
Algoritmos Evolutivos para a Otimização Dinâmica de um Problema Discreto com muitos Objetivos

Thiago Fialho de Queiroz Lafetá



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uberlândia
2025

Thiago Fialho de Queiroz Lafetá

**Algoritmos Evolutivos para a Otimização
Dinâmica de um Problema Discreto com muitos
Objetivos**

Tese de doutorado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Luiz Gustavo Almeida Martins

Uberlândia
2025



ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Tese, 23/2025, PPGCO				
Data:	31 de Julho de 2025	Hora de início:	08:00	Hora de encerramento:	11:10
Matrícula do Discente:	11723CCP006				
Nome do Discente:	Thiago Fialho de Queiroz Lafetá				
Título do Trabalho:	Algoritmos Evolutivos para a Otimização Dinâmica de um Problema Discreto com muitos Objetivos				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Inteligência Artificial				
Projeto de Pesquisa de vinculação:	Computação Bio-inspirada e Aprendizagem de Máquina na Resolução de Problemas				

Reuniu-se por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Paulo Henrique Ribeiro Gabriel - FACOM/UFU, Márcia Aparecida Fernandes - FACOM/UFU, Myriam Regattieri De Biase da Silva Delgado -CPGEI/UTFPR, André Britto de Carvalho - DComp/UFS e Luiz Gustavo Almeida Martins - FACOM/UFU, orientador do candidato.

Os examinadores participaram desde as seguintes localidades: Myriam Regattieri De Biase da Silva Delgado - Curitiba/PR, André Britto de Carvalho - São Cristóvão/SE. Os outros membros da banca e o aluno participaram da cidade de Uberlândia.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Luiz Gustavo Almeida Martins, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação da Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir ao candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado

Esta defesa faz parte dos requisitos necessários à obtenção do título de Doutor.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação

interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Márcia Aparecida Fernandes, Professor(a) do Magistério Superior**, em 04/08/2025, às 10:20, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Myriam Regattieri De Biase da Silva Delgado, Usuário Externo**, em 04/08/2025, às 11:35, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Paulo Henrique Ribeiro Gabriel, Professor(a) do Magistério Superior**, em 04/08/2025, às 13:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **André Britto de Carvalho, Usuário Externo**, em 04/08/2025, às 14:19, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Luiz Gustavo Almeida Martins, Professor(a) do Magistério Superior**, em 05/08/2025, às 15:51, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **6488745** e o código CRC **E24CA7D0**.

Dedico aos meus orientadores, Profa. Dra. Gina Maira Barbosa de Oliveira (in memoriam) e Prof. Dr. Luiz Gustavo Almeida Martins, pela confiança, paciência, incentivo e excelente orientação. Sem o apoio de ambos, este trabalho não teria sido realizado. A eles, muito obrigado.

Dedico aos meus irmãos José Antônio Jr. e Raquel Fialho, por sempre estarem presentes.

Dedico aos meus Pais José Antônio de Queiroz e Regina Coelli Fialho.

Agradecimentos

À Professora Dra. Gina Oliveira, minha orientadora e mentora, com profunda gratidão e admiração. A sua paciência, sabedoria e dedicação foram fundamentais para o meu crescimento acadêmico e pessoal. Obrigado por acreditar em mim, por me guiar, e por me mostrar o verdadeiro valor da ciência e do compromisso com o conhecimento. Esta conquista é, em grande parte, reflexo do seu exemplo e apoio.

Meu sincero agradecimento ao Professor Dr. Luiz Gustavo Almeida Martins. Inicialmente como coorientador e, posteriormente, assumindo a orientação deste trabalho, o senhor foi uma presença constante, oferecendo apoio, conhecimento e dedicação em todas as etapas deste percurso. Agradeço por ter acolhido este projeto com tanto empenho e por ter feito de tudo para que esta jornada chegasse ao fim da melhor forma possível.

À coordenação do Programa de Pós-Graduação em Ciência da Computação, registro meus agradecimentos sinceros. Sou grato pelo suporte institucional, pela organização do curso e pelo comprometimento com a formação acadêmica de qualidade. Agradeço também pela disponibilidade, pelo acolhimento e pelo esforço contínuo em manter um programa que valoriza a pesquisa, o desenvolvimento e o crescimento de seus alunos.

Meus agradecimentos à CAPES pelo apoio institucional e incentivo à pesquisa e à formação acadêmica no Brasil. O suporte da CAPES foi fundamental para o desenvolvimento deste trabalho, contribuindo diretamente para a continuidade dos estudos e a consolidação do conhecimento científico. Agradeço pelo compromisso com a educação de qualidade e pelo investimento constante na formação de pesquisadores e profissionais altamente qualificados.

Agradeço à minha família, meu alicerce e refúgio. Obrigado pelo amor incondicional, pelas palavras de encorajamento e por acreditarem em mim mesmo nos momentos em que eu duvidei.

Agradeço a Deus, fonte de força, sabedoria e esperança, agradeço por ter me sustentado em cada etapa desta caminhada.

O importante é o que importa.
(Rafa Marquez, Alexandre Scher, Rô o Pensador, Tarcísio Lourençon, Eu, ...)

Resumo

Vários problemas de otimização do mundo real são dinâmicos e envolvem múltiplos objetivos. Diferentes pesquisas usando algoritmos evolutivos focam nestas características de forma individual, mas poucos trabalhos investigam problemas que são dinâmicos e multiobjetivo ao mesmo tempo. Trabalhos recentes investigam problemas dinâmicos de otimização multiobjetivo (DMOPs), o que adiciona um desafio a mais à convergência da busca. Embora amplamente exploradas em formulações com múltiplos objetivos para problemas estáticos, as abordagens evolutivas ainda são desafiadas pelos DMOPs definindo um tópico de pesquisa relevante. Algumas estratégias evolutivas para DMOPs surgiram da adaptação de algoritmos multiobjetivo previamente criados para solucionar problemas estáticos de otimização contínua. Neste trabalho são propostos novos algoritmos evolutivos multiobjetivo dinâmicos (DMOEAs), a saber: D-MEANDS, D-MEANDS-MD, D-MEANDS-II e D-MEANDS-III. Esses algoritmos são baseados nos algoritmos MEANDS e MEANDS-II, originalmente propostos para tratar problemas estáticos e discretos, e incorporam estratégias dinâmicas baseadas em memória e diversidade para lidar com otimizações dinâmicas. Também são investigados novos mecanismos evolutivos que visam trabalhar, de forma mais eficiente, com as subpopulações empregadas nas abordagens propostas. Uma versão dinâmica multiobjetivo do problema da mochila, conhecida como *Dynamic Multiobjective Knapsack Problem* (DMKP), é utilizada para avaliar o desempenho dos diferentes algoritmos. As instâncias do DMKP são sujeitas a variações de ambientes ao longo da execução da otimização evolutiva, sendo submetidos a até 20 mudanças de ambiente ao longo da evolução. O comportamento de tais algoritmos foram avaliados em instâncias dinâmicas de até 8 objetivos. Resultados experimentais mostraram que os algoritmos propostos competem com DMOEAs da literatura, alcançando desempenho superior nas métricas multiobjetivo avaliadas, na maioria dos cenários investigados.

Palavras-chave: problemas multiobjetivo dinâmicos, algoritmos evolutivos *many-objective*, problema da mochila.

Abstract

Many real-world optimization problems are dynamic and involve multiple objectives. Different studies using evolutionary algorithms focus on these characteristics individually, but few studies investigate problems that are dynamic and multi-objective at the same time. Recent studies investigate dynamic multi-objective optimization problems (DMOPs), which adds an additional challenge to the convergence of the search. Although widely explored in multi-objective formulations for static problems, evolutionary approaches are still challenged by DMOPs defining a relevant research topic. Some evolutionary strategies for DMOPs emerged from the adaptation of multi-objective algorithms previously created to solve static continuous optimization problems. In this work, we propose new dynamic multiobjective evolutionary algorithms (DMOEAs), namely: D-MEANDS, D-MEANDS-MD, D-MEANDS-II and D-MEANDS-III. These algorithms are based on the MEANDS and MEANDS-II algorithms, originally proposed to solve static and discrete problems, and incorporate dynamic strategies based on memory and diversity to deal with dynamic optimizations. We also investigate new evolutionary mechanisms that aim to work more efficiently with the subpopulations employed in the proposed approaches. A dynamic multiobjective version of the knapsack problem, known as Dynamic Multiobjective Knapsack Problem (DMKP), is used to evaluate the performance of the different algorithms. The DMKP instances are subject to environmental variations throughout the execution of the evolutionary optimization, being subjected to up to 20 environmental changes throughout the evolution. The behavior of such algorithms was evaluated in dynamic instances with up to 8 objectives. Experimental results showed that the proposed algorithms compete with DMOEAs from the literature, achieving superior performance in the evaluated multi-objective metrics, in most of the investigated scenarios.

Keywords: dynamic multiobjective problems, many-objective evolutionary algorithms, Knapsack Problem.

Lista de ilustrações

Figura 1 – Reinicialização da população após uma mudança de ambiente no DNSGA-II e DSNGA-III.	49
Figura 2 – Fluxo básico da estratégia evolutiva usada no MEANDS	50
Figura 3 – Primeira etapa da construção do filtro de soluções do método <i>Filtro-PR</i>	61
Figura 4 – Tempo de processamento para instâncias DMKP (EC=1): 30 e 50 itens.	67
Figura 5 – Tempo de processamento para instâncias DMKP (EC=2): 30 e 50 itens.	68
Figura 6 – Desempenho do DNSGA-II em função do tamanho populacional, considerando uma instância do problema com 30 itens e 8 objetivos.	69
Figura 7 – Desempenho do DNSGA-II* em função do tamanho populacional, considerando uma instância do problema com 30 itens e 8 objetivos.	70
Figura 8 – Tempo de execução do DNSGA-II, MS-MOEA e D-MEANDS em cenários com uma mudança de ambiente.	72
Figura 9 – Tempo de execução do DNSGA-II, MS-MOEA e D-MEANDS em cenários com duas mudança de ambiente.	73
Figura 10 – Tempo de execução do DNSGA-II, MS-MOEA e D-MEANDS em cenários com cinco e dez mudanças de ambiente.	74
Figura 11 – Tempo de execução do D-MEANDS em relação aos demais algoritmos em instâncias com 4 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).	77
Figura 12 – Tempo de execução do D-MEANDS em relação aos demais algoritmos em instâncias com 6 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).	78
Figura 13 – Tempo de execução do D-MEANDS em relação aos demais algoritmos em instâncias com 8 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).	78
Figura 14 – Tempo de execução do D-MEANDS-MD em relação aos demais algoritmos em instâncias com 4 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).	81

Figura 15 – Tempo de execução do D-MEANDS-MD em relação aos demais algoritmos em instâncias com 6 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).	82
Figura 16 – Tempo de execução do D-MEANDS-MD em relação aos demais algoritmos em instâncias com 8 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).	82
Figura 17 – Tempo de execução do D-MEANDS-II em relação aos demais algoritmos em instâncias com 4 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).	85
Figura 18 – Tempo de execução do D-MEANDS-II em relação aos demais algoritmos em instâncias com 6 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).	86
Figura 19 – Tempo de execução do D-MEANDS-II em relação aos demais algoritmos em instâncias com 8 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).	86
Figura 20 – Tempo de execução do D-MEANDS-III em relação aos demais algoritmos em instâncias com 4 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).	89
Figura 21 – Tempo de execução do D-MEANDS-III em relação aos demais algoritmos em instâncias com 6 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).	89
Figura 22 – Tempo de execução do D-MEANDS-III em relação aos demais algoritmos em instâncias com 8 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).	90
Figura 23 – Coordenadas paralelas das soluções encontradas por D-MEANDS-MD (linhas e pontos pretos) e DDIS-MOEA/D-DE (linhas e pontos azuis) em suas melhores execuções para a instância DMKP com 4 objetivos, 100 itens e EC=20.	94
Figura 24 – Coordenadas paralelas das soluções encontradas por (a) D-MEANDS e (b) D-MEANDS-MD em suas melhores execuções para a instância DMKP com 4 objetivos, 100 itens e EC=20.	95
Figura 25 – Fator de crescimento da quantidade de subpopulações entre D-MEANDS-MD e D-MEANDS-II.	96
Figura 26 – Evolução do tempo de execução do D-MEANDS-MD e D-MEANDS-II em função do número de objetivos do problema, considerando instâncias de 30 itens e 20 mudanças de ambiente.	97
Figura 27 – Evolução do tempo de execução do D-MEANDS-MD e D-MEANDS-II em função do número de objetivos do problema, considerando instâncias de 50 itens e 20 mudanças de ambiente.	97

Figura 28 – Evolução do tempo de execução do D-MEANDS-MD e D-MEANDS-II em função do número de objetivos do problema, considerando instâncias de 100 itens e 20 mudanças de ambiente.	98
Figura 29 – Evolução do tamanho populacional entre D-MEANDS-MD e D-MEANDS-II, considerando um limite de 100 indivíduos por subpopulação para o D-MEANDS-MD e 20 para o D-MEANDS-II.	99
Figura 30 – Comparação de desempenho do D-MEANDS-II e suas variantes para instâncias do DMKP com 4 objetivos (EC=20).	120
Figura 31 – Comparação de desempenho do D-MEANDS-II e suas variantes para instâncias do DMKP com 6 objetivos (EC=20).	121
Figura 32 – Comparação de desempenho do D-MEANDS-II e suas variantes para instâncias do DMKP com 8 objetivos (EC=20).	122
Figura 33 – Comparação de desempenho do D-MEANDS-II-S2 com D-MEANDS-II-T4I para instâncias do DMKP com 4 objetivos (EC=20).	124
Figura 34 – Comparação de desempenho do D-MEANDS-II-S2 com D-MEANDS-II-T4I para instâncias do DMKP com 6 objetivos (EC=20).	125
Figura 35 – Comparação de desempenho do D-MEANDS-II-S2 com D-MEANDS-II-T4I para instâncias do DMKP com 8 objetivos (EC=20).	126
Figura 36 – Comparação de desempenho do D-MEANDS-II-S2 com D-MEANDS-II-PR para instâncias do DMKP com 4 objetivos (EC=20).	128
Figura 37 – Comparação de desempenho do D-MEANDS-II-S2 com D-MEANDS-II-PR para instâncias do DMKP com 6 objetivos (EC=20).	129
Figura 38 – Comparação de desempenho do D-MEANDS-II-S2 com D-MEANDS-II-PR para instâncias do DMKP com 8 objetivos (EC=20).	130
Figura 39 – Resultados HVR e $\overline{IGD}$ para instâncias do DMKP com 30 itens (EC=1).	132
Figura 40 – Resultados HVR e $\overline{IGD}$ para instâncias do DMKP com 50 itens (EC=1).	133
Figura 41 – Resultados HVR e $\overline{IGD}$ para instâncias do DMKP com 30 itens (EC=2).	134
Figura 42 – Resultados HVR e $\overline{IGD}$ para instâncias do DMKP com 50 itens (EC=2).	134
Figura 43 – Resultados de HVR e $\overline{IGD}$ para instâncias DMKP com 30 itens, objetivos $m = 4$ e 6 (EC=1). Avaliando o D-MEANDS.	135
Figura 44 – Resultados de HVR e $\overline{IGD}$ para instâncias DMKP com 50 itens, objetivos $m = 4$ e 6 (EC=1). Avaliando o D-MEANDS.	136
Figura 45 – Resultados de HVR e $\overline{IGD}$ para instâncias DMKP com 30 itens, objetivos $m = 4$ e 6 (EC=2). Avaliando o D-MEANDS.	136
Figura 46 – Resultados de HVR e $\overline{IGD}$ para instâncias DMKP com 50 itens, objetivos $m = 4$ e 6 (EC=2). Avaliando o D-MEANDS.	137
Figura 47 – Resultados de HVR e $\overline{IGD}$ para instâncias DMKP com 30 e 50 itens, 4 objetivos ($m=4$) e 5 mudanças de ambiente (EC=5).	137

Figura 48 – Resultados de HVR e $\overline{IGD}$ para instâncias DMKP com 30 e 50 itens, 4 objetivos ($m=4$) e 10 mudanças de ambiente ($EC=10$).	138
Figura 49 – Resultados da métrica Density (AD) em instâncias de 4 objetivos. . . .	139
Figura 50 – Resultados da métrica Density (AD) em instâncias de 6 objetivos. . . .	140
Figura 51 – Resultados da métrica Density (AD) em instâncias de 8 objetivos. . . .	140
Figura 52 – Resultados da métrica Hipervolume (HV) em instâncias de 4 objetivos. . .	141
Figura 53 – Resultados da métrica Hipervolume (HV) em instâncias de 6 objetivos. . .	141
Figura 54 – Resultados da métrica Hipervolume (HV) em instâncias de 8 objetivos. . .	142
Figura 55 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS- MD em instâncias de 4 objetivos.	143
Figura 56 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS- MD em instâncias de 6 objetivos.	143
Figura 57 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS- MD em instâncias de 8 objetivos.	144
Figura 58 – Resultados da métrica Hipervolume (HV). Investigando a proposta D- MEANDS-MD em instâncias de 4 objetivos.	144
Figura 59 – Resultados da métrica Hipervolume (HV). Investigando a proposta D- MEANDS-MD em instâncias de 6 objetivos.	145
Figura 60 – Resultados da métrica Hipervolume (HV). Investigando a proposta D- MEANDS-MD em instâncias de 8 objetivos.	145
Figura 61 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS- MD em instâncias de 4 objetivos.	147
Figura 62 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS- MD em instâncias de 6 objetivos.	147
Figura 63 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS- MD em instâncias de 8 objetivos.	148
Figura 64 – Resultados da métrica Hipervolume (HV). Investigando a proposta D- MEANDS-MD em instâncias de 4 objetivos.	148
Figura 65 – Resultados da métrica Hipervolume (HV). Investigando a proposta D- MEANDS-MD em instâncias de 6 objetivos.	149
Figura 66 – Resultados da métrica Hipervolume (HV). Investigando a proposta D- MEANDS-MD em instâncias de 8 objetivos.	149
Figura 67 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS- III em instâncias de 4 objetivos.	151
Figura 68 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS- III em instâncias de 6 objetivos.	151
Figura 69 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS- III em instâncias de 8 objetivos.	152

Figura 70 – Resultados da métrica Hipervolume (HV). Investigando a proposta D-MEANDS-III em instâncias de 4 objetivos.	152
Figura 71 – Resultados da métrica Hipervolume (HV). Investigando a proposta D-MEANDS-III em instâncias de 6 objetivos.	153
Figura 72 – Resultados da métrica Hipervolume (HV). Investigando a proposta D-MEANDS-III em instâncias de 8 objetivos.	153

Lista de tabelas

Tabela 1	– Exemplo de distâncias entre 4 indivíduos de uma subpopulação.	58
Tabela 2	– Teste de hipótese comparando MS-MOEA aos demais algoritmos em instâncias com uma mudança de ambiente (EC=1).	66
Tabela 3	– Teste de hipótese comparando DMSG-II aos demais algoritmos em instâncias com uma mudança de ambiente (EC=1)	66
Tabela 4	– Teste de hipótese comparando MS-MOEA aos demais algoritmos em instâncias com uma mudança de ambiente (EC=2)	68
Tabela 5	– Teste de hipótese comparando DMSG-II aos demais algoritmos em instâncias com uma mudança de ambiente (EC=2)	68
Tabela 6	– Teste de hipótese comparando D-MEANDS aos demais algoritmos em cenários com uma mudança de ambiente (EC=1).	72
Tabela 7	– Teste de hipótese comparando D-MEANDS aos demais algoritmos em cenários com duas mudança de ambiente (EC=2).	73
Tabela 8	– Teste de hipótese comparando D-MEANDS aos demais algoritmos em cenários com cinco e dez mudanças de ambiente (EC= 5 e 10).	74
Tabela 9	– Teste de hipótese comparando D-MEANDS aos demais algoritmos em cenários com dez mudanças de ambiente (EC=10).	76
Tabela 10	– Teste de hipótese comparando D-MEANDS aos demais algoritmos em cenários com quinze mudanças de ambiente (EC=15).	76
Tabela 11	– Teste de hipótese comparando D-MEANDS aos demais algoritmos em cenários com vinte mudanças de ambiente (EC=20).	76
Tabela 12	– Teste de hipótese comparando D-MEANDS-MD aos demais algoritmos em instâncias com dez mudanças de ambiente (EC=10).	80
Tabela 13	– Teste de hipótese comparando D-MEANDS-MD aos demais algoritmos em instâncias com quinze mudanças de ambiente (EC=15).	80
Tabela 14	– Teste de hipótese comparando D-MEANDS-MD aos demais algoritmos em instâncias com vinte mudanças de ambiente (EC=20).	80

Tabela 15 – Teste de hipótese comparando D-MEANDS-II aos demais algoritmos em instâncias com dez mudanças de ambiente (EC=10).	84
Tabela 16 – Teste de hipótese comparando D-MEANDS-II aos demais algoritmos em instâncias com quinze mudanças de ambiente (EC=15).	84
Tabela 17 – Teste de hipótese comparando D-MEANDS-II aos demais algoritmos em instâncias com vinte mudanças de ambiente (EC=20).	84
Tabela 18 – Teste de hipótese comparando o desempenho do D-MEANDS-III em relação aos seus antecessores, considerando instâncias com EC=10. . .	87
Tabela 19 – Teste de hipótese comparando o desempenho do D-MEANDS-III em relação aos seus antecessores, considerando instâncias com EC=15. . .	88
Tabela 20 – Teste de hipótese comparando o desempenho do D-MEANDS-III em relação aos seus antecessores, considerando instâncias com EC=20. . .	88
Tabela 21 – Comparação dos algoritmos através da métrica CS em instâncias de 30 itens e EC=20.	92
Tabela 22 – Comparação dos algoritmos através da métrica CS em instâncias de 50 itens e EC=20.	92
Tabela 23 – Comparação dos algoritmos através da métrica CS em instâncias de 100 itens e EC=20.	93

