores muito próximos em quase todas as execuções.

Nos DAGs de 100 e 300 tarefas os resultados foram parecidos, com o NSGA-III dominando 93,75% das vezes. No entanto, é possível observar um comportamento interessante, conforme o número de processadores e objetivos aumenta, a diferença entre os valores do NSGA-III e dos outros algoritmos também aumenta. Como é demonstrado na Tabela 11 e Tabela 12, caso comparado a execução de dois processadores com dezesseis.

Tabela 11 – Hiper-Volume das melhores soluções encontradas no DAG de 100 tarefas.

Obj.	$m$	AEMMT1 melhor	AEMMT2 melhor	AEMMD melhor	NSGA-II melhor	NSGA-III melhor
5	2	34.885,12	34.491,13	34.043,13	27.894,12	<b>35.047,13</b>
	4	33.802,22	33.636,21	33.961,20	25.841,19	<b>37.224,22</b>
	8	33.490,28	33.204,25	34.422,29	26.955,18	<b>36.723,27</b>
	16	31.523,41	31.044,42	31.412,34	28.339,34	<b>34.002,45</b>
4	2	31.014,13	31.211,13	30.123,13	24.860,11	<b>31.653,11</b>
	4	31.797,21	31.335,22	30.866,22	23.576,19	<b>34.152,20</b>
	8	30.977,29	30.583,28	30.821,27	25.299,18	<b>33.901,29</b>
	16	28.819,44	28.733,44	28.445,42	26.418,34	<b>31.413,40</b>
3	2	26.309,13	26.540,11	25.571,13	21.178,12	<b>26.958,13</b>
	4	24.507,22	24.736,21	22.448,22	18.048,19	<b>28.511,16</b>
	8	24.071,26	24.953,26	24.095,24	19.561,18	<b>27.222,28</b>
	16	22.048,45	22.341,42	21.906,44	20.157,34	<b>24.020,42</b>
2	2	457,13	482,13	371,13	356,12	<b>499,13</b>
	4	<b>495,23</b>	448,23	458,20	338,18	456,23
	8	467,31	462,29	403,27	351,18	<b>488,28</b>
	16	347,47	330,43	334,42	310,34	<b>352,46</b>

Tabela 12 – Hiper-Volume das melhores soluções encontradas no DAG de 300 tarefas.

Obj.	$m$	AEMMT1 melhor	AEMMT2 melhor	AEMMD melhor	NSGA-II melhor	NSGA-III melhor
5	2	394.601,04	413.758,04	393.290,03	265.234,04	<b>426.621,04</b>
	4	393.134,14	384.621,14	371.778,14	261.542,11	<b>443.779,14</b>
	8	350.745,12	358.393,13	353.307,14	244.767,10	<b>399.300,13</b>
	16	297.405,17	296.217,17	304.449,14	207.732,17	<b>334.929,14</b>
4	2	341.762,04	313.324,03	309.762,03	215.825,04	<b>356.626,01</b>
	4	341.368,14	337.153,14	298.591,13	214.008,11	<b>375.557,14</b>
	8	297.774,14	302.948,15	298.864,16	206.660,10	<b>333.915,16</b>
	16	264.339,19	260.691,15	253.834,15	179.465,17	<b>283.011,20</b>
3	2	288.413,03	294.476,04	300.573,04	204.994,04	<b>343.339,04</b>
	4	287.082,13	277.782,13	257.404,13	198.576,11	<b>341.176,14</b>
	8	286.055,14	285.124,15	247.872,15	190.153,10	<b>315.152,15</b>
	16	228.600,20	241.983,19	208.988,19	160.765,17	<b>257.065,22</b>
2	2	2.131,03	2.056,04	1.602,04	1.214,04	<b>2.467,04</b>
	4	1.981,13	2.200,14	1.667,13	1.325,14	<b>2.618,15</b>
	8	1.746,15	<b>1.871,15</b>	1.406,12	988,14	1.835,15
	16	1.611,23	1.652,19	1.462,16	1.115,19	<b>1.733,22</b>





## 6 Conclusão e Trabalhos Futuros

Com o avanço dos anos, o número de usuários simultâneos nos sistemas cresce cada vez mais, e com isso a demanda por sistemas capazes de lidar com uma grande quantidade de processos ao mesmo tempo. No entanto, otimizar a execução destes processos pode se tornar uma tarefa extremamente complexa e onerosa, visto que muitas vezes os critérios de otimização são conflitantes entre si.