Lista de Siglas

$\overline{IGD}$	<i>Mean Inverted Generational Distance</i> (Distância Geracional Invertida Média)
AD	<i>Density</i> (Densidade)
AG	Algoritmo Genético
CS	<i>Coverage of two Sets</i> (Cobertura de Dois Conjuntos)
D-MEANDS	<i>Dynamic Many-objective Evolutionary Algorithm Based on Nondominated Decomposed Sets</i> (Algoritmo Evolutivo Dinâmico de Muitos Objetivos Baseado em Conjuntos Decompostos Não Dominados)
D-MEANDS-II	<i>Dynamic Many-objective Evolutionary Algorithm Based on Nondominated Decomposed Sets II</i> (Algoritmo Evolutivo Dinâmico de Muitos Objetivos Baseado em Conjuntos Decompostos Não Dominados II)
D-MEANDS-III	<i>Dynamic Many-objective Evolutionary Algorithm Based on Nondominated Decomposed Sets III</i> (Algoritmo Evolutivo Dinâmico de Muitos Objetivos Baseado em Conjuntos Decompostos Não Dominados III)
D-MEANDS-MD	<i>Dynamic Many-objective Evolutionary Algorithm Based on Nondominated Decomposed Sets with Memory and Diversity</i> (Algoritmo Evolutivo Dinâmico de Muitos Objetivos Baseado em Conjuntos Decompostos Não Dominados com Memória e Diversidade)
DDIS-MOEA/D-DE	<i>MOEA/D based on Differential Evolution and Dynamic Diversity Introduction Strategy</i> (MOEA/D baseado em Evolução Diferencial e Estratégia de Introdução de Diversidade Dinâmica)
DMaOPs	<i>Many-objectives Dynamic Optimization Problems</i> (Problemas de Otimização Dinâmico de Muitos Objetivos)
DMKP	<i>Dynamic Multiobjective Knapsack Problem</i> (Problema da Mochila Multiobjetivo Dinâmico)
DMOEA	<i>Dynamic Multi-objective Optimization EA</i> (Algoritmo Evolutivo de Otimização Multiobjetiva Dinâmica)
DMOPs	<i>Dynamic Multiobjective Problems</i> (Problemas Multiobjetivo Dinâmicos)
DNSGA-II	<i>Dynamic Non-dominated Sorting Genetic Algorithm II</i> (Algoritmo Genético de Ordenação Não Dominada Dinâmica II)
DOPs	<i>Dynamic Optimization Problems</i> (Problemas de Otimização Dinâmica)
DSOPs	<i>Dynamic Single Objective Optimization Problems</i> (Problemas de Otimização Dinâmica de Objetivo único)
EAs	<i>Evolutionary Algorithms</i> (Algoritmos Evolutivos)
EC	<i>Environment Changes</i> (Mudanças de Ambiente)
EMO	<i>Evolutionary Multiobjective Optimization</i> (Otimização Multiobjetiva Evolutiva)
EstMOP	<i>Static Multi-Objective Optimization Problem</i> (Problema de Otimização Multi-Objetivo Estática)
HV	<i>Hypervolume</i> (Hypervolume)
HVR	<i>Hypervolume Ratio</i> (Razão do Hypervolume)
MaOPs	<i>Many-objective Optimization Problems</i> (Problemas de Otimização de Muitos Objetivos)
MEANDS	<i>Many-objective Evolutionary Algorithm Method Based on Non-dominated Decomposed Sets</i> (Método de Algoritmo Evolutivo de Muitos Objetivos Baseado em Conjuntos Decompostos)

	Não Dominados)
MEANDS-II	<i>Many-objective Evolutionary Algorithm based on Non-dominated Decomposed Sets II</i> (Método de Algoritmo Evolutivo de Muitos Objetivos Baseado em Conjuntos Decompostos Não Dominados II)
MKP	<i>Multiobjective Knapsack Problem</i> (Problema da Mochila Multiobjetivo)
MOEA/D-KF	<i>Multiobjective Evolutionary Algorithm with Decomposition with Kalman Filter</i> (Algoritmo Evolutivo Multiobjetivo com Decomposição e Filtro de Kalman)
MOEAs	<i>Multiobjective Evolutionary Algorithms</i> (Algoritmos Evolutivos Multiobjetivo)
MOPs	<i>Multiobjective Optimization Problems</i> (Problemas de Otimização Multiobjetivo)
MS-MOEA	<i>Multi-strategy Ensemble MOEA</i> (MOEA em Ensemble com Múltiplas Estratégias)
NSGA-III	<i>Non-dominated Sorting Genetic Algorithm III</i> (Algoritmo Genético de Ordenação Não Dominada III)
SMaOPs	<i>Static Many-objective Optimization Problems</i> (Problemas de Otimização Estáticos de Muitos Objetivos)

Sumário

1	INTRODUÇÃO	29
1.1	Contextualização e Motivação	29
1.2	Hipótese e Objetivos	31
1.3	Organização da Tese	33
2	FUNDAMENTAÇÃO TEÓRICA	35
2.1	Otimização Multiobjetivo	35
2.2	Otimização Multiobjetivo Dinâmica	37
2.3	Algoritmos Evolutivos para DMOPs	38
2.3.1	DNSGA-II	40
2.3.2	MS-MOEA	41
2.3.3	DDIS-MOEA/D-DE	41
2.3.4	MOEA/D-KF	42
2.4	Métricas para Avaliação de Desempenho de MOEAs em DMOPs	43
2.5	Problema da Mochila Multiobjetivo Dinâmico	45
2.6	Considerações Finais	46
3	MOEAS PROPOSTOS	47
3.1	DNSGA-II*	47
3.2	DNSGA-III	48
3.3	D-MEANDS	49
3.4	D-MEANDS-MD	52
3.5	D-MEANDS-II	55
3.6	D-MEANDS-III	57
4	EXPERIMENTOS E ANÁLISE DOS RESULTADOS	63
4.1	Metodologia dos Experimentos	64

4.2	Fase 1: Análise Comparativa de Desempenho de Métodos Evolutivos da Literatura	65
4.2.1	Comparando Métodos da Literatura com EC=1	65
4.2.2	Comparando Métodos da Literatura com EC=2	67
4.2.3	Análise de Desempenho do DNSGA-II*	69
4.3	Fase 2: Análise de Desempenho do Algoritmo D-MEANDS . .	71
4.3.1	Cenários com uma Única Mudança de Ambiente	71
4.3.2	Cenários com Duas Mudanças de Ambiente	72
4.3.3	Cenários com Cinco e Dez Mudanças de Ambiente	73
4.3.4	Cenários com Muitas Mudanças de Ambiente	74
4.4	Fase 3: Análise de Desempenho do Algoritmo D-MEANDS-MD	79
4.5	Fase 4: Análise de Desempenho do Algoritmo D-MEANDS-II .	83
4.6	Fase 5: Análise de Desempenho do Algoritmo D-MEANDS-III	87
4.7	Análises Complementares	90
4.7.1	Comparando os Conjuntos de Soluções retornados pelos MOEAs	90
4.7.2	Aprofundando a Análise do Hipervolume	93
4.7.3	Comparando o Tempo de Execução do D-MEANDS-MD com D-MEANDS-II	95
4.8	Síntese das Análises	99
5	CONCLUSÃO	105
5.1	Trabalhos Futuros	106
5.2	Contribuições em Produção Bibliográfica	107
REFERÊNCIAS		109

APÊNDICES 117

APÊNDICE A	– EXPERIMENTOS PRELIMINARES COM OS ALGORITMOS D-MEANDS-II E D-MEANDS-III	119
A.1	Estudo da Quantidade de Subpopulações do D-MEANDS-II . .	119
A.2	Tour4Ind: Uma Nova Dinâmica para o Cruzamento	123
A.3	Filtro-PR: Novo Método de Mudança de Ambiente	127
APÊNDICE B	– DADOS COMPLEMENTARES DOS EXPERIMENTOS	131
B.1	Fase 1: Análise de Desempenho de Métodos Evolutivos da Literatura	131

B.1.1	Análise Comparativa dos Métodos da Literatura em Cenários com uma Mudança de Ambiente	131
B.1.2	Análise Comparativa de Métodos da Literatura em Cenários com duas Mudanças de Ambiente	133
B.2	Fase 2: Análise de Desempenho do Algoritmo D-MEANDS . .	135
B.2.1	Comparação dos Algoritmos Considerando um Única Mudança de Ambiente	135
B.2.2	Comparação dos Algoritmos Considerando Duas Mudança de Ambiente	136
B.2.3	Comparação dos Algoritmos Considerando Cinco e Dez Mudanças de Ambiente	137
B.2.4	Comparação entre os Algoritmos Considerando Muitas Mudanças de Ambiente	138
B.3	Fase 3: Análise de Desempenho do Algoritmo D-MEANDS-MD	142
B.4	Fase 4: Análise de Desempenho do Algoritmo D-MEANDS-II .	146
B.5	Fase 5: Análise de Desempenho do Algoritmo D-MEANDS-III	150

Introdução

1.1 Contextualização e Motivação

Algoritmos Evolutivos (EAs) são métodos de busca e otimização baseados nos conceitos de seleção natural e evolução (GOLDBERG; KORB; DEB, 1989). Um dos pontos fortes de um EA é sua robustez, sendo capaz de fornecer soluções de alta qualidade para problemas considerados intratáveis por métodos tradicionais de otimização. Os EAs, por meio dos mecanismos de seleção, cruzamento e mutação, alteram dinamicamente o processo de busca para encontrar melhores soluções. Além disso, conseguem avaliar vários indivíduos, produzir múltiplas soluções e apresentam maior capacidade de busca global (KATOCH; CHAUHAN; KUMAR, 2021).

Vários EAs foram propostos ao longo dos anos, sendo que alguns deles são voltados à resolução de problemas de otimização multiobjetivo (MOPs), os quais são chamados de algoritmos evolutivos multiobjetivo (MOEAs). Problemas dessa classe tendem a ser bastante desafiadores, pois o principal objetivo é encontrar soluções que otimizem vários objetivos ao mesmo tempo. Diferentemente dos problemas de otimização de objetivo único, em MOPs não existe uma única solução ótima para todos os objetivos. Uma característica dos MOPs é que os objetivos, em geral, não são correlacionados, de modo que suas otimizações são conflitantes. Por essa razão, pode haver mais de uma solução considerada ótima, e os MOEAs buscam encontrar um conjunto dessas soluções, definido como conjunto Pareto Ótimo (FONSECA; FLEMING, 1993).

Muitas aplicações do mundo real envolvem quatro ou mais objetivos, sendo comumente classificadas como problemas de otimização de muitos objetivos (*many-objective optimization problems*), ou, simplesmente, MaOPs. Em problemas desse tipo, a proporção de soluções que se tornam não dominadas aumenta rapidamente com o número de objetivos (KARAMI; DARIANE, 2022). Isso resulta em uma diminuição significativa da pressão seletiva durante o processo evolutivo (KARAMI; DARIANE, 2022).

Existem, ainda, problemas de otimização que apresentam uma característica dinâmica. Em uma otimização dinâmica, pelo menos um dos componentes fundamentais da formula-

ção do problema — como funções-objetivo, restrições, parâmetros ou o espaço de busca — varia ao longo do tempo, provocando mudanças na solução ótima à medida que o estado do sistema evolui. Os primeiros trabalhos em otimização dinâmica concentravam-se em problemas de um único objetivo (ECKSTEIN; WOLPIN, 1989), havendo pouquíssimos estudos relacionados a problemas multiobjetivo (JIN; BRANKE, 2005; RAQUEL; YAO, 2013). Entretanto, trabalhos recentes em otimização dinâmica tendem a explorar problemas de muitos objetivos, sendo esse um dos desafios da área de computação evolutiva (AZZOUZ; BECHIKH; SAID, 2017). Problemas que lidam com a otimização dinâmica de muitos objetivos são conhecidos como DMaOPs (*Dynamic Many-objective Optimization Problems*) e tendem a estar entre os mais próximos dos problemas encontrados no mundo real (FOGEL, 1998).

Em Problemas de Otimização Multiobjetivo Estática (SMOPs), o objetivo é encontrar um conjunto de melhores soluções para o problema (MURATA; ISHIBUCHI et al., 1995). Já os algoritmos que lidam com DMOPs precisam adaptar seu conjunto de melhores soluções de acordo com as mudanças do ambiente ao longo do tempo. Em outras palavras, algoritmos que resolvem DMOPs devem ter a capacidade de acompanhar a evolução do conjunto Pareto Ótimo, a fim de encontrar as mesmas soluções não dominadas ou soluções próximas a elas (HELBIG et al., 2012). Em razão das mudanças no ambiente, os algoritmos desenvolvidos originalmente para lidar com SMOPs não são eficazes em DMOPs (JIANG et al., 2018). Esse fato motivou o surgimento de alguns algoritmos evolutivos propostos especificamente para esse conjunto de problemas, como o *Dynamic Non-dominated Sorting Genetic Algorithm II* (DNSGA-II) (KALYANMOY; BHASKARA; SINDHYA, 2007), o *Multi-strategy Ensemble MOEA* (MS-MOEA) (WANG et al., 2010) e o *Dynamic Multiobjective Optimization EA* (DMOEA) (ZHENG, 2007), entre outros (MURUGANANTHAM; TAN; VADAKKEPAT, 2016; LIU et al., 2020). Esses algoritmos evolutivos são inspirados em versões anteriores aplicadas a SMOPs, porém incorporam estratégias que se adaptam melhor aos DMOPs. Portanto, um tópico de pesquisa relevante consiste na adaptação de algoritmos recentes, com abordagens diversificadas e originalmente aplicados a SMOPs, com foco na adoção de estratégias capazes de aumentar seu desempenho em problemas dinâmicos.

O uso de algoritmos evolutivos para lidar com DMOPs tem despertado o interesse de diversos pesquisadores (AZZOUZ; BECHIKH; SAID, 2017). No entanto, a maioria dos problemas tratados corresponde a problemas de otimização contínua. Por esse motivo, o foco desta tese está em problemas de otimização discreta, ainda pouco explorados em abordagens para DMOPs. Outra característica importante a ser investigada está na formulação dinâmica de problemas com muitos objetivos, uma vez que é mais comum encontrar, na literatura, trabalhos sobre DMOPs que envolvem três ou menos objetivos (JIANG; ZOU; YAO, 2022).

1.2 Hipótese e Objetivos

Como já citado anteriormente, muitos problemas do mundo real tendem a ser discretos, com muitos objetivos e dinâmicos (AZZOUZ; BECHIKH; SAID, 2017). Contudo, a otimização dinâmica de problemas com quatro ou mais objetivos é difícil de resolver, e estratégias evolutivas convencionais podem não apresentar bons resultados. Portanto, faz-se necessária uma estratégia capaz de lidar com mudanças de ambiente no contexto dinâmico, e o algoritmo deve dispor de um sistema de busca voltado para problemas com muitos objetivos.

Vários EAs aplicados em SMaOPs já se mostraram mais eficazes em problemas discretos do que em problemas contínuos. O método denominado *Many-objective Evolutionary Algorithm Method Based on Non-dominated Decomposed Sets* (MEANDS) (LAFETA et al., 2018), proposto em 2018, apresentou bons resultados em problemas discretos com muitos objetivos. Posteriormente, uma extensão desse algoritmo foi proposta para lidar com instâncias mais complexas, demonstrando melhor convergência e diversidade em relação ao MEANDS original. Essa nova versão foi denominada *Many-objective Evolutionary Algorithm based on Non-dominated Decomposed Sets II* (MEANDS-II) (LAFETA; OLIVEIRA, 2019).

Nesse contexto, a hipótese principal desta pesquisa é que esses algoritmos evolutivos (MEANDS e MEANDS-II) podem ser adaptados com sucesso para lidar com problemas de muitos objetivos que também apresentem características dinâmicas.

Portanto, o objetivo principal deste trabalho é investigar estratégias evolutivas capazes de melhorar a convergência e a diversidade das soluções durante uma otimização dinâmica com muitos objetivos e, ao mesmo tempo, reduzir o impacto do aumento no número de objetivos sobre o custo computacional do algoritmo. Para isso, pretende-se investigar diferentes conceitos voltados a problemas dinâmicos definidos na literatura e incorporá-los aos algoritmos analisados. Para alcançar essa meta, faz-se necessário atingir os seguintes objetivos específicos:

- ❑ **Investigar e avaliar o desempenho de diferentes algoritmos evolutivos** propostos na literatura para problemas de otimização multiobjetivo dinâmicos (DMOPs), considerando instâncias com quatro ou mais objetivos.
- ❑ **Desenvolver novos algoritmos evolutivos (MOEAs)** que combinem técnicas já conhecidas com mecanismos inovadores, visando melhorar a convergência, a diversidade e o tempo de execução em problemas dinâmicos com muitos objetivos.
- ❑ **Propor adaptações de algoritmos existentes (como MEANDS e MEANDS-II)** para que sejam capazes de lidar com cenários dinâmicos, explorando estratégias evolutivas adequadas a esse contexto.

- **Comparar os algoritmos propostos com abordagens consolidadas na literatura**, utilizando o *Dynamic Multiobjective Knapsack Problem* 0-1 para validar a eficácia das soluções desenvolvidas.

Como descrito, um dos objetivos deste trabalho é avaliar o desempenho de diferentes algoritmos evolutivos propostos na literatura para DMaOPs. A partir dessa análise, busca-se propor novos MOEAs que sejam eficientes para lidar com problemas dinâmicos que envolvam muitos objetivos (mais de três objetivos).

O primeiro passo nesse estudo é implementar algoritmos conhecidos na literatura que apresentam bom desempenho na classe de problemas DMOPs (KALYANMOY; BHASKARA; SINDHYA, 2007; MURUGANANTHAM; TAN; VADAKKEPAT, 2016; WANG et al., 2010; LIU et al., 2020). Esses algoritmos empregam diferentes estratégias no processo de busca para lidar com as mudanças que ocorrem nos objetivos ao longo das iterações. Por isso, decidiu-se implementar pelo menos um algoritmo que utilize as seguintes estratégias: introdução de diversidade (KALYANMOY; BHASKARA; SINDHYA, 2007), abordagem baseada em previsão de alterações (KOO; GOH; TAN, 2010) e abordagem baseada em memória (GOH; TAN, 2008). A estratégia conhecida como “abordagens paralelas” (CAMARA; ORTEGA; TORO, 2007) não será abordada neste trabalho devido ao seu alto custo computacional, sendo deixada para pesquisas futuras.

Inicialmente, foram implementados três algoritmos evolutivos encontrados na literatura: *Dynamic Non-dominated Sorting Genetic Algorithm II* (DNSGA-II) (KALYANMOY; BHASKARA; SINDHYA, 2007), *Multi-strategy Ensemble MOEA* (MS-MOEA) (WANG et al., 2010) e *Multiobjective Evolutionary Algorithm with Decomposition with Kalman Filter* (MOEA/D-KF) (MURUGANANTHAM; TAN; VADAKKEPAT, 2016). Cada algoritmo possui uma estratégia evolutiva distinta voltada a DMOPs. Como estudo de caso para as investigações, adotou-se uma versão dinâmica do problema da mochila, considerando instâncias com 4, 6 e 8 objetivos (*Dynamic Multiobjective Knapsack Problem* 0-1) (KELLERER; PFERSCHY; PISINGER, 2004). Além disso, é proposta uma versão do algoritmo *Non-dominated Sorting Genetic Algorithm III* (NSGA-III) (LI et al., 2014), adaptada para lidar com DMaOPs, denominada DNSGA-III. Também é desenvolvida uma variação do DNSGA-II, que incorpora o uso da estratégia evolutiva de arquivo externo, a exemplo da estratégia utilizada pelo MS-MOEA. Essa extensão é chamada de DNSGA-II*. Os resultados dessa investigação resultaram na primeira publicação deste trabalho (LAFETÁ; OLIVEIRA, 2020A).

Em uma segunda etapa da investigação, propõe-se uma adaptação do algoritmo MEANDS (LAFETA et al., 2018), na qual é empregada uma estratégia evolutiva baseada em memória para resolver DMOPs. Essa nova versão do algoritmo é chamada de D-MEANDS. Realizou-se uma análise de desempenho comparando seus resultados com os dois algoritmos da literatura investigados anteriormente: MS-MOEA e DNSGA-II. Assim como nas investigações anteriores, os algoritmos também foram aplicados ao *Dynamic*

Multiobjective Knapsack Problem 0-1. Os resultados dessa etapa foram publicados no artigo (LAFETÁ; OLIVEIRA, 2020B).

Em uma terceira etapa desta pesquisa, é proposta uma nova versão do D-MEANDS, utilizando uma estratégia para inserir diversidade durante a busca. Essa nova versão é chamada de D-MEANDS-MD e apresentou resultados superiores aos do D-MEANDS, com tempo de execução similar. Nessa etapa, também é incorporado um MOEA da literatura conhecido como *MOEA/D based on Differential Evolution and Dynamic Diversity Introduction Strategy* (DDIS-MOEA/D-DE) (LIU et al., 2020), que é um algoritmo para problemas dinâmicos multiobjetivo e utiliza o mesmo processo evolutivo do MOEA/D-DE (LI; ZHANG, 2008), propondo uma estratégia dinâmica adaptativa de acordo com a gravidade da mudança. Os resultados alcançados nessas investigações foram publicados em um novo artigo (LAFETÁ; MARTINS; OLIVEIRA, 2024).

Ainda em relação a problemas discretos, o MEANDS-II (LAFETA; OLIVEIRA, 2019) é um algoritmo que se mostrou mais eficaz que seu predecessor, o MEANDS, em problemas discretos com conjuntos Pareto Ótimo muito grandes (mais de 1000 soluções). Por isso, neste trabalho é proposta uma adaptação do MEANDS-II, que incorpora uma abordagem para diminuir a quantidade de subconjuntos utilizados na exploração do algoritmo, bem como a mesma estratégia dinâmica baseada em memória empregada no D-MEANDS. Esse novo algoritmo é chamado de D-MEANDS-II e tem como principal propósito competir com os algoritmos D-MEANDS e D-MEANDS-MD em termos de desempenho multiobjetivo, mas com tempo de execução significativamente menor.

Por fim, são propostos dois mecanismos evolutivos visando explorar o conceito multipopulacional apresentado no D-MEANDS-II. O primeiro consiste em um novo sistema de seleção de pais para *crossover*, mais simples de implementar. O segundo mecanismo propõe um novo algoritmo de mudança de ambiente baseado em memória. Esse novo algoritmo explora as informações contidas em cada subpopulação do ambiente anterior para formar a nova população do novo ambiente. A partir da integração desses dois mecanismos com a estratégia evolutiva do D-MEANDS-II, é proposto o algoritmo denominado D-MEANDS-III.

1.3 Organização da Tese

Esta tese está estruturada em cinco capítulos, nos quais são apresentados os conceitos teóricos, a metodologia e os resultados da pesquisa. O Capítulo 1 apresenta a motivação, os objetivos e a organização da tese. O Capítulo 2 descreve os fundamentos teóricos relacionados aos problemas de otimização abordados. O Capítulo 3 apresenta os algoritmos propostos. O Capítulo 4 descreve os experimentos realizados e seus resultados. Por fim, o Capítulo 5 apresenta as conclusões, as principais contribuições, os trabalhos futuros e a produção bibliográfica.

Fundamentação Teórica

Neste capítulo, são apresentados os conceitos de otimização multiobjetivo, com foco principal em problemas de otimização de muitos objetivos e em problemas dinâmicos. Também são apresentadas as formulações matemáticas dessas classes de problemas, com o objetivo de facilitar sua compreensão. Além disso, são descritas as métricas para problemas de otimização dinâmica aplicadas nas investigações deste trabalho. Em seguida, são definidos os conceitos relacionados ao *Dynamic Multiobjective Knapsack Problem* 0-1 e, por fim, são apresentados trabalhos da literatura que investigaram problemas de otimização dinâmica.

2.1 Otimização Multiobjetivo

A maioria dos problemas do mundo real na área de otimização tem como desafio atingir diversos objetivos simultaneamente (IGLESIAS; GÁLVEZ, 2023). É comum que esses objetivos sejam conflitantes, ou seja, a melhoria de um deles pode prejudicar os demais. Assim, pode não ser possível obter uma única solução que contenha os melhores valores para todos os objetivos. Nesse contexto, pode ser mais interessante encontrar um conjunto das melhores soluções para o problema, e não apenas uma única solução (FONSECA; FLEMING, 1993).

Esses problemas são conhecidos como Problemas de Otimização Multiobjetivo (MOPs). Problemas dessa natureza envolvem a minimização (ou maximização) simultânea de um conjunto de objetivos, sujeitos a determinadas restrições. Mesmo diante da existência de vários objetivos conflitantes, algumas soluções conseguem superar outras em todos os objetivos. Nesses casos, diz-se que essas soluções dominam as demais. Em um MOP, as melhores soluções não são dominadas por nenhuma outra solução do espaço de busca e são definidas como soluções não dominadas (ARROYO et al., 2002).

Diferentemente de problemas de objetivo único, os MOPs geralmente possuem mais de uma solução ótima a ser encontrada, pois seus objetivos podem estar em conflito; isto é, ao melhorar um objetivo, a busca pode prejudicar outro (MANNION et al., 2018). No en-

tanto, ainda é possível definir se uma solução é melhor que outra no espaço de busca de um MOP considerando o conceito de dominância de Pareto (FONSECA; FLEMING, 1993). Dado o conjunto de possíveis soluções para o problema investigado, pode-se determinar que uma solução x domina outra solução y ($x \prec y$, em um problema de minimização) se x for melhor que y em pelo menos um objetivo e não for superada nos demais. Quando uma solução não é dominada por nenhuma outra em todo o espaço de busca, ela pertence ao conjunto das melhores soluções do MOP, denominado conjunto Pareto Ótimo (P^*). Portanto, o objetivo de um algoritmo de otimização multiobjetivo é encontrar P^* ou, pelo menos, um conjunto de soluções próximas a ele, denominado conjunto Pareto aproximado.

Considere um problema π com m objetivos e Ω como o conjunto de todas as soluções factíveis de π (espaço de busca). Um MOP que considere a minimização de todos os objetivos de π pode ser formulado matematicamente como:

$$\min f(x) = \{f_1(x), f_2(x), \dots, f_m(x)\}, \text{ onde } g(x) > 0, h(x) = 0 \quad (1)$$

onde x é uma solução do espaço de busca ($x \in \Omega$); $f(x)$ é o conjunto das funções objetivo consideradas, sendo $f_i(x)$ o valor do i -ésimo objetivo para x ($i = 1, \dots, m$); e as funções $g(x)$ e $h(x)$ representam respectivamente o conjunto de restrições de desigualdade e igualdade do problema. Nesse contexto, dizemos que x domina outra solução do espaço de busca Ω ($x \prec y$) se e somente se:

$$\begin{aligned} f_i(x) &< f_i(y), \text{ para algum } i \in \{1, \dots, m\}; \text{ e} \\ f_j(x) &\leq f_j(y), \text{ para todo } j \in \{1, \dots, m\} \text{ e } j \neq i \end{aligned} \quad (2)$$

À medida que aumenta o número de objetivos a serem otimizados, o espaço de busca cresce exponencialmente e, conseqüentemente, aumenta também a complexidade de se encontrar o conjunto Pareto Ótimo (ISHIBUCHI; AKEDO; NOJIMA, 2013). Métodos de otimização convencionais, como aqueles baseados no gradiente descendente (STIEFEL, 1952) e no método Simplex (NELDER; MEAD, 1965), quando adaptados a problemas multiobjetivo, apresentam limitações e tornam-se ineficientes. Desse modo, métodos de otimização inexatos têm sido empregados em problemas com múltiplos objetivos e, dentre eles, os algoritmos evolutivos multiobjetivo (MOEAs) têm se destacado (PEREIRA, 2004).

Embora a maioria dos algoritmos de otimização inexatos desenvolvidos para lidar com MOPs apresente bom desempenho em problemas com dois ou três objetivos (DEB et al., 2002; ZITZLER; LAUMANN; THIELE, 2001), eles tendem a reduzir sua capacidade de busca a partir de quatro objetivos. Embora existam outros fatores que também influenciem a capacidade de exploração do algoritmo, como o formato da fronteira de Pareto e as restrições do problema (PANICHELLA, 2022), o aumento no número de objetivos torna mais difícil a busca pelo conjunto Pareto Ótimo (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008). No caso dos MOEAs, esse aumento faz com que a população contenha um

grande número de soluções não dominadas, reduzindo a pressão seletiva (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008). Nesse contexto, problemas com mais de três objetivos são categorizados como problemas de otimização de muitos objetivos (MaOPs) e demandam algoritmos adaptados para lidar, de forma mais eficiente, com o aumento no número de objetivos e suas consequências.

2.2 Otimização Multiobjetivo Dinâmica

Nos últimos anos, contribuições significativas têm sido propostas para a solução de problemas de otimização multiobjetivo de natureza estática (SMOPs). No entanto, existem muitos problemas de otimização cujos objetivos, restrições e parâmetros podem mudar ao longo do tempo. Esses problemas são chamados de problemas de otimização multiobjetivo dinâmicos (DMOPs). Um DMOP pode ser formalmente definido como:

$$\min f(x, t) = \{f_1(x, t), \dots, f_m(x, t)\}, \text{ onde } x \in \Omega, g(x, t) > 0, h(x, t) = 0 \quad (3)$$

onde, t representa o tempo ou a natureza dinâmica do problema. Ou seja, a formulação matemática apresentada na Equação (1) é adaptada para que suas funções considerem mudanças no ambiente no tempo t .

Embora os problemas de otimização multiobjetivo dinâmicos mantenham o mesmo conceito de dominância utilizado em sua versão estática, eles se caracterizam por lidar com mudanças no ambiente ao longo do tempo, afetando as funções objetivo e as restrições do problema (AZZOUZ; BECHIKH; SAID, 2017). Essas mudanças impactam o espaço objetivo, modificando o conjunto Pareto Ótimo. Isso faz com que soluções antes válidas e não dominadas possam se tornar inviáveis ou dominadas após a mudança (HELBIG et al., 2012). Por exemplo, em um problema de planejamento de rotas, determinadas vias podem se tornar indisponíveis em decorrência de acidentes, obras ou congestionamentos. Isso altera as restrições de viabilidade, exigindo alterações nas rotas planejadas a fim de considerar as novas condições de tráfego. Em sistemas de redes sem fio, o nível de interferência do sinal pode variar dinamicamente, afetando o conjunto de soluções (configuração do sistema) que visam maximizar a taxa de transmissão e minimizar o consumo de energia. No escalonamento de uma linha de produção, uma máquina pode apresentar falhas ou perder eficiência, alterando a capacidade de processamento e exigindo uma readequação da ordem de produção para atender aos prazos e reduzir custos. Portanto, algoritmos evolutivos multiobjetivo utilizados em otimização dinâmica devem apresentar rápida convergência para uma boa aproximação do conjunto Pareto Ótimo sempre que o ambiente muda (BRANKE, 2012).

Existem duas características que definem o grau de dificuldade de um problema dinâmico: frequência e severidade da mudança (RICHTER, 2013). A mudança de ambiente

pode ocorrer em diferentes momentos e, quanto mais rápida ela for, menor será o tempo disponível para o algoritmo efetuar a busca entre as diversas configurações do problema. Nesse caso, diz-se que a frequência de mudança é alta. Quanto menor a frequência de mudança, maior será o tempo disponível para o algoritmo encontrar as melhores soluções para aquela configuração do problema (ambiente). Quando o ambiente muda, o espaço de busca é afetado. Nesse sentido, a severidade da mudança está relacionada ao tamanho de seu impacto (perturbação no espaço). Quanto mais severa for a mudança, maior será esse impacto. Uma severidade muito alta faz com que o espaço de busca se altere a ponto de modificar todo o conjunto Pareto Ótimo. Nesse caso, pode ser mais adequado reiniciar a busca do zero do que empregar alguma estratégia dinâmica.

2.3 Algoritmos Evolutivos para DMOPs

Os Algoritmos Evolutivos (EAs) foram propostos com o objetivo de estudar os fenômenos relacionados à adaptação das espécies e à seleção natural, bem como de desenvolver uma maneira de incorporar esses conceitos à computação (HOLLAND, 1975). Os primeiros estudos foram voltados à solução de problemas mono-objetivo, sujeitos a uma única função de avaliação. Por algumas décadas, esse tipo de problema de otimização constituiu o foco das pesquisas na literatura. Apesar de serem métodos apropriados para problemas não lineares, com uma vasta gama de aplicabilidade, com o tempo surgiu a necessidade de aperfeiçoar o modelo básico de EAs para melhorar seu desempenho em determinadas classes de problemas que desafiavam a busca evolutiva.

Desde os anos 1980, diferentes estudos investigavam estratégias evolutivas voltadas à manutenção da diversidade na população, buscando corrigir a chamada “convergência prematura” nos EAs (JONG; SPEARS et al., 1989; ESHELMAN, 1991; GOLDBERG; KORB; DEB, 1989; GOLDBERG, 1990; TSUTSUI; FUJIMOTO et al., 1993; TSUTSUI; FUJIMOTO; HAYASHI, 1994). Na década seguinte, diversas pesquisas propuseram EAs baseados em hibridização (COSTA, 1995; GLOVER, 1994; GLOVER; KELLY; LAGUNA, 1995; KITANO, 1990; MALEK et al., 1989a; MALEK et al., 1989b; MÜHLENBEIN, 1992; MÜHLENBEIN; GORGES-SCHLEUTER; KRÄMER, 1988; POWELL; TONG; SKOLNICK, 1989; ULDER et al., 1990). Nessa abordagem híbrida, algoritmos evolutivos foram utilizados em conjunto com paradigmas como busca tabu, redes neurais artificiais, *simulated annealing*, entre outros. Ainda nessa década, pesquisadores passaram a investigar o potencial dos EAs em problemas de otimização multiobjetivo (MOPs), que envolvem o desafio de otimizar mais de um objetivo simultaneamente.

Uma forte motivação para essa linha de investigação reside no fato de que problemas do mundo real tendem a envolver múltiplos objetivos (ZBIGNIEW, 1996). Dessa forma, pesquisadores passaram a investigar as características desses problemas, desenvolvendo novas estratégias evolutivas capazes de manipular mais de uma função de avaliação simultanea-

mente (CANTÚ-PAZ, 1999; COELLO, 1999; SCHAFFER, 1985; SRINIVAS; DEB, 1994; TAMAKI; KITA; KOBAYASHI, 1996; FONSECA; FLEMING et al., 1993; HIROYASU; MIKI; WATANABE, 1999; HORN; NAFPLIOTIS; GOLDBERG, 1993). Inicialmente, os problemas investigados envolviam dois ou três objetivos conflitantes. Além disso, esses problemas apresentam mais de uma solução ótima no espaço de busca, o que levou à necessidade de lidar com o conceito de conjunto Pareto Ótimo, que representa o conjunto das melhores soluções não dominadas no espaço de busca.

Após quase duas décadas de refinamento, a comunidade consolidou dois algoritmos evolutivos multiobjetivo (MOEAs): o NSGA-II (DEB et al., 2002) e o SPEA2 (ZITZLER; LAUMANN; THIELE, 2001). Entretanto, problemas multiobjetivo tendem a se tornar mais complexos à medida que o número de objetivos cresce (GAO; SHI; YAO, 2000). O aumento no número de objetivos altera características importantes no espaço de objetivos, refletindo diretamente no espaço de busca. Uma das consequências mais impactantes é a redução da pressão seletiva durante a busca (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008). Por essa razão, problemas com quatro ou mais objetivos são caracterizados como problemas de otimização de muitos objetivos (MaOPs) e desafiam os MOEAs consagrados, como o NSGA-II e o SPEA2 (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008). Problemas discretos e de muitos objetivos representam uma parcela significativa dos problemas de otimização do mundo real (GU et al., 2022).

Mais recentemente, uma nova classe de problemas de otimização vem sendo investigada pela comunidade da computação evolutiva, por apresentar características fortemente associadas a problemas do mundo real. Trata-se dos chamados problemas de otimização multiobjetivo dinâmicos (DMOPs), cujas características (ou ambiente) mudam com o passar do tempo (AZZOUZ; BECHIKH; SAID, 2017), conforme descrito na Seção 2.1.

Ao lidar com DMOPs, os EAs devem ser capazes de encontrar um conjunto de soluções de alta qualidade e diversificadas, além de acompanhar as mudanças do ambiente ao longo do tempo. Existem duas maneiras principais de reagir a essas mudanças. A primeira consiste em reiniciar o processo de busca do EA a cada mudança de ambiente. No entanto, essa solução é frequentemente impraticável, como ilustrado em (BRANKE, 2012). Outra abordagem consiste em utilizar o conhecimento obtido na busca anterior para acelerar a evolução do EA após a mudança no ambiente (JIN; BRANKE, 2005). Essa abordagem é a mais promissora, porém o EA deve garantir adaptabilidade, uma vez que a convergência durante o processo de busca pode provocar perda de diversidade (AZZOUZ; BECHIKH; SAID, 2017). Algumas estratégias já foram investigadas em DMOPs, sendo as mais comuns encontradas na literatura: (i) introdução de diversidade (KALYANMOY; BHASKARA; SINDHYA, 2007); (ii) estratégias baseadas em previsão de alterações (KOO; GOH; TAN, 2010); (iii) estratégias baseadas em memória (GOH; TAN, 2008); e (iv) abordagens paralelas (CAMARA; ORTEGA; TORO, 2007).

Algoritmos evolutivos são especialmente adequados para problemas de otimização mul-

tiobjetivo, pois lidam com uma população de pontos no espaço de busca, em vez de uma única solução. Um MOEA é caracterizado por uma população de candidatos a soluções, e o processo de reprodução permite a combinação de candidatos existentes para gerar novas soluções. Isso possibilita que a busca evolutiva encontre vários membros do conjunto Pareto Ótimo em uma única execução, em vez de realizar uma série de execuções separadas, como ocorre em alguns processos estocásticos convencionais (HORN; NAFPLIOTIS; GOLDBERG, 1993). Além disso, os MOEAs podem ser facilmente adaptados a diferentes problemas, independentemente da forma como seus objetivos e restrições são representados. Essa flexibilidade é ainda mais importante em problemas dinâmicos, uma vez que essas características — objetivos e restrições — podem mudar ao longo do tempo.

As estratégias para construir uma nova população a cada mudança de ambiente em um DMOP baseiam-se em dois princípios (HATZAKIS; WALLACE, 2006): (i) inserir diversidade para explorar regiões inacessíveis no novo ambiente; e (ii) utilizar informações do ambiente anterior para orientar a busca no novo ambiente. Essas duas características se adaptam bem aos algoritmos evolutivos, pois eles otimizam várias soluções simultaneamente, possibilitando a reutilização da informação de um ambiente para outro, ao mesmo tempo em que permitem a manutenção da diversidade por meio da introdução de pontos inexplorados no espaço de busca. A seguir, são apresentados os principais MOEAs aplicados a DMOPs.

2.3.1 DNSGA-II

O *Dynamic Nondominated Sorting Genetic Algorithm II* (DNSGA-II) (KALYANMOY; BHASKARA; SINDHYA, 2007) é uma adaptação do conhecido algoritmo NSGA-II, proposto por (DEB et al., 2002). Essa variação foi adaptada em (KALYANMOY; BHASKARA; SINDHYA, 2007) para problemas dinâmicos, utilizando uma estratégia de introdução de diversidade sempre que o ambiente empregado para avaliar as soluções sofre alguma perturbação. O comportamento evolutivo típico do NSGA-II foi mantido no DNSGA-II, exceto quando ocorre uma mudança de ambiente. O NSGA-II consiste em dividir a população em fronteiras hierárquicas, nas quais uma solução pertencente a uma fronteira superior apresenta melhor aptidão do que uma solução pertencente a uma fronteira inferior. As fronteiras são classificadas com base no conceito de dominância de Pareto, de modo que uma solução não dominada por nenhuma outra é classificada na fronteira mais alta. Essas fronteiras representam as soluções não dominadas da população atual, e o objetivo do algoritmo é aproximar essa fronteira superior do conjunto Pareto Ótimo do espaço de busca. As demais soluções são organizadas em uma hierarquia de fronteiras, na qual as soluções de uma fronteira superior dominam as soluções das fronteiras inferiores. A métrica *crowding distance* é utilizada para diferenciar as soluções pertencentes a uma mesma fronteira, atribuindo maior valor de *fitness* às soluções que se encontram mais isoladas nessa fronteira.

Na estrutura do DNSGA-II, uma porcentagem da população atual é modificada para promover a diversidade sempre que o ambiente muda. Duas estratégias para essa modificação foram avaliadas no DNSGA-II (KALYANMOY; BHASKARA; SINDHYA, 2007), definindo diferentes versões do algoritmo. A primeira versão foi denominada DNSGA-II-A, na qual uma porcentagem da população é mantida intacta para a próxima geração após a mudança do ambiente (uma forma de elitismo), enquanto novos indivíduos são gerados aleatoriamente para substituir as piores soluções. A segunda versão foi denominada DNSGA-II-B, na qual o elitismo também é utilizado para manter uma porcentagem das melhores soluções, enquanto os novos indivíduos que substituem as piores soluções são gerados por meio da aplicação de mutação sobre a elite.

2.3.2 MS-MOEA

O processo evolutivo do *Multi-strategy Ensemble MOEA* (MS-MOEA) (WANG et al., 2010) é baseado no conceito de não dominância, uma vez que sua população tende a manter soluções não dominadas e a descartar aquelas que são dominadas. Como a população possui tamanho fixo, é natural que, no início da busca, nem todos os indivíduos sejam não dominados. Novos descendentes não dominados podem ser gerados ao longo da busca e passam a substituir os indivíduos dominados da população. Por outro lado, quando a população atual não contém indivíduos dominados, a substituição é realizada de forma aleatória.

Como algumas soluções não dominadas podem ser removidas da população ao longo do processo evolutivo, o algoritmo armazena todas as soluções não dominadas encontradas durante a busca em um arquivo externo. Entretanto, como esse arquivo participa diretamente do processo evolutivo, fornecendo soluções para alguns cruzamentos, ele não é considerado um arquivo externo no sentido clássico (BOSMAN; THIERENS, 2003). A cada mudança de ambiente, algumas soluções armazenadas nesse arquivo sofrem mutação, e os novos indivíduos gerados são inseridos na população do ambiente seguinte. No entanto, apenas uma parte da população do novo ambiente é gerada a partir da mutação das soluções não dominadas encontradas no ambiente anterior. O restante da população é gerado aleatoriamente.

2.3.3 DDIS-MOEA/D-DE

O *MOEA/D based on Differential Evolution and Dynamic Diversity Introduction Strategy* (DDIS-MOEA/D-DE) (LIU et al., 2020) é um algoritmo para problemas multiobjetivo dinâmicos que utiliza o mesmo processo evolutivo do MOEA/D-DE (LI; ZHANG, 2008), propondo uma estratégia dinâmica adaptativa de acordo com a gravidade da mudança. O MOEA/D-DE é um algoritmo evolutivo multiobjetivo que decompõe o problema em vários subproblemas, os quais são evoluídos simultaneamente. Essa decomposição é

realizada por meio de uma função de escalarização, responsável por transformar cada indivíduo em um único valor escalar (*fitness*). A população atual é composta pelas melhores soluções para cada subproblema. A vizinhança de cada subproblema é definida com base nas distâncias entre seus vetores de coeficientes de ponderação.

É importante destacar que o MOEA/D-DE utiliza operadores de evolução diferencial (LI; ZHANG, 2008). O DDIS-MOEA/D-DE emprega duas estratégias para criar a população durante a mudança de ambiente. A primeira estratégia é utilizada quando a gravidade da mudança é alta: nesse caso, o algoritmo aproveita a população do ambiente anterior e substitui parte dela por indivíduos gerados aleatoriamente. Por outro lado, quando a gravidade da mudança é baixa, o algoritmo mantém a população do ambiente anterior e aplica mutação a uma parte dessa população. Nos experimentos, as mudanças de ambiente são controladas de modo a manter a gravidade da mudança o mais baixa possível; assim, a segunda estratégia é sempre utilizada. O número de indivíduos que sofrem mutação durante a mudança de ambiente corresponde a 20% da população, conforme adotado em (LIU et al., 2020).

2.3.4 MOEA/D-KF

O *Multi-Objective Evolutionary Algorithm based on Decomposition* (MOEA/D) (ZHANG; LI, 2007) é um algoritmo evolutivo multiobjetivo que resolve problemas de otimização por meio da decomposição de um problema com múltiplos objetivos em vários subproblemas escalarizados. Essa decomposição é realizada por uma função de escalarização, cuja função é transformar cada indivíduo em um único valor escalar (*fitness*). Assim, em vez de trabalhar diretamente com o conceito de dominância, o algoritmo associa cada indivíduo a um vetor de pesos, que define um subproblema escalar. Cada subproblema é otimizado simultaneamente e de forma colaborativa com seus vizinhos em um processo coevolutivo. A vizinhança de cada subproblema é definida com base nas distâncias entre seus vetores de coeficientes de ponderação. A técnica normalmente utilizada pelo algoritmo é a decomposição de Tchebycheff (MIETTINEN, 1999). Os operadores genéticos (cruzamento e mutação) atuam preferencialmente entre vizinhos para preservar a diversidade e explorar eficientemente o espaço de busca. O MOEA/D foi originalmente proposto para problemas multiobjetivo com até três objetivos, mas mostrou ser um algoritmo bem-sucedido também para formulações com muitos objetivos (FRANÇA; MARTINS; OLIVEIRA, 2018; MARTINS; FRANÇA; OLIVEIRA, 2019).

Muruganantham, Tan e Vadakkepat (2016) propuseram uma adaptação do MOEA/D para problemas dinâmicos, denominada MOEA/D-KF. Essa adaptação mantém a dinâmica evolutiva do algoritmo original, mas modifica a população quando ocorre uma mudança de ambiente. Nesse processo, sempre que o ambiente muda, parte da população é gerada aleatoriamente, enquanto a construção dos demais indivíduos é conduzida por um Filtro de Kalman (KALMAN; BUCY, 1961). Esse filtro não prediz indivíduos dire-

tamente, mas sim parâmetros do problema de decomposição, como os valores ótimos dos subproblemas escalares ou os centroides das soluções associadas aos vetores de peso.

2.4 Métricas para Avaliação de Desempenho de MOEAs em DMOPs

Quando lidamos com problemas estáticos, geralmente é suficiente avaliar apenas a população para a qual o algoritmo converge ao final da busca, com base em métricas de convergência e diversidade (COELLO, 1999). No entanto, em um contexto dinâmico, as métricas de desempenho devem avaliar não apenas a qualidade da população final, mas também a robustez do algoritmo de busca frente às mudanças no ambiente (BRANKE, 2012). Isso inclui o quão bem o algoritmo é capaz de detectar alterações, explorar novas regiões do espaço de busca e identificar soluções ideais à medida que avança no processo de otimização. Diversas métricas foram propostas na literatura para avaliar o desempenho de algoritmos voltados à resolução de problemas de otimização multiobjetivo dinâmicos. Neste trabalho, foram utilizadas as seguintes métricas:

- **Mean Inverted Generational Distance ($\overline{IGD}$)** (WANG; LI, 2009): essa métrica é baseada no IGD (KNOWLES; CORNE, 2002) e consiste em utilizar a média dos valores de IGD calculados um passo antes de cada mudança de ambiente:

$$IGD(P^*, P) = \frac{\sum_{v \in P} d(v, P^*)}{|P|} \quad (4)$$

$$\overline{IGD} = \frac{1}{EC} \sum_{i=1}^{EC} IGD_i \quad (5)$$

onde P^* é o conjunto Pareto Ótimo; P é o conjunto Pareto encontrado na busca; v é uma solução não dominada de P ; $d(v, P^*)$ é a distância euclidiana mínima entre v e P^* , ou seja, essa função calcula a distância entre v e cada solução de P^* e retorna a menor delas; EC é o número de mudanças de ambiente; e IGD_i é o valor calculado antes da ocorrência da próxima mudança de ambiente ($i + 1$). Essa métrica avalia simultaneamente a convergência e a diversidade, sendo que, quanto mais próximo de zero, melhor é o seu valor.