Para atender este cenário, é necessário uma ferramenta capaz de otimizar múltiplos critérios, conflitantes ou não, simultaneamente. Com isso, surgiu a ideia de aplicar neste tipo problema, algoritmos computacionais baseados na evolução, em que boas soluções servem como base (material genético) para outras soluções em gerações futuras. Este tipo de algoritmo é conhecido como AEMO (algoritmo evolutivo multi-objetivo).

Com isto, surgem estudos que buscam observar e comparar os resultados entre diferentes AEMOs, a fim de descobrir qual deles é melhor. Contudo, a resposta não é tão simples, uma vez que cada AEMO é desenvolvido visando um cenário específico, geralmente relacionado ao número objetivos sendo otimizados simultaneamente, e variar este número pode alterar completamente a eficácia de um AEMO.

Como um dos vários estudos na área, Santos (2023) contempla e compara diversos AEMOs no contexto do escalonamento de tarefas em múltiplos processadores, sendo eles: o AEMMT (Algoritmo Evolutivo Multi-objetivo com Múltiplas Tabelas) de média simples, o AEMMT de média harmônica, o AEMMD (Algoritmo Evolutivo Multi-objetivo com Múltiplas Dominâncias) e o NSGA-II (*Non-dominated Sorting Genetic Algorithm II*).

Sendo assim, este trabalho teve como objetivo adicionar o NSGA-III (*Non-dominated Sorting Genetic Algorithm III*) ao estudo de Santos (2023). Busca-se com isto, ampliar ainda mais a comparação entre diferentes AEMOs, tendo em vista que, ao serem executados no mesmo ambiente e com os mesmos parâmetros, cria-se uma comparação muito mais assertiva.

O NSGA-III surgiu como uma evolução de seu predecessor, o qual era limitado a problemas de até no máximo dois objetivos. Ao contrário do NSGA-II que usava um sistema baseado na distância de aglomeração (*crowding distance*), o NSGA-III é baseado em pontos de referência. Este conjunto de referências permitem uma melhor distribuição das soluções na Fronteira de Pareto, independente da quantidade de objetivos. Assim, o NSGA-III surgiu para atender problemas de alta complexidade, em que é necessário otimizar dezenas de critérios simultaneamente.

Ao longo do estudo, foram utilizadas quatro métricas para comparação entre os

AEMOs, sendo elas: a quantidade de soluções produzidas em intervalos com bons resultados, a média simples, a média harmônica e o hiper-volume. Através das tabelas e imagens apresentadas ao longo do capítulo de resultados, foi demonstrado que o NSGA-III foi superior aos outros algoritmos em quase todas as métricas aplicadas, sendo inferior apenas ao AEMMT de média simples na execução do DAG de 100 tarefas com 2 objetivos e 4 processadores.

No entanto, por mais que tenha demonstrado resultados melhores, é necessário ressaltar que quanto menor o número de objetivos e processadores, menor foi a tendência do NSGA-III de produzir o melhor indivíduo. Isto fica demonstrado nas comparações de média simples e média harmônica, em que o AEMMT de média correspondente produziu soluções com a qualidade muito similar à do NSGA-III.

Além disto, a comparação por hiper-volume também deixou clara a preferência por problemas complexos do NSGA-III. Foi observado ao longo do estudo que, quanto maior a complexidade do problema, maior foi a diferença nos resultados entre o NSGA-III e os outros AEMOs.

No contexto da otimização multi-objetivo, cinco objetivos ainda é considerado uma quantidade baixa. Portanto, como continuação deste trabalho, seria interessante ampliar o número de objetivos. Problemas de dez objetivos ou mais, por exemplo, fariam com que os AEMOs fossem forçados ao limite, o enfatizaria ainda mais os pontos positivos e negativos de cada um dos algoritmos.