- **Hipervolume (HV)** (ZITZLER; THIELE, 1998): essa métrica mede o volume da região delimitada entre os pontos das soluções do conjunto Pareto encontrado e um ponto de referência. Matematicamente, para cada solução i pertencente ao conjunto Pareto encontrado, um hipercubo é construído em relação a um ponto de referência W_0 . A soma dos volumes desses hipercubos resulta no hipervolume. Um valor mais

elevado de HV indica melhor convergência e diversidade do algoritmo. O ponto W_0 utilizado neste trabalho corresponde à origem dos eixos do plano cartesiano.

- **Hypervolume Ratio (HVR)** (ZITZLER, 1999): essa métrica representa a razão entre os hipervolumes (ZITZLER; THIELE, 1998) do conjunto Pareto encontrado na busca (P) e do conjunto Pareto Ótimo (P^*), conforme definido a seguir:

$$HVR(P^*, W_0) = \frac{HV(P, W_0)}{HV(P^*, W_0)} \quad (6)$$

onde $HV(P, W_0)$ e $HV(P^*, W_0)$ são, respectivamente, os hipervolumes de P e P^* em relação ao ponto de referência W_0 .

- **Density (AD)** (ZHANG; QIAN, 2011): essa métrica é utilizada para medir a eficiência da distribuição das soluções não dominadas obtidas pelo MOEA nos diferentes ambientes. Quanto menor o valor da métrica, melhor é a distribuição das soluções. Considere A^t como o conjunto Pareto encontrado no ambiente t ; EC como o número de mudanças ambientais; e d_j^t como a distância entre a solução j e outra solução distinta da população que esteja o mais próxima possível no ambiente t , considerando suas respectivas projeções no espaço de objetivos. Assim, o valor de AD é dado por:

$$AD = \frac{1}{EC} \sum_{t=1}^{EC} \sqrt{\frac{1}{|A^t| - 1} \sum_{j=1}^{|A^t|} (\bar{d}^t - d_j^t)^2} \quad (7)$$

onde,

$$d_j^t = \min_{k \neq j, 1 \leq k \leq |A^t|} \{ \|f(x_j) - f(x_k)\|, (x_j, x_k) \in A^t \}, \bar{d}^t = \frac{1}{|A^t|} \sum_{j=1}^{|A^t|} d_j^t \quad (8)$$

- **Coverage of two Sets (CS)** (ZITZLER, 1999): Dados dois conjuntos de Pareto, X e Y , essa métrica avalia a relação de dominância de X sobre Y , retornando um valor entre 0 e 1. Se $CS = 1$, isso significa que todas as soluções de Y são dominadas por algum ponto de X , enquanto $CS = 0$ indica que nenhuma solução de Y é dominada por X . Note que isso não implica, necessariamente, que exista algum ponto de Y que domine uma solução de X .

Formalmente, a métrica CS é calculada da seguinte forma:

$$CS(X, Y) = \frac{|y \in Y | \exists x \in X : x \succeq y|}{|Y|} \quad (9)$$

Essa métrica permite que os algoritmos sejam comparados par a par, verificando percentualmente quantas soluções encontradas por um algoritmo são piores em relação às soluções de outro.

2.5 Problema da Mochila Multiobjetivo Dinâmico

Dada uma mochila com capacidade máxima de peso Q e um conjunto de itens disponíveis I , em que cada item possui um respectivo peso w_i e valor v_i ($i \in I$), o problema da mochila (*Knapsack Problem* - KP) (KELLERER; PFERSCHY; PISINGER, 2004) consiste em escolher um subconjunto de itens I^* , dentre aqueles pertencentes a I , de modo a maximizar o valor total dos itens selecionados. No entanto, a soma dos pesos desses itens não pode ultrapassar a capacidade da mochila (Q). Na versão multiobjetivo desse problema (*Multiobjective Knapsack Problem* - MKP) (ISHIBUCHI; AKEDO; NOJIMA, 2013), cada item $i \in I$ possui um valor $v_{i,j}$ e um peso $w_{i,j}$ para cada objetivo j . O objetivo do algoritmo é encontrar um subconjunto de itens I^* ($I^* \subseteq I$), respeitando a restrição de peso de cada objetivo (Q_j), de forma a maximizar os respectivos valores. Esse problema multiobjetivo pode ser formulado da seguinte maneira: dado um conjunto de itens I (em que $i = 0, \dots, n$) e um conjunto de objetivos J (em que $j = 1, \dots, m$), x_i é uma variável binária que assume valor 1 quando o item i está presente na mochila e 0 caso contrário, enquanto Q_j representa o peso máximo permitido para o objetivo j .

$$\sum_{j=1}^m [(max \sum_{i=0}^n v_{i,j} * x_i); (\sum_{i=0}^n w_{i,j} * x_i \leq Q_j)] \quad (10)$$

O problema da mochila multiobjetivo dinâmico 0-1 (*Dynamic Multiobjective Knapsack Problem* 0-1 - DMKP) é uma extensão do MKP, na qual os objetivos e/ou as restrições são modificados a cada mudança de ambiente (BRANKE, 2012). Alterar um objetivo significa modificar o conjunto de valores e pesos associados a ele. Já as mudanças de restrição consistem na utilização de mochilas com diferentes capacidades (limites de peso dos objetivos) a cada mudança de ambiente. Na formulação do DMKP adotada nesta tese, as alterações são realizadas apenas no conjunto de objetivos (FARINA; DEB; AMATO, 2004), pois é difícil controlar a severidade da mudança quando se altera o valor de Q_j (peso máximo). Em outras palavras, torna-se mais difícil visualizar o impacto da alteração quando a restrição é modificada. Alterando-se apenas o conjunto de objetivos, é possível realizar um experimento mais controlado, no qual se pode definir previamente o número de objetivos a serem modificados ao longo de uma execução completa. Uma mudança de alta severidade não é recomendada (BRANKE, 2012), pois pode alterar o espaço de objetivos de forma tão significativa que seria mais adequado reiniciar todo o processo de busca. Neste trabalho, adotou-se uma baixa severidade de mudança, na qual apenas um objetivo é modificado a cada mudança de ambiente. Por exemplo, em um problema com quatro objetivos $\{obj1, obj2, obj3, obj4\}$, durante uma mudança de ambiente, um dos objetivos é substituído por um novo, mantendo-se constante o número total de objetivos. Qualquer um dos quatro objetivos pode ser escolhido para substituição. Supondo que o objetivo $obj2$ seja substituído por $obj5$, o conjunto de objetivos no novo ambiente passa a ser $\{obj1, obj3, obj4, obj5\}$.

2.6 Considerações Finais

Embora os estudos sobre algoritmos evolutivos multiobjetivo aplicados a problemas dinâmicos tenham ganhado bastante destaque nos últimos anos (AZZOUZ; BECHIKH; SAID, 2017), ainda é comum encontrar, na literatura, DMOPs limitados a dois ou três objetivos. Como já discutido, quanto maior o número de objetivos envolvidos, maior é o desafio para os algoritmos evolutivos, uma vez que isso ocasiona um aumento significativo no número de soluções não dominadas, reduzindo a pressão seletiva durante a busca (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008).

Nesse sentido, esta pesquisa visa investigar abordagens evolutivas multiobjetivo que sejam eficientes e eficazes na otimização dinâmica de problemas com muitos objetivos (quatro ou mais). Os estudos foram conduzidos em instâncias do problema da mochila multiobjetivo dinâmico 0-1 (DMKP), o qual consiste em um problema discreto e dinâmico que apresenta características e conceitos presentes em diversos problemas do mundo real.

MOEAs Propostos

Neste capítulo, são descritos os algoritmos implementados neste trabalho. Inicialmente, é apresentado o algoritmo DNSGA-II*, que consiste em uma variação do algoritmo DNSGA-II (KALYANMOY; BHASKARA; SINDHYA, 2007). Em seguida, é apresentada uma adaptação do NSGA-III (LI et al., 2014) para DMOPs, denominada D-NSGA-III. As propostas seguintes correspondem a adaptações dos algoritmos MEANDS (LAFETA et al., 2018) e MEANDS-II (LAFETA; OLIVEIRA, 2019) para DMOPs. A primeira adaptação é o D-MEANDS, que preserva o mesmo mecanismo de busca do MEANDS e introduz um novo algoritmo de mudança de ambiente baseado em memória. Em seguida, é apresentado o D-MEANDS-MD, que utiliza o mesmo mecanismo de busca do MEANDS e o mesmo procedimento de mudança de ambiente do D-MEANDS, mas incorpora uma estratégia de inserção de diversidade para lidar de forma mais eficiente com DMOPs com mais de três objetivos. A adaptação seguinte é o D-MEANDS-II, que utiliza o mesmo mecanismo de busca do MEANDS-II, porém com uma redução significativa na quantidade de subpopulações. Além disso, o D-MEANDS-II emprega o mesmo mecanismo de mudança de ambiente do D-MEANDS. Por fim, é apresentado o D-MEANDS-III, que implementa o mesmo mecanismo de busca do D-MEANDS-II, mas com duas diferenças principais: utiliza um novo modelo de *crossover* e um novo algoritmo de mudança de ambiente.

3.1 DNSGA-II*

O *Dynamic Nondominated Sorting Genetic Algorithm II* (DNSGA-II) (KALYANMOY; BHASKARA; SINDHYA, 2007) é uma adaptação do conhecido algoritmo NSGA-II, proposto por Deb e colaboradores (2002), para tratar problemas dinâmicos. A versão dinâmica do algoritmo mantém o funcionamento típico do NSGA-II e incorpora uma estratégia de introdução de diversidade sempre que o ambiente utilizado para avaliar as soluções sofre alguma perturbação. Durante a mudança de ambiente, as soluções que passam a violar as restrições no novo ambiente do DMKP são ajustadas para voltar a atendê-las.

Neste trabalho, foi proposta uma nova versão do DNSGA-II, na qual a estratégia de arquivo externo (BOSMAN; THIERENS, 2003) foi introduzida. Essa variação foi denominada DNSGA-II* (LAFETÁ; OLIVEIRA, 2020B). O arquivo externo armazena todas as soluções não dominadas encontradas na população até a geração atual, mas não participa diretamente do processo evolutivo. Dessa forma, qualquer solução não dominada é mantida até o final da busca, mesmo que deixe de pertencer à fronteira superior em razão da limitação do tamanho da população. Essa estratégia é bastante comum em diferentes algoritmos multiobjetivo, como o MOEA/D (ZHANG; LI, 2007), o SPEA2 (ZITZLER; LAUMANN; THIELE, 2001), entre outros. Resultados experimentais demonstraram que o DNSGA-II* supera o DNSGA-II em métricas multiobjetivo, embora demande um tempo adicional de processamento, o qual não se mostrou significativo.

3.2 DNSGA-III

O *Non-dominated Sorting Genetic Algorithm III* (NSGA-III) é uma extensão do NSGA-II proposta para lidar com SMOPs de muitos objetivos (LI et al., 2014). O NSGA-III manteve a mesma estrutura básica do NSGA-II, baseada em fronteiras de dominância hierárquica, mas algumas adaptações foram propostas para lidar com problemas de muitos objetivos. Em vez de utilizar a métrica *crowding distance*, o algoritmo aplica um método de nichamento baseado na distância entre as soluções e pontos de referência distribuídos em um hiperplano, o que permite representar melhor a diferença entre soluções em espaços de alta dimensionalidade. Essa abordagem corresponde a uma aplicação da estratégia de pontos de referência previamente proposta em (DAS; DENNIS, 1998).

Embora o DNSGA-II tenha sido proposto com base no NSGA-II, não foi encontrada, na literatura, uma abordagem promissora que utilizasse o NSGA-III como base para a otimização dinâmica com muitos objetivos. Diante disso, foi proposta uma versão dinâmica desse algoritmo, baseada na estratégia dinâmica empregada no DNSGA-II. Essa versão é denominada, neste trabalho, DNSGA-III (LAFETÁ; OLIVEIRA, 2020B), na qual, a cada mudança de ambiente, uma porcentagem da população é gerada aleatoriamente. Nessa estratégia, durante a mudança de ambiente, 80% da população anterior é mantida e 20% é substituída por soluções geradas aleatoriamente. A Figura 1 ilustra o funcionamento dessa estratégia de criação da nova população após uma mudança de ambiente. Assim, a nova população é composta por 80% de soluções oriundas do ambiente anterior e 20% de soluções aleatórias, com o objetivo de preservar parte do conhecimento acumulado e, ao mesmo tempo, garantir diversidade no início do novo ambiente. As demais etapas do processo evolutivo seguem as mesmas estratégias utilizadas no NSGA-III, incluindo os operadores de seleção, cruzamento e mutação, bem como os mecanismos de reprodução e substituição da população (LI et al., 2014).

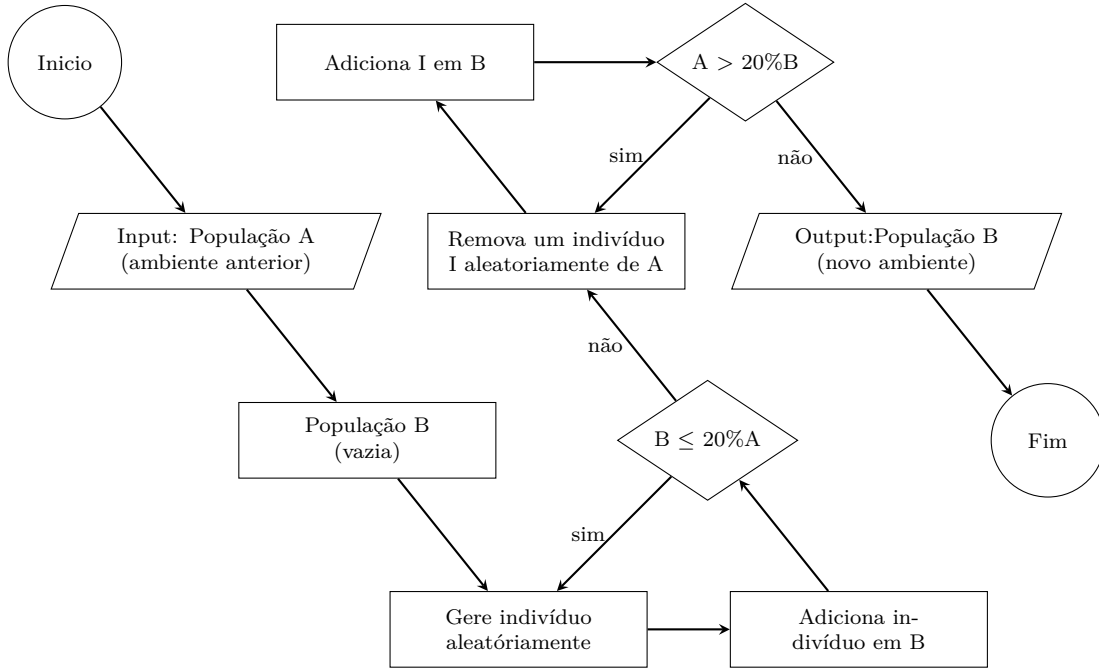


Figura 1 – Reinicialização da população após uma mudança de ambiente no DNSGA-II e DSGA-III.

3.3 D-MEANDS

O D-MEANDS foi proposto durante o desenvolvimento desta tese, sendo fortemente baseado em seu antecessor, o MEANDS.

O *Many-objective Evolutionary Algorithm Based on Nondominated Decomposed Sets* (MEANDS) (LAFETA et al., 2018) foi proposto para solucionar MaOPs discretos e estáticos. O MEANDS manipula várias subpopulações (ou subconjuntos) de soluções não dominadas. Cada subpopulação avalia a relação de dominância entre os indivíduos com base em um subconjunto de objetivos. O funcionamento básico do algoritmo é mostrado na Figura 2. O MEANDS inicializa sua população gerando uma amostra de indivíduos aleatórios de tamanho A e verificando a relação de dominância para inserir as soluções não dominadas nas subpopulações, de acordo com o subconjunto de objetivos associado a cada uma delas. O processo de inserção e remoção de indivíduos nas subpopulações é independente, pois cada subpopulação utiliza um grupo distinto de objetivos. Para compreender melhor como o conjunto de objetivos é dividido entre as subpopulações, considere um problema com m objetivos. O MEANDS gera uma subpopulação para cada subconjunto de tamanho k , com k variando de 2 até m , contemplando todas as combinações possíveis de objetivos para cada tamanho. Em outras palavras, são criadas subpopulações para todas as combinações dos m objetivos, considerando subconjuntos com 2 objetivos ($k = 2$), 3 objetivos ($k = 3$) e assim sucessivamente, até $k = m$. Cada subpopulação armazena apenas soluções não dominadas extraídas da amostra aleatória. Para isso, cada indivíduo x da amostra é avaliado sequencialmente, considerando apenas os objetivos representados

em cada subconjunto, sendo inserido na subpopulação correspondente somente se não for dominado por nenhum outro indivíduo daquela subpopulação. Além disso, se x dominar alguma outra solução y já existente na subpopulação, então y é removida dessa subpopulação. Portanto, durante a inicialização da população, apesar de ser gerada uma amostra com A indivíduos, nem todos sobreviverão.

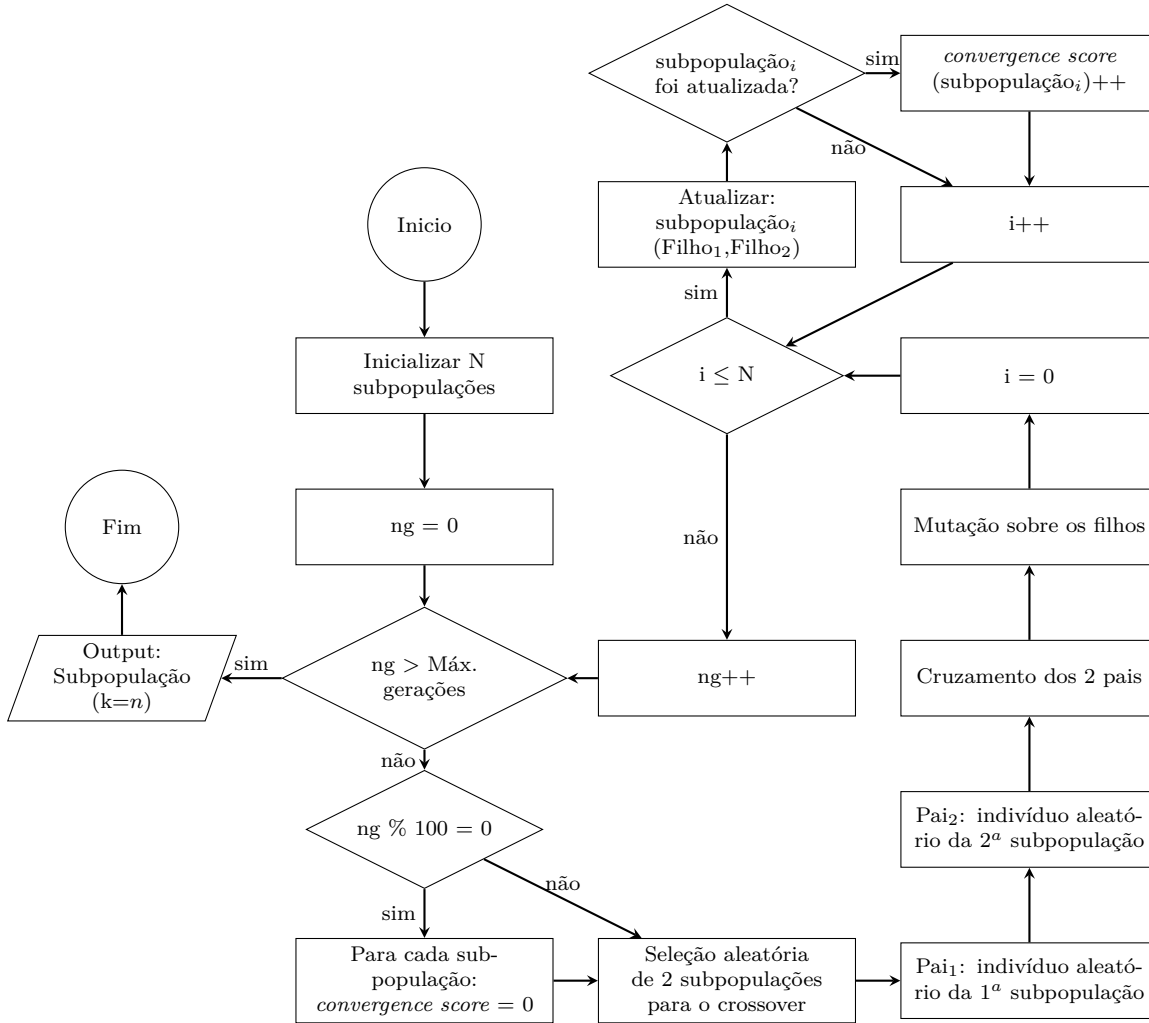


Figura 2 – Fluxo básico da estratégia evolutiva usada no MEANDS

Cada subpopulação armazena o conjunto Pareto encontrado até o momento, relacionado ao seu respectivo grupo de objetivos, ou seja, o algoritmo trabalha apenas com as soluções pertencentes à primeira fronteira encontrada para o subproblema associado a esses objetivos. Todo novo indivíduo gerado pelo cruzamento é avaliado de acordo com o subconjunto de objetivos de cada subpopulação, verificando-se sua relação de dominância em comparação com os indivíduos atualmente armazenados. O custo computacional desse processo é a principal razão pela qual o MEANDS gera apenas um indivíduo por geração. Além disso, as subpopulações são pontuadas de acordo com suas respectivas contribuições para o processo de busca, por meio da métrica denominada *convergence score*. Uma subpopulação recebe um ponto adicional em seu *convergence score* sempre que um indivíduo descendente (gerado após o *crossover*) é inserido com sucesso. Quanto

maior o *convergence score* de uma subpopulação, maior é a quantidade de novas soluções inseridas nela ao longo da busca, o que sugere que essa subpopulação pode conter soluções mais próximas do conjunto Pareto ótimo. As subpopulações com maior pontuação participam com mais frequência do processo de cruzamento. O *convergence score* é redefinido a cada 100 gerações, a fim de evitar estagnação. A seleção dos pais é dividida em duas etapas: inicialmente, são selecionadas duas subpopulações e, em seguida, escolhe-se um indivíduo de cada uma delas. A subpopulação é selecionada por torneio, com base no *convergence score*, enquanto o indivíduo é selecionado aleatoriamente em cada uma dessas subpopulações. Após a operação de *crossover*, os indivíduos descendentes percorrem todas as subpopulações. Em cada subpopulação, caso um descendente não seja dominado por nenhum dos indivíduos presentes, ele é inserido, e todas as soluções por ele dominadas são removidas. O tamanho de cada subpopulação é ilimitado e não permite a inserção de indivíduos duplicados. Ao final da execução do algoritmo, a subpopulação que considera todos os objetivos do problema ($k = m$) é retornada como o conjunto Pareto encontrado.

O *Dynamic Many-objective Evolutionary Algorithm Based on Nondominated Decomposed Sets* (D-MEANDS) (LAFETÁ; OLIVEIRA, 2020B) foi proposto para lidar com problemas dinâmicos de otimização de muitos objetivos (DMaOPs). A adaptação realizada para tratar MaOPs dinâmicos consiste no fato de que, durante as mudanças de ambiente, os indivíduos da subpopulação que considera todos os objetivos do problema ($k = m$), denominada arquivo, são utilizados para repovoar as novas subpopulações. Isto é, as subpopulações do ambiente anterior são descartadas e substituídas por novas. Observe que nem todos os indivíduos extraídos do arquivo do ambiente anterior serão reinseridos nas novas subpopulações. O aproveitamento de uma solução do ambiente anterior depende de sua não dominância em relação ao subconjunto de objetivos considerado em cada subpopulação. Além disso, indivíduos aleatórios também são gerados para repovoar as subpopulações, considerando os novos objetivos do ambiente. Durante esse processo, as soluções do arquivo que deixarem de atender às restrições do novo ambiente são adaptadas para restabelecer sua viabilidade. O Algoritmo 1 apresenta o funcionamento dessa dinâmica.

Algoritmo 1 Método de mudança de ambiente do D-MEANDS**Require:** Arquivo: Conjunto com as melhores soluções do ambiente anterior**Require:** NI: N° de indivíduos gerados na inicialização

```

1: Gerar a lista S //Cria uma lista com as subpopulações vazias
2: for (cada ind ∈ Arquivo) do
3:   ind.calcularNovoFitness() //recalcula o fitness no novo ambiente
4: end for
5: for (int i=0; i<NI; i++) do
6:   if (Arquivo não está vazio) then
7:     //Adiciona um indivíduo do arquivo nas subpopulações de S
8:     pos = rand() % Arquivo.size() //seleção aleatória de um indivíduo do Arquivo
9:     ind = Arquivo[pos]
10:    Arquivo.remove(ind) //remove o indivíduo selecionado
11:    S.adiciona(ind)
12:   else
13:     //Arquivo vazio
14:     //Adiciona um indivíduo aleatório nas subpopulações de S
15:     ind: geraCromossomo() //Gera aleatoriamente um novo indivíduo
16:     S.adiciona(ind) //adiciona o indivíduo nas subpopulações S
17:   end if
18: end for
19: return S //retorna as subpopulações de S

```

3.4 D-MEANDS-MD

O *Dynamic Many-objective Evolutionary Algorithm Based on Nondominated Decomposed Sets with Memory and Diversity* (D-MEANDS-MD) é uma extensão do D-MEANDS que introduz um novo mecanismo de cruzamento, com o objetivo de aumentar a diversidade da população ao longo das gerações. Semelhante à adaptação incorporada no D-MEANDS para lidar com problemas de otimização dinâmica, quando o ambiente sofre alguma mudança, a população é reinicializada, mas não de forma exclusivamente aleatória. Os indivíduos do arquivo atual, que considera todos os objetivos na relação de dominância, são usados para repovoar todos os subconjuntos da nova população. Portanto, essa estratégia de repovoamento aplicada em caso de mudanças pode ser vista como uma estratégia de memória, pois conserva parte da população antiga para iniciar uma nova. No entanto, para injetar alguma diversidade nessas mudanças, indivíduos aleatórios também são gerados para repovoar os subconjuntos, considerando os novos objetivos do ambiente. O número de indivíduos aleatórios é igual ao tamanho inicial da amostra A (citada na Seção 3.3) menos o número de soluções não dominadas extraídas do ambiente anterior.

No D-MEANDS, a diversidade é aplicada apenas quando a população é reinicializada logo após uma perturbação no ambiente. No entanto, após nossos experimentos iniciais com o D-MEANDS (relatados na Seção 4.3), foi possível observar que, quando um grande número de mudanças de ambiente é empregado, a convergência da população baseada na memória do ambiente anterior pode não ser tão rápida quanto o exigido por proble-

mas dinâmicos complexos. Nesses casos, mostrou-se importante injetar algum grau de aleatoriedade ao longo das gerações entre duas mudanças consecutivas. A nova variação, denominada D-MEANDS-MD, emprega um cruzamento com certo grau de aleatoriedade, controlado por um novo parâmetro de equilíbrio chamado B_{MD} . Cada *crossover* tem uma probabilidade percentual de B_{MD} de ser executado exatamente como no D-MEANDS. Caso contrário, um novo indivíduo é construído aleatoriamente para atuar como o segundo pai do cruzamento. Todos os experimentos deste trabalho utilizam $B_{MD} = 90\%$. Portanto, essa versão do MOEA emprega estratégias de memória e diversidade para equilibrar o aproveitamento de indivíduos não dominados pré-existentes com a diversidade de indivíduos totalmente novos, visando extrair o melhor desempenho da busca evolutiva. O percentual utilizado nesse balanço (10% de aleatoriedade e 90% de memória) foi ajustado empiricamente para as instâncias do problema investigadas neste trabalho (DMKP). No entanto, B_{MD} pode ser ajustado para novos cenários ou até mesmo para novos problemas dinâmicos. Destaca-se que, com exceção desse novo mecanismo de cruzamento, o restante do processo evolutivo segue as mesmas estratégias usadas pelo D-MEANDS. O Algoritmo 2 mostra o funcionamento básico do D-MEANDS-MD, destacando a mudança proposta em sua estratégia evolutiva.

Algoritmo 2 D-MEANDS-MD.

Require: MaxGer : Número de gerações; T_{mut} : Taxa de mutação; B_{MD}

```

1: Gerar a lista S //lista das subpopulações(vazia)
2: for (j=0; j < 1000; j++) do
3:   ind = geraCromossomo() //Gera um indivíduo aleatório
4:   for (cada subpopulação sp ∈ S) do
5:     //verifica se ind não é dominado por nenhum indivíduo de sp
6:     if (sp.nãoDomina(ind)) then
7:       sp.add(ind)
8:     end if
9:   end for
10: end for
11: ng = 0
12: while (ng < MaxGer) do
13:   ng++
14:   if (ng % 100 = 0) then
15:     for (cada subpopulação sp ∈ S) do
16:       sp.convergence score = 0
17:     end for
18:   end if
19:   sp1: S.Seleciona Subpopulação //Seleciona uma subpopulação
20:   tam = sp1.size()
21:   pai1 = sp1.posicao(rand()%tam) //seleção aleatória de um indivíduo da
   subpopulação
22:   if (rand() ≤ BMD) then
23:     sp2: S.Seleciona Subpopulação //Seleciona uma subpopulação (sp2 ≠ sp1)
24:     tam = sp2.size()
25:     pai2 = sp2.posicao(rand()%tam) //seleção aleatória de um indivíduo da
   subpopulação
26:   else
27:     pai2 = geraCromossomo() //Gera aleatoriamente um novo indivíduo
28:   end if
29:   [Filho1, Filho2]: Crossover(pai1, pai2)
30:   if (rand() ≤ Tmut) then
31:     Filho1.mutação()
32:   end if
33:   if (rand() ≤ Tmut) then
34:     Filho2.mutação()
35:   end if
36:   for (cada subpopulação sp ∈ S) do
37:     sp.add(Filho1) sp.add(Filho2)
38:     if (sp foi atualizada) then
39:       sp.convergence score++
40:     end if
41:   end for
42:   if (mudança ambiente = verdade) then
43:     atualizar subpopulacoes(spall, 1000) //(Execução do Algoritmo[1])
44:   end if
45: end while

```

3.5 D-MEANDS-II

O algoritmo MEANDS (LAFETA et al., 2018) foi proposto sem limitar a quantidade de indivíduos em cada subpopulação, deixando seu crescimento sem qualquer restrição de tamanho. Essa estratégia funciona bem em problemas com cardinalidade do conjunto Pareto ótimo de pequeno ou médio porte. No entanto, em problemas com cardinalidade muito grande, é necessária a adoção de algum limitador para controlar o custo computacional do processo de atualização das subpopulações. Vários MOEAs na literatura limitam o tamanho da população para encontrar uma boa diversidade no processo evolutivo. Um exemplo é o NSGA-II (DEB et al., 2002), que aplica o método *crowding distance* para selecionar um conjunto de indivíduos que apresente boa dispersão no espaço de busca. Nessa linha, foi proposto o MEANDS-II (LAFETA; OLIVEIRA, 2019), uma variação do MEANDS, que emprega um parâmetro de limite para o tamanho das subpopulações e usa o *crowding distance* para eliminar indivíduos quando o número de soluções não dominadas excede esse limite. Outra diferença entre os algoritmos está no mecanismo de pontuação das subpopulações. Como já visto, o MEANDS utiliza um sistema de pontuação baseado no *convergence score*, o qual aumenta a pontuação de uma subpopulação sempre que ela é atualizada. No entanto, no MEANDS-II, o emprego do *convergence score* é ineficaz, uma vez que um mesmo indivíduo pode ser eliminado e reinserido várias vezes, gerando pontos excedentes para esse critério. Para contornar esse problema, o algoritmo usa o *contribution score*, que é a mesma métrica de pontuação adotada pelo MEAMT (LAFETA et al., 2016). Essa métrica só aumenta a pontuação de uma subpopulação quando ela gera uma boa solução, ou seja, quando o cruzamento de um indivíduo dessa subpopulação resulta em uma nova solução (filho) que seja inserida em alguma das subpopulações do problema. Devido ao limite de tamanho da população, várias soluções não dominadas podem desaparecer durante a evolução. Essa é a razão pela qual o MEANDS-II usa um arquivo externo (que não participa da evolução) para armazenar as soluções não dominadas, considerando todos os objetivos do problema.

Outro ponto importante é que o MEANDS-II usa uma técnica de relaxamento da dominância de Pareto para aumentar a pressão seletiva (LAUMANN et al., 2002), resultando em uma melhor convergência para o conjunto Pareto ótimo. Essa técnica é chamada de *Controlling Dominance Area of Solutions* (CDAS) (SATO; AGUIRRE; TANAKA, 2007) e, como o próprio nome sugere, controla a área de dominância das soluções, possibilitando sua expansão ou contração. Destaca-se que o tamanho dessa área está diretamente associado à pressão seletiva do processo evolutivo. Em outras palavras, a expansão da área de dominância aumenta a pressão seletiva do algoritmo, enquanto sua contração provoca a sua redução.

O *Dynamic Many-objective Evolutionary Algorithm Based on Nondominated Decomposed Sets II* (D-MEANDS-II) foi proposto com a ideia de incorporar as estratégias do MEANDS-II ao mecanismo evolutivo do D-MEANDS-MD. Entretanto, nessa nova ver-

são do algoritmo, não é usada nenhuma técnica de relaxamento de dominância (como o CDAS), visto que não é necessário aumentar a pressão seletiva da busca. Em sua versão inicial, o D-MEANDS-II usa o mesmo número de subpopulações e a mesma estratégia de mudança de ambiente utilizados no D-MEANDS-MD. No entanto, a fim de diminuir o tempo de execução, foram propostas variações do algoritmo que adotam um número reduzido de subpopulações (Seção A.1). Os MOEAs antecessores, tanto nas versões estáticas (MEANDS e MEANDS-II) quanto nas dinâmicas (D-MEANDS e D-MEANDS-MD), geram subpopulações que evoluem seus indivíduos em função do subconjunto de objetivos considerado em cada uma. Por exemplo, em um problema com 4 objetivos, são geradas 6 subpopulações de 2 objetivos, uma para cada par de objetivos; 4 subpopulações de 3 objetivos; e uma subpopulação que considera todos os 4 objetivos. Cada subpopulação evolui mantendo as soluções da primeira fronteira de Pareto, considerando seu grupo de objetivos. Porém, utilizar muitos subconjuntos exige um tempo de processamento elevado para a manutenção de suas subpopulações, e esse tempo cresce exponencialmente à medida que o número de objetivos do problema aumenta. Por isso, foi proposta uma abordagem que limita a quantidade de subpopulações utilizadas durante a evolução do algoritmo. Após experimentos preliminares, descritos na Seção A.1, a versão do D-MEANDS-II proposta neste trabalho, denominada D-MEANDS-II-S2, considera apenas as subpopulações de dois objetivos no processo evolutivo, conforme descrito no Algoritmo 3.

Algoritmo 3 D-MEANDS-II-S2.**Require:** NG: Número de gerações; T_{mut} : Taxa de mutação;

```

1: Gerar a lista S (2 objetivos) //lista das subpopulações(vazias que consideram apenas
   pares de objetivos)
2: Inicializar Arquivo           //Criar subpopulação vazia que considera todos os objetivos
3: Inicializar(S)                //inicializa as subpopulações de S com indivíduos aleatórios
4: ng = 0
5: while (ng < NG) do
6:   ng++
7:   if (ng % 100 = 0) then
8:     for (cada subpopulação sp ∈ S) do
9:       sp.contribution score = 0
10:    end for
11:  end if
12:  sp1: S.Seleciona Subpopulação           //Seleciona uma subpopulação
13:  tam = sp1.size()
14:  pai1 = sp1.posicao(rand()%tam)           //Seleção aleatória de um indivíduo da
   subpopulação
15:  sp2: S.Seleciona Subpopulação           //Seleciona uma subpopulação
16:  tam = sp2.size()
17:  pai2 = sp2.posicao(rand()%tam)           //Seleção aleatória de um indivíduo da
   subpopulação
18:  [Filho1, Filho2]: Crossover(pai1, pai2)
19:  if (rand() ≤  $T_{mut}$ ) then
20:    Filho1.mutação()
21:  end if
22:  if (rand() ≤  $T_{mut}$ ) then
23:    Filho2.mutação()
24:  end if
25:  for (cada subpopulação sp ∈ S) do
26:    sp.add(Filho1) sp.add(Filho2)
27:    Arquivo.add(Filho1) Arquivo.add(Filho2)
28:  end for
29:  if (Alguma subpopulação sp ∈ S foi atualizada) then
30:    sp1.contribution score++
31:    sp2.contribution score++
32:  end if
33:  if (mudança ambiente = verdade) then
34:    atualiza subpoputalcoes(Arquivo, 1000)           //Execução do Algoritmo [1]
35:  end if
36: end while

```

3.6 D-MEANDS-III

O *Dynamic Many-objective Evolutionary Algorithm Based on Nondominated Decomposed Sets III* (D-MEANDS-III) é uma extensão do D-MEANDS-II que visa explorar melhor o modelo multipopulacional do algoritmo, por meio de duas alterações em seu

mecanismo de busca: uma no método de seleção dos pais para o *crossover* e a outra no método de mudança de ambiente.

No D-MEANDS-II e em suas versões anteriores, o *crossover* segue três passos:

- i. seleção de duas subpopulações distintas, que participarão do cruzamento, com base no *contribution score*;
- ii. seleção aleatória de um indivíduo pai de cada subpopulação escolhida;
- iii. realização do *crossover* entre esses dois pais.

Nota-se que esses algoritmos investem em uma metodologia mais complexa para escolher as subpopulações do que para selecionar os indivíduos pais dentro dos respectivos conjuntos. A proposta do D-MEANDS-III é empregar uma estratégia mais elaborada para a seleção dos indivíduos pais (passo ii), em que, para cada subpopulação escolhida (passo i), busca-se selecionar o indivíduo que esteja mais próximo ao centro daquele conjunto de soluções não dominadas. Essa estratégia, denominada Tour4Ind, consiste em selecionar 4 indivíduos da subpopulação e calcular a distância entre eles no espaço de objetivos. Em seguida, é contabilizada a distância total de cada indivíduo, que corresponde à soma de suas distâncias em relação aos demais. O indivíduo que tiver a menor distância total tende a ser o mais centralizado entre os indivíduos considerados e, portanto, é o que possui maior chance de estar próximo ao centro da subpopulação. Como os centros das subpopulações são distintos, já que consideram subconjuntos de objetivos diferentes, aumenta-se a chance de selecionar pais mais distantes entre si e, conseqüentemente, favorecer a diversidade. Portanto, esse indivíduo é selecionado para o *crossover*. A Tabela 1 exemplifica esse processo, mostrando as distâncias individuais (entre pares de soluções) e a distância total (soma das distâncias), geradas para quatro indivíduos escolhidos aleatoriamente de uma subpopulação específica. Nesse exemplo, o indivíduo 2 (destacado em vermelho) é considerado o mais centralizado, já que possui a menor distância total, sendo escolhido para representar aquela subpopulação no cruzamento.

Tabela 1 – Exemplo de distâncias entre 4 indivíduos de uma subpopulação.

	Indivíduo 1	Indivíduo 2	Indivíduo 3	Indivíduo 4	Distância Total
Indivíduo 1	0	1.41	5.83	8.48	15.72
Indivíduo 2	1.41	0	4.47	7.07	12.95
Indivíduo 3	5.83	4.47	0	3.16	13.46
Indivíduo 4	8.48	7.07	3.16	0	18.71

O funcionamento desse novo método de seleção de pais é apresentado no Algoritmo 4.

Para se ajustarem ao dinamismo do problema, os algoritmos evolutivos propostos neste trabalho (D-MEANDS e suas variações) reconstroem a população a cada mudança de ambiente, usando um método baseado em memória (descrito na Seção 3.3). Esse tipo

Algoritmo 4 Método de seleção de pais Tour4Ind.

Require: ubpopulação (sp) de onde será escolhido um indivíduo para participar do crossover

```

1: tam = sp.size()
2: if (tam < 4) then
3:   //Retorna um indivíduo aleatório de sp
4:   pos = rand()
5:   return sp[pos]
6: end if
7: //Seleção aleatória de 4 indivíduos de sp
8: distinct positions(0,tam-1)           //Gera 4 posições distintas entre 0 e tam-1
9: ind1 = sp[pos1]
10: ind2 = sp[pos2]
11: ind3 = sp[pos3]
12: ind4 = sp[pos4]
13: dist.initialize()    //Cria um vetor zerado para armazenar as distâncias totais dos 4
    indivíduos
14: for (int i=0; i<3; i++) do
15:   for (int j=i+1; i<4; i++) do
16:     d = distancia euclidiana(indi, indj)
17:     dist[i] += d
18:     dist[j] += d
19:   end for
20: end for
21: //procura pelo menor valor no vetor dist[]
22: index = 0
23: for (int i=1; i<4; i++) do
24:   if (dist[index] > dist[i]) then
25:     index = i
26:   end if
27: end for
28: return indindex

```

de estratégia resgata informações do ambiente anterior para reconstruir a população no novo ambiente. Nesse sentido, nossos MOEAs utilizam os indivíduos armazenados no arquivo externo, o qual contempla todas as soluções da primeira fronteira de Pareto do problema, para reconstruir a nova população. Essa abordagem de mudança de ambiente é modificada no D-MEANDS-III para filtrar as soluções do arquivo externo que serão introduzidas no novo ambiente. O intuito é resgatar apenas os indivíduos que estejam centralizados ou localizados nas extremidades do conjunto de soluções não dominadas do problema (arquivo externo), considerando suas respectivas projeções no espaço de objetivos. Assim, reduz-se a possibilidade de haver muitas soluções próximas em um único ponto no espaço de objetivos. Com as soluções mais espalhadas, aumenta a possibilidade de cruzar dois indivíduos distantes no espaço de objetivos. Com isso, a tendência é gerar descendentes que explorem novas regiões no espaço de busca, além de contribuir para

o aumento da dispersão das soluções no conjunto não dominado (TOFFOLO; BENINI, 2003). Também é importante ressaltar que, com a aplicação desse filtro, há uma redução natural na quantidade de soluções do arquivo que são reaproveitadas no novo ambiente, demandando um número maior de indivíduos gerados aleatoriamente. Dessa forma, a estratégia também fortalece a diversidade da busca. Vale lembrar que as subpopulações (conjuntos) do algoritmo guardam apenas soluções não dominadas; portanto, os indivíduos do arquivo que não atendem a esse critério são descartados. Assim, mesmo que uma solução passe pelo filtro inicial (posicionamento dentro do conjunto), não há garantia de sua permanência na nova população.

A construção do filtro usado pelo algoritmo nas mudanças de ambiente envolve duas etapas: calcular um ponto de referência (PR), que fica próximo do centro da população do arquivo externo, e determinar a área de aceitação. Ambas as etapas são baseadas na projeção das soluções de cada subpopulação no espaço de objetivos. Vale ressaltar que, embora as subpopulações evoluam com base apenas em seus dois objetivos, seus indivíduos são avaliados em relação a todos os objetivos do problema, e os valores resultantes são considerados em seus respectivos posicionamentos no espaço de objetivos. As ações pertinentes a cada etapa são descritas a seguir:

A primeira etapa começa com a definição do ponto central de cada conjunto de soluções (subpopulação), o que envolve o cálculo da média aritmética dos respectivos pontos no espaço de objetivos. Em seguida, é calculada a média aritmética dos pontos centrais das subpopulações, a fim de determinar o ponto de referência (PR), que corresponde ao ponto central da população analisada. Esse processo é ilustrado na Figura 3, considerando um problema com 4 objetivos. Os pontos pretos representam o posicionamento das soluções de cada subpopulação i ($1 \leq i \leq 6$) no espaço de objetivos. Os pontos vermelhos indicam os pontos centrais (C_i) dos conjuntos de soluções, enquanto o ponto azul representa o ponto de referência PR resultante.

O primeiro passo da segunda etapa envolve o cálculo da distância de referência (DR). Ela é obtida a partir da média aritmética das distâncias euclidianas entre os pontos centrais dos conjuntos de soluções e o ponto de referência. A área de aceitação é baseada no valor da distância de referência, de modo que uma solução X só passa pelo filtro se atender a um dos seguintes critérios:

- A distância entre X e PR for menor ou igual a $DR/2$; ou
- A distância entre X e PR for maior ou igual a DR .

Nos demais casos, a solução X é descartada.

Este método, chamado de *Filtro-PR*, é utilizado pelo D-MEANDS-III durante a mudança de ambiente com o objetivo de selecionar as soluções que estão mais distantes de PR e as que estão mais próximas do ponto de referência. Essas soluções serão inseridas na

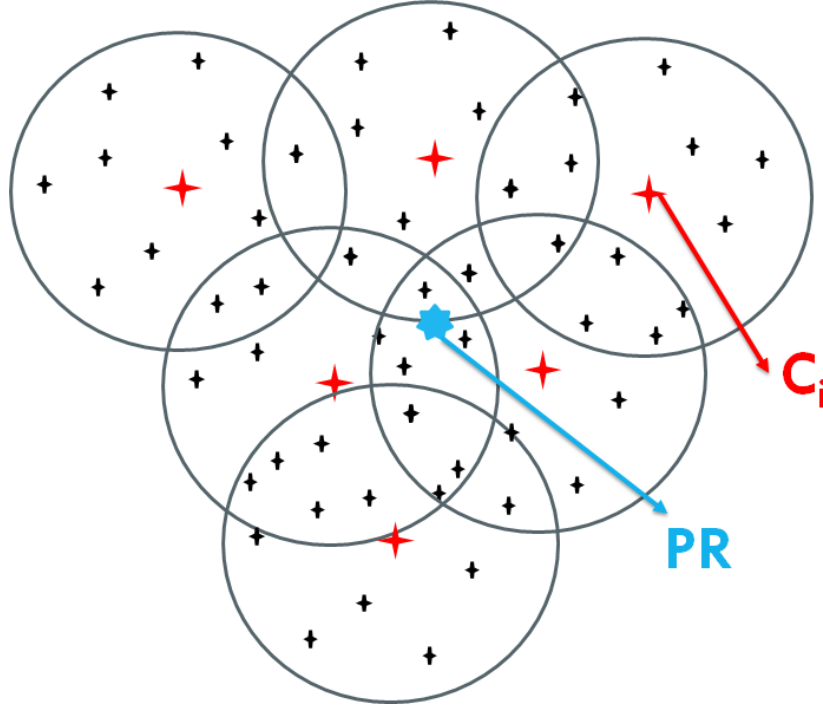


Figura 3 – Primeira etapa da construção do filtro de soluções do método *Filtro – PR*.

população do próximo ambiente, visando garantir uma distribuição dos pontos no espaço de objetivos que favoreça a diversidade durante a busca.