Existem também, diversos outros AEMOs amplamente estudados no cenário da computação, e a implementação de um deles pode ser uma ótima forma de continuar este estudo. Como boas escolhas, tem-se o SPEA-II (*Strength Pareto Evolutionary Algorithm II*), um AEMO baseado na força de dominância de Pareto, ou seja, na capacidade de dominação entre as soluções. Há também o MOEAD/D (*Multi-Objective Evolutionary Algorithm based on Decomposition*), que é um algoritmo baseado na decomposição, o qual decompõe um problema multi-objetivo em vários outros problemas de apenas um objetivo.

Outro estudo interessante para um trabalho futuro, seria a migração do AGScheduling de Java 8 para Java 17 e a adaptação das classes CORE do algoritmo para contemplar a arquitetura da biblioteca JMetal. Devido às diferenças entre as versões da biblioteca, o processo de implementação envolveu ajustar manualmente o código relacionado ao NSGA-III. A refatoração do AGScheduling para Java 17, estendendo classes chave do JMetal, permitiria acesso a uma ampla gama de algoritmos, problemas famosos de otimização multi-objetivo e ferramentas de indicadores de qualidade presentes na biblioteca. Contudo, é necessário um esforço considerável para remodelar a arquitetura do algoritmo e dos cromossomos conforme os padrões exigidos para a integração.



# Referências

ARABNEJAD, H.; BARBOSA, J. G. List scheduling algorithm for heterogeneous systems by an optimistic cost table. **IEEE Transactions on Parallel and Distributed Systems**, v. 25, n. 3, p. 682–694, 2014. Disponível em: <[10.1109/TPDS.2013.57](https://doi.org/10.1109/TPDS.2013.57)>. Citado na página 13.

BRASIL, C. R. S. **Algoritmo evolutivo de muitos objetivos para predição ab initio de estrutura de proteínas**. Tese (Doutorado) — Universidade de São Paulo, São Carlos, SP, maio 2012. Disponível em: <<https://doi.org/10.11606/t.55.2012.tde-20072012-163056>>. Citado na página 15.

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, v. 34, n. 20, p. 1719–1734, maio 2013. Disponível em: <<https://doi.org/10.1002/jcc.23315>>. Citado na página 15.

COELLO, C. A. C.; LAMONT, G. B.; VELDHUIZEN, D. A. V. Strength pareto evolutionary algorithm 2: Theory and applications. **Computer Engineering and Design**, v. 28, n. 2, p. 149–168, 2007. Citado na página 15.

DEB, K.; JAIN, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. **IEEE Transactions on Evolutionary Computation**, v. 18, n. 4, p. 577–601, 2014. Disponível em: <[10.1109/TEVC.2013.2281535](https://doi.org/10.1109/TEVC.2013.2281535)>. Citado 3 vezes nas páginas 18, 27 e 28.

DEB, K.; KALYANMOY, D. **Multi-Objective Optimization Using Evolutionary Algorithms**. USA: John Wiley & Sons, Inc., 2001. ISBN 047187339X. Citado 2 vezes nas páginas 9 e 15.

DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast elitist multi-objective genetic algorithm: NSGA-II. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 6, n. 2, p. 182–197, 2002. Citado 2 vezes nas páginas 15 e 16.

GABRIEL, P. H. R.; MELO, V. V. de; DELBEM, A. C. B. Algoritmos evolutivos e modelo HP para predição de estruturas de proteínas. **Revista Controle & Automação**, v. 23, n. 1, p. 25–37, jan. 2012. Disponível em: <<https://doi.org/10.1590/S0103-17592012000100003>>. Citado na página 15.

HOLLAND, J. H. Outline for a logical theory of adaptive systems. **J. ACM**, Association for Computing Machinery, New York, NY, USA, v. 9, n. 3, p. 297–314, jul 1962. ISSN 0004-5411. Disponível em: <<https://doi.org/10.1145/321127.321128>>. Citado na página 13.