Tanto o método *Tour4Ind* quanto o método *Filtro-PR* apresentaram melhorias significativas no comportamento do D-MEANDS-III. A Seção A.2 mostra o resultado de um experimento com o *Tour4Ind*, no qual houve melhora nos valores das métricas HV e AD, com um pequeno aumento no tempo de execução. Na Seção A.3, foram realizados alguns experimentos com o *Filtro-PR*, nos quais houve uma melhora significativa no tempo de execução, mas com uma ligeira queda nas demais métricas. A proposta do D-MEANDS-III é manter a mesma dinâmica do D-MEANDS-II, na qual são utilizados apenas conjuntos de dois objetivos, juntamente com o *Tour4Ind* e o *Filtro-PR*. Espera-se que o equilíbrio entre esses dois métodos melhore o desempenho do algoritmo tanto nas métricas quanto no tempo de execução.

Experimentos e Análise dos Resultados

Neste capítulo, algoritmos evolutivos multiobjetivo dinâmicos (DMOEAs), já aplicados a problemas de otimização multiobjetivo dinâmicos contínuos (DMOPs contínuos), são investigados aqui com o objetivo de verificar se eles também são capazes de resolver, de forma eficiente, um problema de otimização multiobjetivo dinâmico discreto (DMOP discreto). Os experimentos estão divididos em fases, em que cada fase apresenta os resultados de uma parte fundamental da pesquisa desta tese. As fases estão organizadas de acordo com a evolução de nossa investigação sobre DMOEAs.

Na primeira fase, quatro algoritmos (dos quais três são da literatura) foram investigados: DNSGA-II, DNSGA-III, MOEA/D-KF e MS-MOEA. Na sequência, uma análise mais aprofundada do DNSGA-II nos levou a propor uma pequena alteração no algoritmo; resultando em uma nova versão que foi chamada de DNSGA-II*. Nessa fase, a análise comparativa dos algoritmos foi conduzida considerando instâncias mais simples do problema e baseada em duas métricas paramétricas: *Mean Inverted Generational Distance* ($\overline{IGD}$) (WANG; LI, 2009) e *Hypervolume Ratio* (HVR) (ZITZLER, 1999).

Na segunda fase dos experimentos, o algoritmo D-MEANDS (Seção 3.3) foi proposto e seu desempenho foi comparado aos algoritmos investigados na etapa anterior. Nas análises finais, também foi considerado um novo algoritmo, denominado DDIS-MOEA/D-DE (LIU et al., 2020). Por se tratar de um algoritmo evolutivo mais recente, decidimos incluí-lo nos experimentos. Além das métricas usadas na fase anterior (HVR e $\overline{IGD}$), nas análises finais dessa fase, também foram introduzidas as métricas não paramétricas *Density* (AD) (ZHANG; QIAN, 2011) e Hipervolume (HV) (ZITZLER; THIELE, 1998). O uso dessas métricas não paramétricas possibilita a análise de instâncias mais complexas e com maior número de mudanças de ambiente, visto que não é necessário qualquer conhecimento prévio do conjunto Pareto ótimo para os cenários investigados.

Na terceira fase, é introduzido o D-MEANDS-MD (Seção 3.4), que teve seu desempenho avaliado, com base nas métricas não paramétricas *Density* (AD) e Hipervolume (HV) (ZITZLER; THIELE, 1998).

A validação da eficiência e eficácia do D-MEANDS-II (Seção 3.5) é apresentada na

quarta fase dos experimentos. Esse algoritmo é proposto com o objetivo de superar o desempenho em tempo de execução do D-MEANDS-MD e do D-MEANDS, sem comprometer significativamente a qualidade do conjunto Pareto encontrado. Para comparar o desempenho multiobjetivo dos algoritmos, também foram utilizadas as métricas não paramétricas AD e HV. Além disso, é apresentada uma análise comportamental dos algoritmos D-MEANDS-MD e D-MEANDS-II quanto à variação do tempo de execução à medida que o número de objetivos aumenta.

Na quinta fase dos experimentos, são apresentados os resultados alcançados com o algoritmo D-MEANDS-III (Seção 3.6), o qual apresenta uma dinâmica que aprofunda o uso da característica multipopulacional do próprio algoritmo. Para avaliar seu desempenho, foram usadas as mesmas métricas de desempenho e instâncias do problema adotadas na fase anterior.

Por fim, experimentos adicionais foram realizados a fim de comparar a capacidade de convergência entre os melhores algoritmos analisados nas cinco fases, usando a métrica *Coverage of Two Sets* (CS) (ZITZLER, 1999). Também foi feita uma análise mais aprofundada do comportamento do Hipervolume (HV), apresentada com o objetivo de compreender as características necessárias no conjunto Pareto encontrado para se obter um HV melhor.

4.1 Metodologia dos Experimentos

O problema discreto utilizado em todas as fases é o *Dynamic Multiobjective Knapsack Problem* 0-1 (DMKP), considerando uma formulação em que ocorre apenas a variação de objetivos a cada mudança de ambiente. Investigou-se instâncias com 30, 50 e 100 itens, usando formulações com 4, 6 e 8 objetivos, em cenários dinâmicos que envolvem 1, 2, 5, 10, 15 e 20 mudanças de ambiente (ECs). Na primeira fase dos experimentos, foram utilizados cenários com até 10 ECs, enquanto as demais fases empregaram instâncias mais complexas (de 10 a 20 ECs). As mudanças ocorrem ao longo das gerações do algoritmo, em intervalos igualmente distribuídos. Por exemplo, em um cenário com apenas uma mudança de ambiente ($EC = 1$), a primeira metade da busca é guiada por um ambiente (pré mudança), enquanto a metade final considera outro (pós mudança).

Em todos os experimentos, os algoritmos foram executados 100 vezes para cada instância considerada na fase. Em cada execução, foram aferidos tanto o tempo de processamento quanto os valores das métricas de desempenho multiobjetivo, calculados a partir do conjunto de Pareto encontrado. A análise comparativa entre os algoritmos foi realizada com base nos valores médios e nos desvios padrão dessas métricas, considerando o conjunto das 100 execuções. Adicionalmente, foi aplicado um teste de hipóteses para verificar se a superioridade observada de um algoritmo em relação aos demais possui relevância estatística ou se é decorrente do ruído inerente à natureza estocástica dos métodos

evolutivos. Neste trabalho, foi utilizado o teste t-student (MANKIEWICZ, 2000), com nível de significância de 0,01, por ser amplamente empregado na comparação de médias e adequado para amostras independentes com variância aproximadamente normal.

4.2 Fase 1: Análise Comparativa de Desempenho de Métodos Evolutivos da Literatura

Nesta seção, realiza-se a análise comparativa do o desempenho de alguns algoritmos evolutivos encontrados na literatura, considerando as métricas *Mean Inverted Generational Distance* ($\overline{IGD}$) e *Hypervolume Ratio* (HVR). Os algoritmos investigados (DNSGA-II, DNSGA-III, MOEA/D-KF e MS-MOEA) divergem quanto à estratégia dinâmica adotada. Eles foram implementados em linguagem JAVA e executados na mesma máquina (processador Intel Core i5-6400 2.7 GHz e 8GB de RAM).

Os algoritmos foram executados com os seguintes parâmetros: 100 gerações, população de 100 indivíduos, 100% de cruzamento e 10% de mutação. Considerando esses parâmetros, o número de indivíduos avaliados é o mesmo para todos os algoritmos. Foram utilizadas instâncias com 4, 6 e 8 objetivos (m), considerando 30 e 50 itens do DMKP. Além disso, foram avaliados cenários dinâmicos com 1 e 2 mudanças de ambiente (EC). Para os algoritmos DNSGA-II e DNSGA-III, 20% da população foi substituída a cada mudança de ambiente. No MS-MOEA, a cada mudança de ambiente, 20% da população é composta por indivíduos do arquivo que sofreram mutação, enquanto os 80% restantes são indivíduos aleatórios. O MOEA/D-KF utiliza 10 vizinhos para formar as subpopulações.

4.2.1 Comparando Métodos da Literatura com $EC=1$

Inicialmente, é realizada uma análise comparativa entre o desempenho dos MOEAs investigados, considerando instâncias em que a mudança de ambiente ocorre uma única vez, na geração 51 (metade da execução do AG). A partir dos resultados experimentais, nota-se que o MS-MOEA supera os demais algoritmos em relação às métricas multiobjetivo, seguido pelo DNSGA-II. Enquanto o MOEA/D-KF, no geral, retornou os piores valores para as métricas, embora seu desempenho no HVR melhore à medida que o número de objetivos aumenta. Uma descrição mais detalhada desses resultados experimentais é apresentada no Apêndice B (Seção B.1.1).

A fim de verificar se as diferenças de desempenho são significativas, foi aplicado um teste de hipótese *t-student* (MANKIEWICZ, 2000), comparando o algoritmo que alcançou os melhores valores para as métricas (MS-MOEA) com os demais. Esse teste também foi empregado para validar a superioridade do DNSGA-II (segundo melhor algoritmo) em relação ao DNSGA-III e ao MOEA/D-KF. Ambos os testes foram conduzidos, considerando 99% de confiança.

A Tabela 2 mostra os resultados do teste de hipótese que adota o MS-MOEA como base de comparação. As células verdes indicam que o MS-MOEA foi significativamente melhor (retorna valores maiores para HVR e menores para $\overline{IGD}$), enquanto as células brancas indicam que não houve diferença significativa. Analisando essa tabela, observa-se que o MS-MOEA foi significativamente melhor em todas as instâncias, considerando tanto HVR quanto $\overline{IGD}$.

Tabela 2 – Teste de hipótese comparando MS-MOEA aos demais algoritmos em instâncias com uma mudança de ambiente (EC=1).

		Objetivos					
		4		6		8	
Algoritmos	Itens	HVR	$\overline{IGD}$	HVR	$\overline{IGD}$	HVR	$\overline{IGD}$
DNSGA-II	30	>	<	>	<	>	<
	50	>	<	>	<	>	<
DNSGA-III	30	>	<	>	<	>	<
	50	>	<	>	<	>	<
MOEA/D-KF	30	>	<	>	<	>	<
	50	>	<	>	<	>	<

A Tabela 3 mostra os resultados do teste de hipótese que adota o DNSGA-II como base de comparação (o MS-MOEA é omitido, uma vez que já foi comparado ao DNSGA-II na tabela anterior). Como pode ser observado, no geral, o DNSGA-II foi significativamente melhor que o DNSGA-III e o MOEA/D-KF.

Tabela 3 – Teste de hipótese comparando DNSGA-II aos demais algoritmos em instâncias com uma mudança de ambiente (EC=1)

		Objetivos					
		4		6		8	
Algoritmos	Itens	HVR	$\overline{IGD}$	HVR	$\overline{IGD}$	HVR	$\overline{IGD}$
DNSGA-III	30	>	<	>	<	>	<
	50	>	=	>	=	>	<
MOEA/D-KF	30	>	<	>	<	>	=
	50	>	<	>	<	>	<

Também foi analisado o tempo de execução dos algoritmos evolutivos testados. A Figura 4 mostra o tempo médio de processamento de cada algoritmo para cada instância analisada neste experimento. O MS-MOEA apresenta, claramente, o maior tempo de execução, sendo que o aumento do número de itens e objetivos torna sua execução mais demorada. O MOEA/D-KF mostrou-se o mais rápido entre os algoritmos investigados. Os algoritmos DNSGA-II e DNSGA-III demandam tempos de processamento semelhantes, no entanto, mas com o DNSGA-III ficando mais lento a medida que o número de objetivos aumenta.

Portanto, embora o MS-MOEA tenha apresentado o melhor desempenho nas métricas $\overline{IGD}$ e HVR , esse bom desempenho demanda um tempo de processamento elevado em

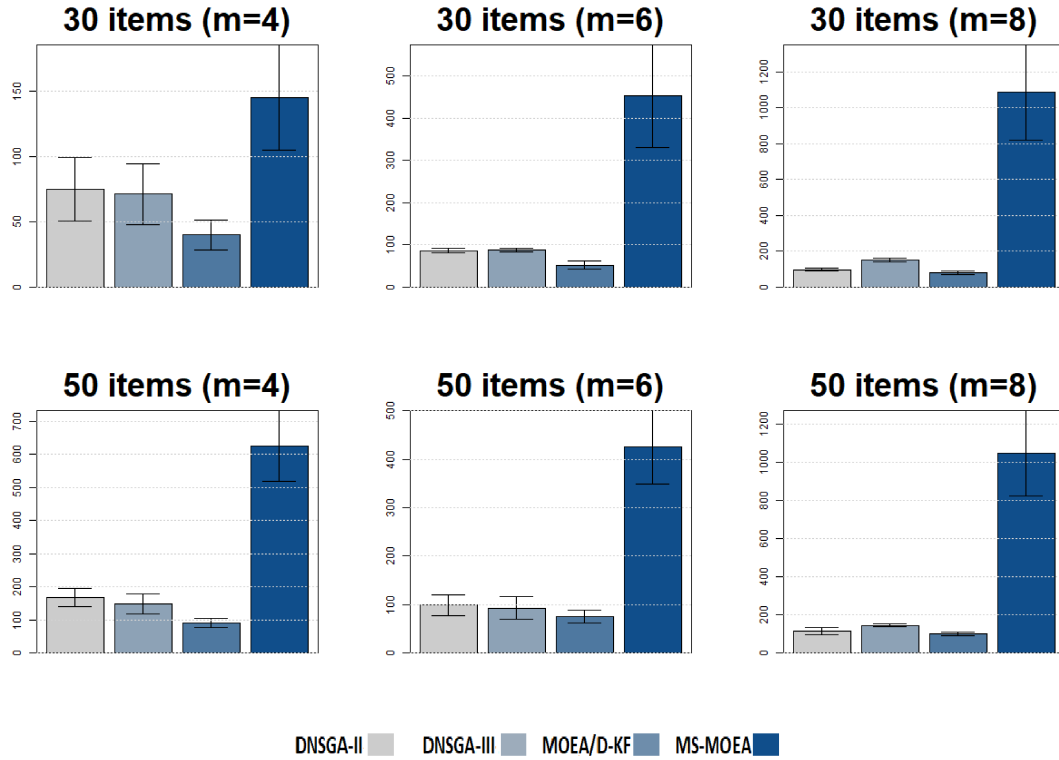


Figura 4 – Tempo de processamento para instâncias DMKP ($EC=1$): 30 e 50 itens.

relação aos demais algoritmos. Por outro lado, o DNSGA-II é o segundo melhor método, em relação às métricas multiobjetivo, e é significativamente mais rápido que o MS-MOEA.

4.2.2 Comparando Métodos da Literatura com $EC=2$

Nesta seção, serão analisados os resultados obtidos pelos algoritmos ao tratar instâncias do DMKP com duas mudanças. Neste experimento, essas mudanças ocorrem nas gerações 34 e 67, ou seja, a cada um terço da execução do algoritmo. Os gráficos com os valores médios das métricas HVR e $\overline{IGD}$ alcançados pelos MOEAs analisados são apresentados no Apêndice B (Seção B.1.2).

Observando os resultados dos testes de hipótese t , apresentados nas Tabelas 4 e 5, pode-se observar que a análise comparativa dos quatro algoritmos é muito semelhante à dos experimentos com uma única mudança de ambiente ($EC=1$). A análise do tempo de processamento também é bastante semelhantes, como mostrado na Figura 5.

Tabela 4 – Teste de hipótese comparando MS-MOEA aos demais algoritmos em instâncias com uma mudança de ambiente (EC=2)

Algoritmos	Itens	Objetivos					
		4		6		8	
		HVR	$\overline{IGD}$	HVR	$\overline{IGD}$	HVR	$\overline{IGD}$
DNSGA-II	30	=	=	>	<	>	<
	50	>	<	>	<	>	<
DNSGA-III	30	>	=	>	<	>	<
	50	>	<	>	<	>	<
MOEA/D-KF	30	>	<	>	<	>	<
	50	>	<	>	<	>	<

Tabela 5 – Teste de hipótese comparando DNSG-II aos demais algoritmos em instâncias com uma mudança de ambiente (EC=2)

Algoritmos	Itens	Objetivos					
		4		6		8	
		HVR	$\overline{IGD}$	HVR	$\overline{IGD}$	HVR	$\overline{IGD}$
DNSGA-III	30	>	=	>	<	>	<
	50	>	=	>	=	>	<
MOEA/D-KF	30	>	<	>	<	>	<
	50	>	<	>	<	>	<

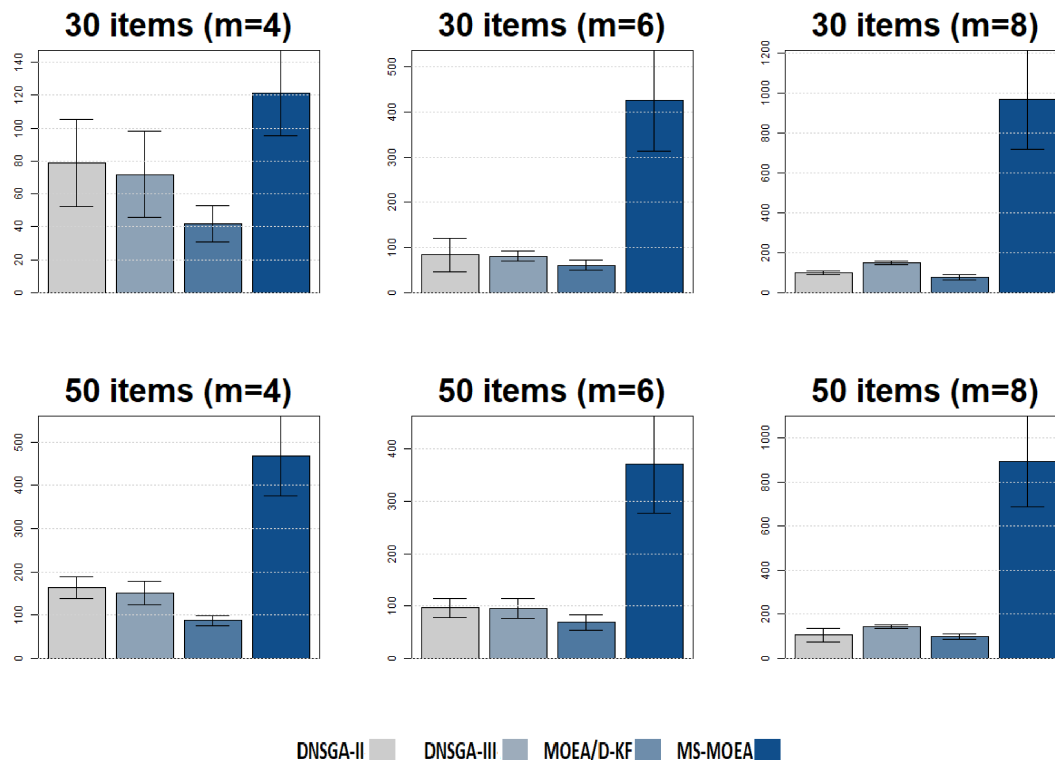


Figura 5 – Tempo de processamento para instâncias DMKP (EC=2): 30 e 50 itens.

4.2.3 Análise de Desempenho do DNSGA-II*

Embora o MS-MOEA supere os demais algoritmos considerando as métricas HVR e $\overline{IGD}$, ele demanda mais tempo de execução, sendo que o mesmo número de avaliações de aptidão foi utilizado para todos os algoritmos. Por outro lado, o DNSGA-II apresentou um tempo de execução muito menor (principalmente nas instâncias com 6 e 8 objetivos) e um bom desempenho nas métricas multiobjetivo, superando os outros dois algoritmos investigados: DNSGA-III e MOEA/D-KF. Decidiu-se por investigar se mesmo com um aumento no tempo de execução do DNSGA-II, a atualização de parâmetros mais robustos (maior tamanho da população e/ou maior número de gerações) poderia ajudá-lo a superar o MS-MOEA. Esses experimentos foram realizados usando a instância de 30 itens e 8 objetivos, pois a instância com mais objetivos retornaria resultados mais discrepantes entre MS-MOEA e DNSGA-II, além de ser mais demorada.

Primeiro, realizou-se algumas execuções preliminares com o DNSGA-II apenas aumentando o número de indivíduos e/ou o número de gerações. Embora ambas as modificações aumentem o tempo de execução do DNSGA-II (como esperado), ficou claro que as métricas eram mais sensíveis a um incremento no tamanho da população (T_p). Portanto, novas execuções foram realizadas, variando seu tamanho populacional ($T_p = 200, 300, 400$ e 500), mantendo o número de gerações fixo em 100. A Figura 6 mostra esses resultados, onde também são replicados os resultados de DNSGA-II e MS-MOEA nos experimentos anteriores com 100 indivíduos.

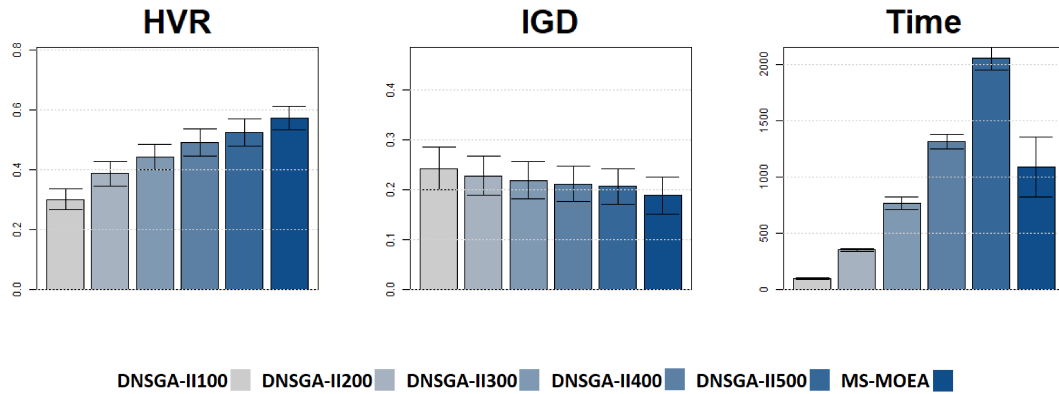


Figura 6 – Desempenho do DNSGA-II em função do tamanho populacional, considerando uma instância do problema com 30 itens e 8 objetivos.

Observa-se que o aumento sucessivo do tamanho da população faz com que o DNSGA-II melhore nas métricas em relação aos resultados iniciais. No entanto, mesmo utilizando $T_p = 500$, não foi possível alcançar os resultados do MS-MOEA. Além disso, a partir de 400 indivíduos, o DNSGA-II apresenta um tempo de execução maior que o do MS-MOEA.

A principal explicação para essa limitação do DNSGA-II em comparação ao MS-MOEA é o emprego de um arquivo externo irrestrito no MS-MOEA para armazenar todas as soluções não dominadas encontradas durante as gerações, enquanto o DNSGA-

II armazena as soluções em um tamanho restrito. Dessa forma, a população atual e algumas soluções de Pareto podem ser perdidas ao longo da evolução. De fato, o número de soluções não dominadas no arquivo externo apresentado nas execuções do MS-MOEA foi em torno de 1.000 para a instância de 8 objetivos e 30 itens. Portanto, para armazenar tantas soluções não dominadas, o DNSGA-II precisaria ser executado com $T_p = 1000$, o que não é prático e aumentaria consideravelmente o tempo de execução em relação ao MS-MOEA.

Como alternativa promissora, implementamos uma modificação no DNSGA-II para incluir também um arquivo externo, de forma semelhante ao MS-MOEA. Chamamos essa versão de DNSGA-II* e realizamos experimentos adicionais mostrados na Figura 7, aumentando seu tamanho populacional ($T_p = 100, 200, 300, 400$ e 500), mantendo 100 gerações. É possível observar uma melhoria significativa em todas as execuções do DNSGA-II* em comparação com o DNSGA-II padrão. Devido à inclusão do arquivo externo, os resultados das métricas HVR e $\overline{IGD}$ (principalmente HVR) apresentaram melhorias, com apenas um pequeno aumento no tempo de processamento. Na versão do DNSGA-II* com $T_p = 400$, foi possível aproximar os resultados do MS-MOEA, mas com um tempo de execução superior.

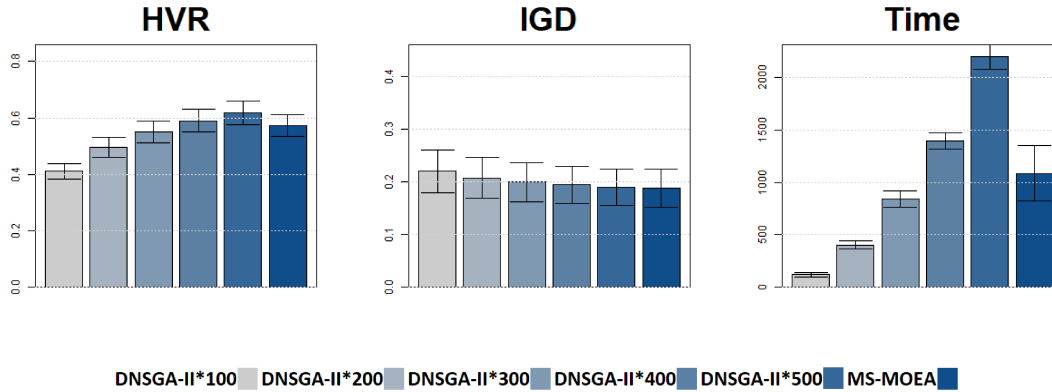


Figura 7 – Desempenho do DNSGA-II* em função do tamanho populacional, considerando uma instância do problema com 30 itens e 8 objetivos.

Os resultados e as análises apresentadas nas seções 4.2.1, 4.2.2 e 4.2.3 foram utilizados na elaboração do artigo (LAFETÁ; OLIVEIRA, 2020A).

4.3 Fase 2: Análise de Desempenho do Algoritmo D-MEANDS

A partir da investigação dos métodos propostos na literatura, principalmente o DNSGA-II (e sua variação com arquivo externo, DNSGA-II*) e o MS-MOEA, um novo MOEA foi proposto, incorporando estratégias dinâmicas ao *framework* do algoritmo MEANDS (LAFETA et al., 2018), criado originalmente para problemas estáticos. Esse novo MOEA foi denominado D-MEANDS (LAFETÁ; OLIVEIRA, 2020B) e visa lidar com DMOPs com muitos objetivos. Investigamos sua aplicação no DMKP com o objetivo de avaliar o desempenho desse novo algoritmo em um problema de otimização discreta multiobjetivo dinâmico, além de compará-lo aos métodos propostos na literatura. Como base de comparação, utilizamos dois algoritmos anteriormente aplicados ao DMKP: DNSGA-II e MS-MOEA, visto que esses foram os que mais se destacaram nas análises apresentadas anteriormente (Seções 4.2.1, 4.2.2 e 4.2.3).

Como o D-MEANDS realiza apenas um cruzamento por geração, seus parâmetros foram escolhidos de modo a realizar o mesmo número de cruzamentos que os demais algoritmos (DNSGA-II e MS-MOEA), sendo: 10.000 gerações, 10% de mutação e 1.000 indivíduos utilizados para inicializar a população. Nessa fase, cada algoritmo foi executado 100 vezes para cada instância e os resultados experimentais são apresentados no Apêndice B.2. Os gráficos mostram a média e o desvio padrão dessas 100 execuções.

4.3.1 Cenários com uma Única Mudança de Ambiente

Primeiramente, analisamos o desempenho dos MOEAs quando ocorre uma única mudança de ambiente ($EC = 1$). A mudança acontece na metade do número de gerações: para DNSGA-II e MS-MOEA, a alteração ocorre na geração 51; para D-MEANDS, ocorre na geração 5.001.

O teste de hipótese t foi realizado para avaliar a significância alcançados pelo resultados do D-MEANDS (algoritmo proposto) em relação aos demais. O teste foi conduzido com 99% de confiança, e os resultados podem ser observados na Tabela 6 para instâncias de 30 e 50 itens, com 4 e 6 objetivos. Cada linha da tabela apresenta a comparação do D-MEANDS com outro algoritmo, e cada coluna indica a métrica comparada. As células verdes mostram que o D-MEANDS foi significativamente melhor, enquanto as células vermelhas indicam que foi significativamente pior. As células em branco representam que não houve diferença significativa. O D-MEANDS superou o DNSGA-II, e o MS-MOEA apresentou desempenho similar ao D-MEANDS, com alguma vantagem na métrica HVR .

Ao analisar o tempo de execução dos algoritmos apresentado na Figura 8, observa-se que o MS-MOEA apresenta um tempo de execução muito maior que o DNSGA-II em todas as instâncias. Concluímos que o D-MEANDS apresenta bom desempenho nas

Tabela 6 – Teste de hipótese comparando D-MEANDS aos demais algoritmos em cenários com uma mudança de ambiente (EC=1).

Algoritmos	Items	Objetivos			
		4		6	
		HVR	$\overline{IGD}$	HVR	$\overline{IGD}$
DNSGA-II	30	>	<	>	<
	50	>	<	>	<
MS-MOEA	30	=	=	<	=
	50	<	=	=	=

métricas HVR e $\overline{IGD}$, aliado a um tempo de execução eficiente nas instâncias com 4 e 6 objetivos. Considerando as instâncias com 4 objetivos, o tempo de execução do D-MEANDS é menor que o do DNSGA-II. Entretanto, o aumento do número de objetivos requer um incremento considerável no tempo de execução do D-MEANDS.

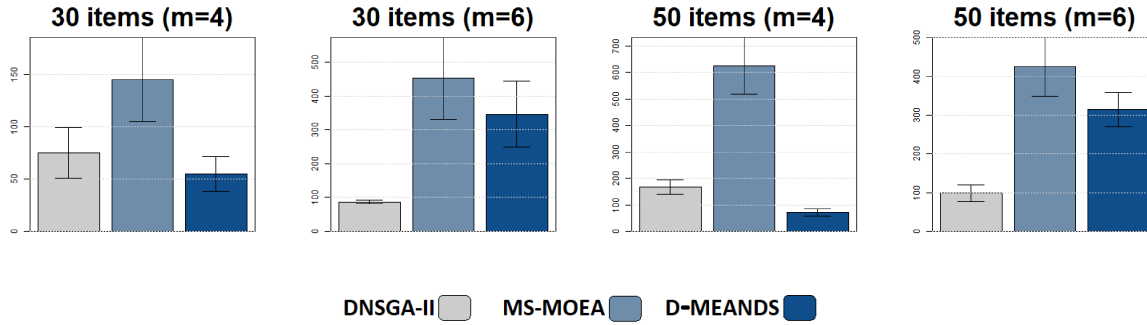


Figura 8 – Tempo de execução do DNSGA-II, MS-MOEA e D-MEANDS em cenários com uma mudança de ambiente.

4.3.2 Cenários com Duas Mudanças de Ambiente

Nesses experimentos, as duas mudanças de ambiente ($EC = 2$) acontecem em $1/3$ e $2/3$ do número de gerações. Em outras palavras, para o DNSGA-II e o MS-MOEA, elas ocorrem nas gerações 33 e 66, enquanto, para o D-MEANDS, ocorrem nas gerações 3.333 e 6.666.

A Tabela 7 apresenta os resultados obtidos pelo teste de hipótese t ao comparar o D-MEANDS com os demais MOEAs. Analisando a tabela, observa-se que o D-MEANDS tende a ser significativamente melhor que o DNSGA-II na instância com 6 objetivos. O D-MEANDS apresenta desempenho similar ao MS-MOEA na maioria das instâncias. No entanto, em instâncias com 50 itens e 4 objetivos, o D-MEANDS mostra-se inferior ao MS-MOEA. O DNSGA-II supera o D-MEANDS na métrica HVR nas instâncias com 4 objetivos e 50 itens. Nota-se uma melhora do D-MEANDS na instância de 50 itens com 6 objetivos (comparado à instância de 4 objetivos e 50 itens), sendo significativamente melhor que o DNSGA-II e equiparando-se ao MS-MOEA. Assim, percebe-se que o aumento

do número de objetivos torna o D-MEANDS mais competitivo nas métricas HVR e $\overline{IGD}$, embora apresente um aumento considerável no tempo de execução. Também foi possível observar uma reação positiva do DMSGa-II com o aumento do número de mudanças de ambiente (EC). Informações adicionais sobre essa análise comparativa são apresentadas no Apêndice B (Seção B.2.2).

Tabela 7 – Teste de hipótese comparando D-MEANDS aos demais algoritmos em cenários com duas mudança de ambiente (EC=2).

Algoritmos	Items	Objetivos			
		4		6	
		HVR	$\overline{IGD}$	HVR	$\overline{IGD}$
DMSGa-II	30	=	=	>	<
	50	<	=	>	<
MS-MOEA	30	=	=	=	=
	50	<	>	=	=

Ao comparar os algoritmos em relação ao seu tempo de execução (Figura 9), observa-se comportamentos semelhantes aos experimentos com $EC = 1$. Isto é, o DMSGa-II continua sendo o algoritmo mais rápido, enquanto o MS-MOEA apresenta os piores tempos.

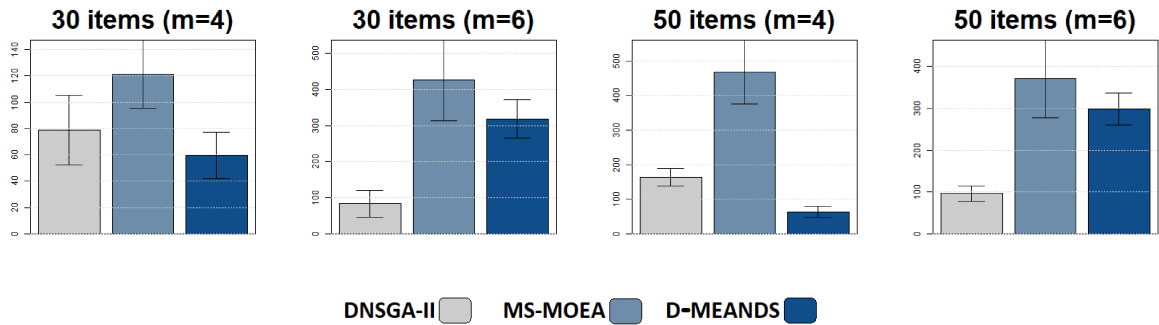


Figura 9 – Tempo de execução do DMSGa-II, MS-MOEA e D-MEANDS em cenários com duas mudança de ambiente.

Considerando que o D-MEANDS alcançou um desempenho próximo ao MS-MOEA, em relação as métricas HVR e $\overline{IGD}$, e apresenta um tempo de execução menor, pode-se concluir que, de um modo geral, ele entrega um melhor balanceamento (custo-benefício) entre desempenho multiobjetivo e custo computacional.

4.3.3 Cenários com Cinco e Dez Mudanças de Ambiente

Posteriormente, o desempenho dos três MOEAs também foi investigado em experimentos que envolveram um número maior de mudanças de ambiente ($EC=5$ e $EC=10$) ao longo da execução do AG. Nesses experimentos, apenas as instâncias com 4 objetivos foram utilizadas devido ao custo computacional de cada experimento.

A Tabela 8 mostra a comparação entre o D-MEANDS e os demais algoritmos evolutivos, considerando o teste de hipótese t . Como pode ser observado, o D-MEANDS apresentou desempenho significativamente melhor em todas as instâncias, considerando ambas as métricas. Isso é um indício que a estratégia proposta pelo algoritmo favorece sua adaptação em cenários com maior dinamicidade, propiciando uma melhor convergência em um período menor de evolução. Uma análise comparativa mais detalhada é apresentada no Apêndice B (Seção B.2.3).

Tabela 8 – Teste de hipótese comparando D-MEANDS aos demais algoritmos em cenários com cinco e dez mudanças de ambiente (EC= 5 e 10).

Algoritmos	Items	Objetivos (m=4)			
		EC=5		EC=10	
		HVR	$\overline{IGD}$	HVR	$\overline{IGD}$
DNSGA-II	30	>	<	>	<
	50	>	<	>	<
MS-MOEA	30	>	<	>	<
	50	>	<	>	<

Por fim, apresentamos a análise do tempo de execução na Figura 10. É possível observar que, nos experimentos com maior número de mudanças de ambiente (EC=5 e EC=10), os tempos demandados pelo DNSGA-II e pelo D-MEANDS são similares e menores que o tempo de execução do MS-MOEA. Entretanto, cabe ressaltar que, nesses testes, apenas instâncias com 4 objetivos foram utilizadas.

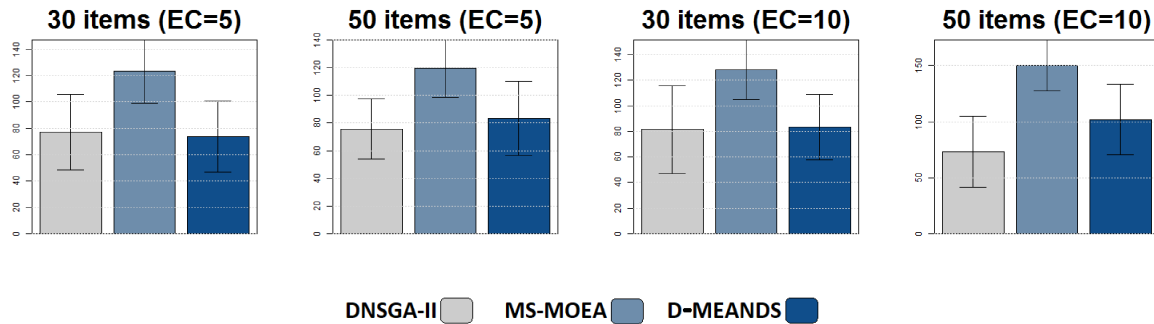


Figura 10 – Tempo de execução do DNSGA-II, MS-MOEA e D-MEANDS em cenários com cinco e dez mudanças de ambiente.

Os resultados e análises das seções 4.3.1, 4.3.2 e 4.3.3 foram publicados no artigo (LAFETÁ; OLIVEIRA, 2020B), que também apresenta a proposta D-MEANDS.

4.3.4 Cenários com Muitas Mudanças de Ambiente

Um dos grandes impeditivos para a realização de experimentos com um maior número de objetivos em cenários mais dinâmicos ($EC \geq 10$) deve-se ao tempo elevado necessário para calcular as métricas HVR e $\overline{IGD}$, que dependem do cálculo do conjunto de Pareto

aproximado (pois são métricas paramétricas). Por isso, nesta seção são apresentados resultados de experimentos usando métricas não paramétricas.

Nesses experimentos, o desempenho dos três algoritmos, DNSGA-II*, MS-MOEA e D-MEANDS, é avaliado em ambientes mais dinâmicos, ou seja, com um número maior de mudanças (EC=10, 15 e 20). Além disso, um algoritmo mais recente na literatura foi adicionado à investigação, conhecido como *MOEA/D based on Differential Evolution and Dynamic Diversity Introduction Strategy* (DDIS-MOEA/D-DE) (LIU et al., 2020).

Na seção anterior, as instâncias do DMKP foram limitadas a apenas 4 objetivos quando foram empregadas 10 mudanças de ambiente. Essa limitação se deu, principalmente, devido ao uso das métricas paramétricas HVR e $\overline{IGD}$, que aumentaram muito o custo computacional. Por outro lado, a adoção de métricas não paramétricas, como Densidade (AD) e Hipervolume (HV), permitem analisar o comportamento dos MOEAs investigados em cenários mais complexos, usando instâncias com até 8 objetivos e 20 mudanças de ambiente.

Nesse experimento, foram utilizadas 27 instâncias do DMKP, divididas pelo número de itens, objetivos e mudanças de ambiente (EC). Nossas análises consideraram instâncias de 30, 50 e 100 itens; 4, 6 e 8 objetivos ; e 10, 15 ou 20 mudanças nos ambientes. Os algoritmos DDIS-MOEA/D-DE, DNSGA-II* e MS-MOEA foram configurados com os seguintes parâmetros: 500 gerações, 100% cruzamento, 10% mutação, e população de 100 indivíduos. Para o DNSGA-II*, 20% da população foi substituída em cada mudança de ambiente. No MS-MOEA, a cada mudança de ambiente, 20% da população é composta por indivíduos do arquivo que sofreu mutação e 80% são indivíduos aleatórios. O DDIS-MOEA/D-DE usa 10 vizinhos para formar as subpopulações. Como o D-MEANDS realiza apenas um cruzamento por geração, seus parâmetros foram escolhidos para realizar o mesmo número de cruzamentos que os demais algoritmos: 50.000 gerações, 10% de mutação e população de 1000 indivíduos. Cada algoritmo foi executado 100 vezes para cada instância do problema e os gráficos do Apêndice B (Seção B.2.4) mostram a média e o desvio padrão dessas 100 execuções.

O teste de hipótese t foi empregado para consolidar os resultados apresentados nos gráficos e avaliar a significância do desempenho do D-MEANDS em relação aos outros MOEAs. Os resultados obtidos em cada um dos cenários de mudança de ambiente analisados (EC = 10, 15 e 20) são apresentados nas Tabelas 9, 10 e 11, respectivamente. Os testes de hipóteses foram feitos com 99% de confiança. Cada linha das tabelas representa a comparação do D-MEANDS com outro algoritmo, e a coluna indica a métrica analisada. As células verdes mostram que o D-MEANDS foi significativamente melhor, e as células vermelhas mostram quando ele foi significativamente pior. As células brancas, indicam que não houve diferença significativa. Como pode ser observado, as tabelas apresentam resultados muito parecidos para os três cenários de mudança de ambiente (EC = 10, 15 e 20). No geral, o teste de hipóteses confirma que o D-MEANDS é significativamente supe-

rior na métrica AD em quase todos os cenários. Na métrica HV, o D-MEANDS se destaca mais que os demais algoritmos. O MS-MOEA se torna mais competitivo com o aumento do número de mudanças de ambiente (EC); no entanto, o aumento do número de objetivos favorece o D-MEANDS. Em 18 instâncias, o D-MEANDS é significativamente superior ao MS-MOEA na métrica HV e significativamente inferior em apenas 6 instâncias.

Tabela 9 – Teste de hipótese comparando D-MEANDS aos demais algoritmos em cenários com dez mudanças de ambiente (EC=10).

Objetivos	Algoritmos	DDIS-MOEA/D-DE			DNSGA-II*			MS-MOEA		
	Métricas/Itens	30	50	100	30	50	100	30	50	100
4	HV	>	>	>	>	<	<	>	=	=
	AD	<	<	<	<	<	<	<	<	<
6	HV	>	>	>	>	>	<	>	>	>
	AD	<	<	<	<	<	<	<	<	<
8	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	<	<

Tabela 10 – Teste de hipótese comparando D-MEANDS aos demais algoritmos em cenários com quinze mudanças de ambiente (EC=15).

Objetivos	Algoritmos	DDIS-MOEA/D-DE			DNSGA-II*			MS-MOEA		
	Métricas/Itens	30	50	100	30	50	100	30	50	100
4	HV	>	>	>	>	<	<	>	=	<
	AD	<	<	<	<	<	<	<	<	<
6	HV	>	>	>	>	>	<	>	>	>
	AD	<	<	<	<	<	<	<	<	<
8	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	<	<

Tabela 11 – Teste de hipótese comparando D-MEANDS aos demais algoritmos em cenários com vinte mudanças de ambiente (EC=20).

Objetivos	Algoritmos	DDIS-MOEA/D-DE			DNSGA-II*			MS-MOEA		
	Métricas/Itens	30	50	100	30	50	100	30	50	100
4	HV	>	>	>	>	<	<	<	<	<
	AD	<	<	<	<	<	<	<	<	<
6	HV	>	>	>	>	>	<	<	>	<
	AD	<	<	<	<	<	<	<	<	<
8	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	<	<

Os tempos médios de execução dos algoritmos em cenários com 4, 6 e 8 objetivos são mostrados nas Figuras 11, 12 e 13. Claramente, o tempo de execução do DNSGA-II* é muito menor do que o dos outros algoritmos na maioria das instâncias. Essa observação torna-se mais evidente com o aumento da complexidade dos cenários investigados. O D-MEANDS apresenta um custo computacional próximo ao do DNSGA-II* nas instâncias

de 4 objetivos, mas, à medida que o número de objetivos aumenta, seu tempo de execução cresce consideravelmente em relação ao DNSGA-II*. Em relação aos demais MOEAs, o D-MEANDS se mostra mais rápido em todas as instâncias de 4 e 6 objetivos; no entanto, seu tempo de execução aproxima-se do do MS-MOEA em instâncias de 8 objetivos. Embora, o DDIS-MOEA/D-DE tenha o pior tempo de execução nas instâncias com 4 objetivos, a medida que o número de objetivos aumenta, seu tempo de execução torna-se melhor que o do MS-MOEA e do D-MEANDS.

O D-MEANDS encontra os melhores resultados nas métricas AD e HV na maioria dos casos, sendo que seu desempenho se destaca com o aumento do número de objetivos. No entanto, seu tempo de execução cresce de forma exponencial à medida que o número de objetivos aumenta. Por outro lado, o DNSGA-II* mostrou-se bastante competitivo em algumas instâncias, exigindo um menor tempo de execução mesmo com o aumento do número de objetivos.

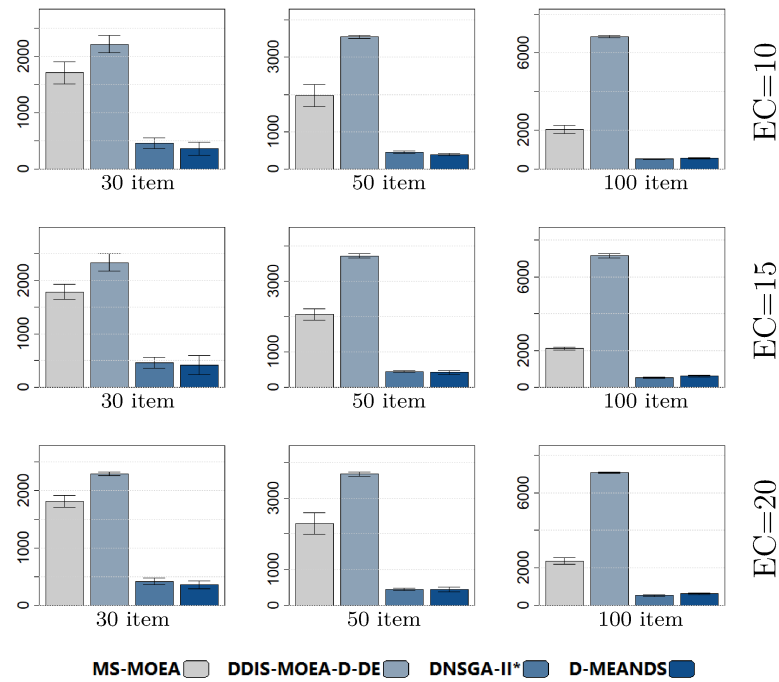


Figura 11 – Tempo de execução do D-MEANDS em relação aos demais algoritmos em instâncias com 4 objetivos e muitas mudanças de ambiente ($EC = 10, 15$ e 20).