ISHIBUCHI, H.; IMADA, R.; SETOGUCHI, Y.; NOJIMA, Y. Performance comparison of nsga-ii and nsga-iii on various many-objective test problems. In: **2016 IEEE Congress on Evolutionary Computation (CEC)**. Vancouver: [s.n.], 2016. p. 3045–3052. Disponível em: <[10.1109/CEC.2016.7744174](https://doi.org/10.1109/CEC.2016.7744174)>. Citado 3 vezes nas páginas 9, 22 e 41.

KNOWLES, J. D.; CORNE, D. W. The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. In: **Proceedings of the 1999 Congress on Evolutionary Computation**. USA: IEEE, 1999. p. 85–105. Disponível em: <<https://doi.org/10.1109/CEC.1999.781913>>. Citado na página 15.

LAFETÁ, T.; BUENO, M. L. P.; BRASIL, C. R. S.; OLIVEIRA, G. M. B. MEANDS: A many-objective evolutionary algorithm based on non-dominated decomposed sets applied to multicast routing. **Applied Soft Computing**, Elsevier BV, v. 62, p. 851–866, jan 2018. Disponível em: <<https://doi.org/10.1016/j.asoc.2017.09.017>>. Citado na página 16.

NEBRO, A. J.; DURILLO, J. J.; VERGNE, M. Redesigning the jMetal multi-objective optimization framework. In: **Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation**. New York, NY, USA: Association for Computing Machinery, 2015. (GECCO Companion '15), p. 1093–1100. ISBN 9781450334884. Disponível em: <<https://doi.org/10.1145/2739482.2768462>>. Citado na página 23.

OMARA, F. A.; ARAFA, M. M. Genetic algorithms for task scheduling problem. **Journal of Parallel and Distributed Computing**, v. 70, n. 1, p. 13–22, 2010. ISSN 0743-7315. Disponível em: <<https://doi.org/10.1016/j.jpdc.2009.09.009>>. Citado 2 vezes nas páginas 9 e 13.

ROBERT, Y. Task graph scheduling. In: PADUA, D. (Ed.). **Encyclopedia of Parallel Computing**. Boston, MA: Springer US, 2011. p. 2013–2025. Disponível em: <[https://doi.org/10.1007/978-0-387-09766-4\\_42](https://doi.org/10.1007/978-0-387-09766-4_42)>. Citado na página 9.

SANTOS, A. C.; DELBEM, A. C. B.; LONDON JR., J. B. A.; BRETAS, N. G. Node-depth encoding and multiobjective evolutionary algorithm applied to large-scale distribution system reconfiguration. **IEEE Transactions on Power Systems**, Institute of Electrical and Electronics Engineers (IEEE), v. 25, n. 3, p. 1254–1265, ago. 2010. Disponível em: <<https://doi.org/10.1109/tpwrs.2010.2041475>>. Citado na página 15.

SANTOS, J. F. **Algoritmos evolutivos multiobjetivo baseados em tabelas para escalonamento de tarefas em ambientes multiprocessados**. 111 p. Dissertação (Mestrado) — Universidade Federal de Uberlândia, 2023. Disponível em: <<http://doi.org/10.14393/ufu.di.2023.163>>. Citado 14 vezes nas páginas 2, 9, 10, 13, 21, 23, 24, 25, 30, 31, 32, 43, 49 e 53.

SECARA, I.-A. Challenges and considerations in developing and architecting large-scale distributed systems. **International Journal of Internet and Distributed Systems**, Scientific Research Publishing, Inc., v. 04, n. 01, p. 1–13, 2020. Disponível em: <[10.4236/ijids.2020.41001](https://doi.org/10.4236/ijids.2020.41001)>. Citado na página 9.

SILVA, E. C. da. **Representações de algoritmos genéticos para o problema de escalonamento estático de tarefas em multiprocessadores**. 231 p. Dissertação (Mestrado) — Universidade Federal de Uberlândia, 2020. Disponível em: <<http://doi.org/10.14393/ufu.di.2020.104>>. Citado 4 vezes nas páginas 9, 13, 21 e 22.

SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. **Evolutionary Computation**, v. 2, n. 3, p. 221–248, set. 1995. Disponível em: <<https://doi.org/10.1162/evco.1994.2.3.221>>. Citado na página 15.