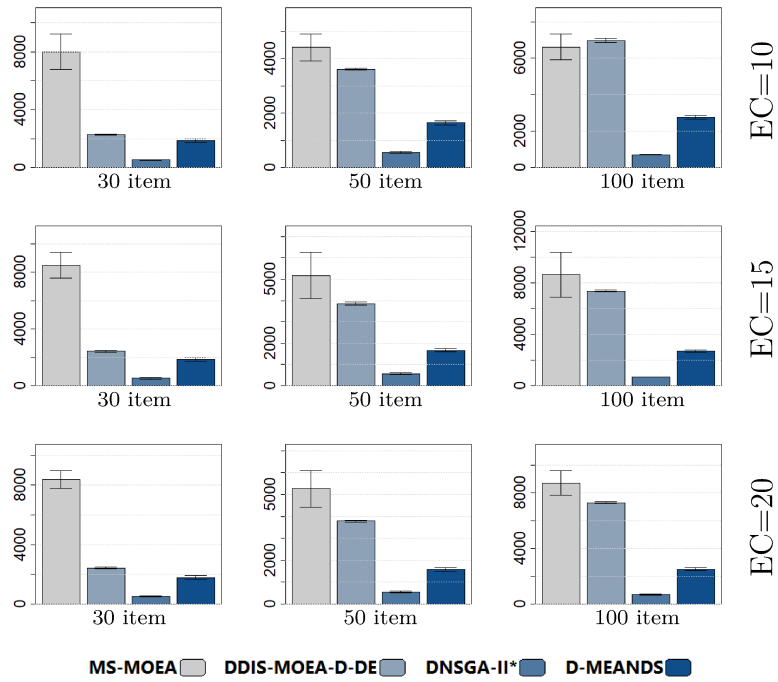


Figura 12 – Tempo de execução do D-MEANDS em relação aos demais algoritmos em instâncias com 6 objetivos e muitas mudanças de ambiente ($EC = 10, 15$ e 20).

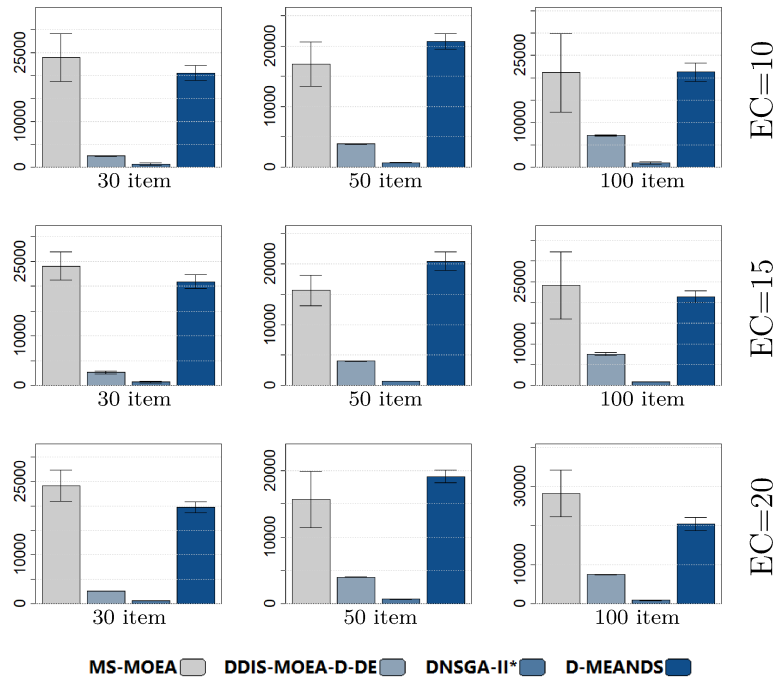


Figura 13 – Tempo de execução do D-MEANDS em relação aos demais algoritmos em instâncias com 8 objetivos e muitas mudanças de ambiente ($EC = 10, 15$ e 20).

4.4 Fase 3: Análise de Desempenho do Algoritmo D-MEANDS-MD

Ao investigar as possíveis causas para a queda de desempenho do D-MEANDS em função do aumento no número de mudanças no ambiente aumenta, foi possível observar que a convergência deste MOEA é prejudicada, principalmente pelo severo decaimento na variabilidade das soluções não dominadas ao longo das gerações. Isso ocorre porque o D-MEANDS, assim como sua versão para problemas discretos (MEANDS), descarta todas as informações de soluções dominadas. Portanto, fica clara a necessidade de um novo mecanismo para injetar alguma aleatoriedade, o que resultou na proposta de um novo MOEA denominado D-MEANDS-MD, explicado na Seção 3.4.

Nesta seção, o D-MEANDS-MD é comparado com os melhores MOEAs investigados na Seção 4.3.4, a saber: DNSGA-II* e D-MEANDS, a fim de avaliar o impacto da modificação implementada. O DNSGA-II* foi selecionado por ser o algoritmo mais rápido, enquanto o D-MEANDS por retornar o melhor desempenho em relação as métricas AD e HV. Além disso, foi investigada uma variação do DNSGA-II*, chamada DNSGA-II⁺, na qual o mesmo mecanismo de diversidade proposto para o D-MEANDS-MD foi incorporado, ou seja, 10% dos cruzamentos do DNSGA-II⁺ usam um pai aleatório.

A comparação entre os algoritmos foi realizada com base nas mesmas 27 instâncias do problema utilizadas no experimento anterior (Seção 4.3.4), as quais foram obtidas a partir da combinação de diferentes números de itens (30, 50 e 100), objetivos (4, 6 e 8) e mudanças de ambiente (10, 15 e 20). Nesse experimento, os algoritmos DNSGA-II* e DNSGA-II⁺ foram executados com os seguintes parâmetros: 500 gerações, 100% de cruzamento, 10% de mutação, população de 100 indivíduos, com 20% da população sendo substituída a cada mudança de ambiente. Como o D-MEANDS e o D-MEANDS-MD realizam apenas um cruzamento por geração, seus parâmetros foram escolhidos para realizar o mesmo número de cruzamentos do DNSGA-II*: 50.000 gerações, 10% de mutação e 1.000 indivíduos na população. Cada algoritmo foi executado 100 vezes para cada instância, sendo que os gráficos discutidos no Apêndice B (Seção B.3) mostram a média e o desvio padrão dessas 100 execuções, considerando cada métrica de desempenho e tempo de execução.

O teste de hipótese t foi utilizado para avaliar a significância dos resultados do D-MEANDS-MD em relação aos demais algoritmos, considerando 99% de confiança. As Tabelas 12, 13 e 14 apresentam, respectivamente, os resultados desse teste, considerando as métricas densidade (AD) e hipervolume (HV), para as instâncias com EC=10, EC=15 e EC=20, sendo que cada linha representa a comparação do D-MEANDS-MD com outro algoritmo, e a coluna indica as métricas. As células verdes indicam que o D-MEANDS-MD foi significativamente melhor, enquanto as células vermelhas indicam que foi significativamente pior. Nas células brancas, não houve diferença significativa. É possível observar que nossa proposta supera significativamente todos os outros algoritmos nas 27 instâncias

investigadas.

Tabela 12 – Teste de hipótese comparando D-MEANDS-MD aos demais algoritmos em instâncias com dez mudanças de ambiente (EC=10).

Objetivos	Metricas\Itens	DNSGA-II*			DNSGA-II ⁺			D-MEANDS		
		30	50	100	30	50	100	30	50	100
4	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	<	<
6	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	<	<
8	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	<	<

Tabela 13 – Teste de hipótese comparando D-MEANDS-MD aos demais algoritmos em instâncias com quinze mudanças de ambiente (EC=15).

Objetivos	Metricas\Itens	DNSGA-II*			DNSGA-II ⁺			D-MEANDS		
		30	50	100	30	50	100	30	50	100
4	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	<	<
6	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	<	<
8	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	<	<

Tabela 14 – Teste de hipótese comparando D-MEANDS-MD aos demais algoritmos em instâncias com vinte mudanças de ambiente (EC=20).

Objetivos	Metricas\Itens	DNSGA-II*			DNSGA-II ⁺			D-MEANDS		
		30	50	100	30	50	100	30	50	100
4	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	<	<
6	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	<	<
8	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	<	<

As Figuras 14, 15 e 16 mostram o tempo de execução de cada MOEA investigado. O D-MEANDS e D-MEANDS-MD foram mais rápidos que o DNSGA-II* e o DNSGA-II⁺ em algumas instâncias de 4 objetivos. Em contrapartida, nas formulações com 6 e 8 objetivos, o DNSGA-II* e o DNSGA-II⁺ mostraram-se claramente mais rápidos. Entre os algoritmos baseados em subpopulações, o tempo de execução do D-MEANDS-MD foi próximo ao do D-MEANDS, embora ligeiramente mais elevado. Portanto, embora o D-MEANDS-MD tenha apresentado o melhor desempenho em termos de métricas multiobjetivo, ele é tão dispendioso quanto seu precursor D-MEANDS, e esse custo computacional aumenta com o crescimento do número de objetivos.

A proposta do D-MEANDS-MD, assim como parte dos resultados e análises descritos nesta seção e no Apêndice B (Seção B.3) também são apresentados no artigo publicado no *Knowledge Engineering Review Journal* (LAFETÁ; MARTINS; OLIVEIRA, 2024).

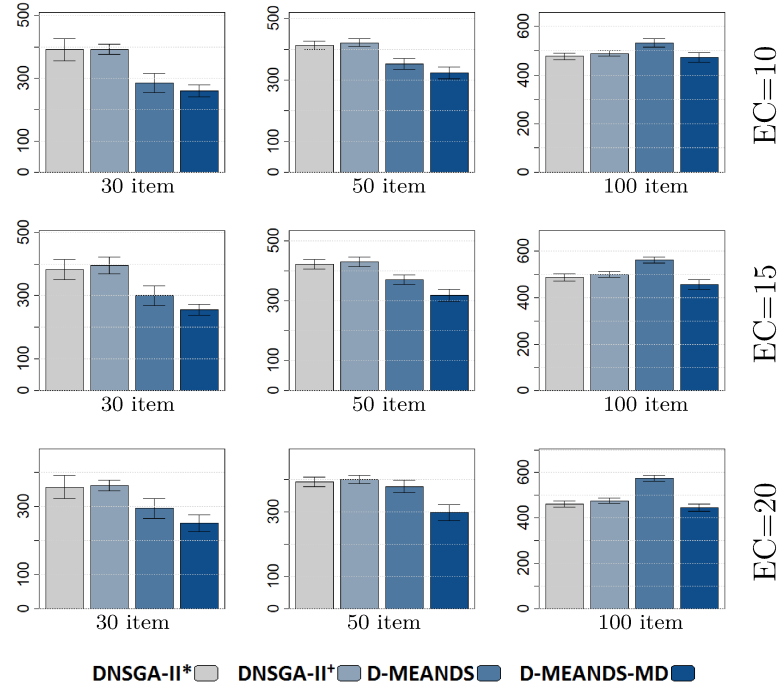


Figura 14 – Tempo de execução do D-MEANDS-MD em relação aos demais algoritmos em instâncias com 4 objetivos e muitas mudanças de ambiente ($EC = 10, 15$ e 20).

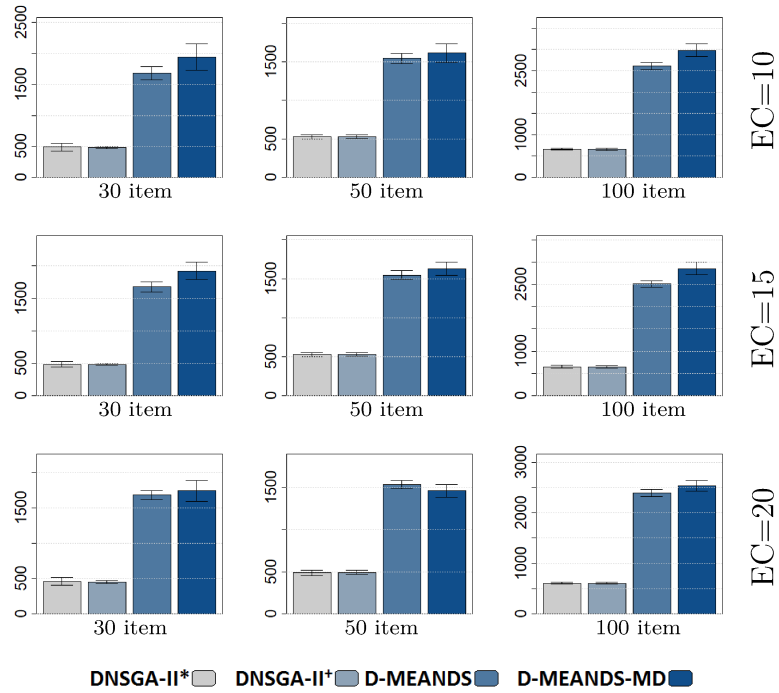


Figura 15 – Tempo de execução do D-MEANDS-MD em relação aos demais algoritmos em instâncias com 6 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).

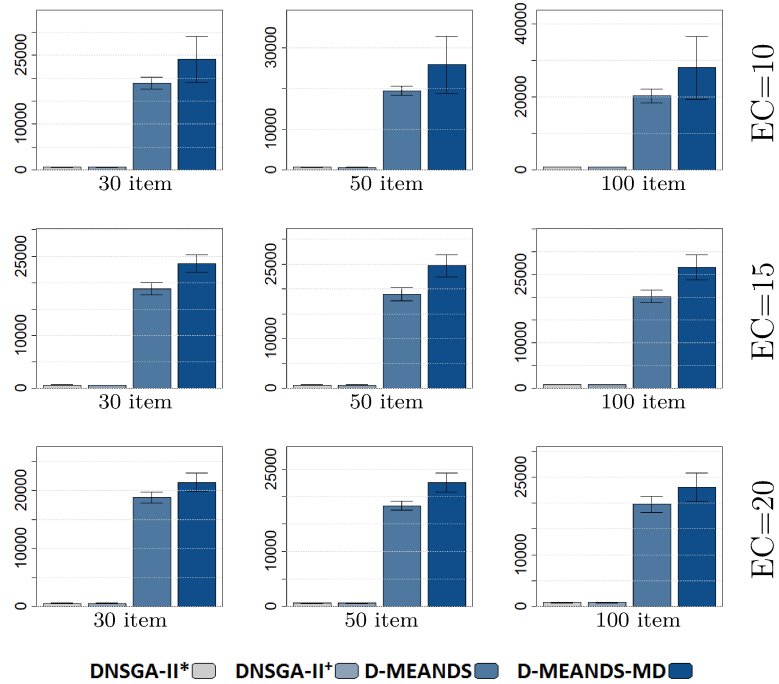


Figura 16 – Tempo de execução do D-MEANDS-MD em relação aos demais algoritmos em instâncias com 8 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).

4.5 Fase 4: Análise de Desempenho do Algoritmo D-MEANDS-II

O D-MEANDS-MD apresentou excelentes desempenho para as métricas HV e AD, no entanto, seu tempo de execução cresce exponencialmente à medida que aumenta o número de objetivos do problema. O D-MEANDS-II, que é uma variação do MEANDS-II (algoritmo que trata problemas estáticos), foi proposto com o objetivo de reduzir o tempo de execução do D-MEANDS e do D-MEANDS-MD. Para validar sua eficiência, esse novo algoritmo é comparado com as abordagens que alcançaram os melhores desempenhos nas fases anteriores, tanto em relação as métricas multiobjetivo (MS-MOEa na fase 1 e D-MEANDS-MD na fase 3), quanto ao tempo de execução (DnSGA-II*). O D-MEANDS (fase 2) não é apresentado, uma vez que seu desempenho é superado pelo seu sucessor D-MEANDS-MD. Neste experimento, também foram utilizadas as mesmas instâncias do DMKP usadas nas fases anteriores (cenários com 30,50 e 100 itens; 4, 6 e 8 objetivos; e 10, 15 e 20 mudanças de ambiente), bem como as duas métricas não-paramétricas (AD e HV) empregadas para avaliar o desempenho multiobjetivo dos algoritmos.

Foram utilizados os seguintes parâmetros para o DnSGA-II* e o MS-MOEa: 500 gerações, 100% de cruzamento, 10% de mutação, população de 100 indivíduos, e 20% da população foi substituída em cada mudança de ambiente. Como o D-MEANDS-MD e o D-MEANDS-II realizam apenas um cruzamento por geração, seus parâmetros foram definidos de modo a manter a mesma quantidade de cruzamentos do DnSGA-II* e do MS-MOEa, que são: 50.000 gerações, 10% de mutação e 1.000 indivíduos na população. Cada algoritmo foi executado 100 vezes para cada instância investigada. Os resultados experimentais, com as médias e desvio padrão das métricas de desempenho obtidas nessas execuções, são mostrados e discutidos detalhadamente no Apêndice B (Seção B.4).

A partir desses resultados, o teste de hipótese t foi utilizado para avaliar a significância estatística dos valores alcançados pelo D-MEANDS-II em relação aos demais MOEAs, considerando uma confiança de 99%. As Tabelas 15, 16 e 17 mostram os resultados desse teste para as diferentes instâncias do problema analisadas em cada cenário de mudança (EC=10, 15 e 20), respectivamente. Cada linha da tabela representa a comparação do D-MEANDS-II com outro algoritmo, e a coluna indica a métrica analisada. As células verdes mostram que o D-MEANDS-II foi significativamente melhor, as vermelhas mostram que ele foi significativamente pior, enquanto as brancas indicam que não houve diferença significativa.

Ao analisar as tabelas, percebe-se um comportamento similar para os 3 cenários de mudança de ambiente. No geral, o D-MEANDS-II supera significativamente o MS-MOEa e o DnSGA-II* em todas as 27 instâncias.

Comparado ao D-MEANDS-MD, o D-MEANDS-II tem um desempenho inferior na métrica AD em 21 instâncias, sendo superior apenas nos cenários de 30 itens com 6 e 8

objetivos. Em contrapartida, o novo algoritmo supera o D-MEANDS-MD na métrica HV em 21 instâncias, não havendo diferença significativa em outras cinco e só é superado na formulação com 30 itens, 8 objetivos e EC=15. A superioridade na métrica AD, indica que o D-MEANDS-MD encontra um conjunto de soluções melhor distribuído pelo espaço objetivo. Por outro lado, o melhor desempenho do D-MEANDS-II na métrica HV, indica que ele é capaz de encontrar um conjunto de soluções mais equilibrado em termos de convergência e diversidade.

Tabela 15 – Teste de hipótese comparando D-MEANDS-II aos demais algoritmos em instâncias com dez mudanças de ambiente (EC=10).

Objetivos	Metricas/Itens	MS-MOEA			DNSGA-II*			D-MEANDS-MD		
		30	50	100	30	50	100	30	50	100
4	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	>	>	>
6	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	>	>
8	HV	>	>	>	>	>	>	=	=	>
	AD	<	<	<	<	<	<	<	>	>

Tabela 16 – Teste de hipótese comparando D-MEANDS-II aos demais algoritmos em instâncias com quinze mudanças de ambiente (EC=15).

Objetivos	Metricas/Itens	MS-MOEA			DNSGA-II*			D-MEANDS-MD		
		30	50	100	30	50	100	30	50	100
4	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	>	>	>
6	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	>	>
8	HV	>	>	>	>	>	>	<	=	>
	AD	<	<	<	<	<	<	<	>	>

Tabela 17 – Teste de hipótese comparando D-MEANDS-II aos demais algoritmos em instâncias com vinte mudanças de ambiente (EC=20).

Objetivos	Metricas/Itens	MS-MOEA			DNSGA-II*			D-MEANDS-MD		
		30	50	100	30	50	100	30	50	100
4	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	>	>	>
6	HV	>	>	>	>	>	>	>	>	>
	AD	<	<	<	<	<	<	<	>	>
8	HV	>	>	>	>	>	>	=	=	>
	AD	<	<	<	<	<	<	<	>	>

As Figuras 17, 18 e 19 mostram o tempo de execução dos algoritmos em cada uma das 27 instâncias do DMKP. Nas instâncias de 4 objetivos, o D-MEANDS-II foi o algoritmo

mais rápido, seguido pelo D-MEANDS-MD e D-NSGA-II*, enquanto o MS-MOEA demanda um tempo computacional elevado. Já nas instâncias de 6 objetivos, o D-NSGA-II* supera os demais algoritmos, com o D-MEANDS-II apresentando resultados bem próximos. O D-MEANDS-MD apresenta um crescimento considerável no tempo de execução, demandando aproximadamente o dobro do tempo do D-MEANDS-II, mas ainda bem abaixo do MS-MOEA. Nas instâncias de 8 objetivos, o D-MEANDS-MD e o MS-MOEA apresentam os piores tempos de execução. O D-NSGA-II* continua sendo o algoritmo mais rápido, mas com o D-MEANDS-II também apresentando um bom desempenho computacional. Considerando todas as instâncias investigadas, pode-se observar que o aumento do tempo de execução do D-MEANDS-II segue um comportamento próximo ao D-NSGA-II* e claramente superior aos demais algoritmos avaliados.

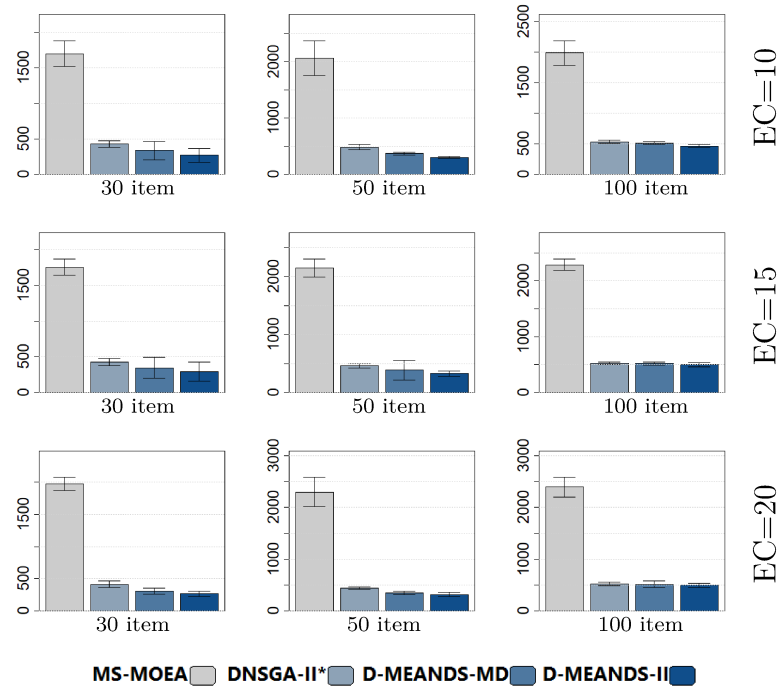


Figura 17 – Tempo de execução do D-MEANDS-II em relação aos demais algoritmos em instâncias com 4 objetivos e muitas mudanças de ambiente (EC = 10, 15 e 20).

No geral, o D-MEANDS-II apresenta um desempenho competitivo em relação ao D-MEANDS-MD, sendo superior no hipervolume, mas perdendo na densidade. Por outro lado, o D-MEANDS-II é significativamente mais eficiente em relação ao tempo de execução, principalmente com o aumento no número de objetivos. Em outras palavras, o D-MEANDS-II consegue ser competitiva com o D-MEANDS-MD em termos de convergência e diversidade, com um custo computacional (tempo de execução) muito menor.

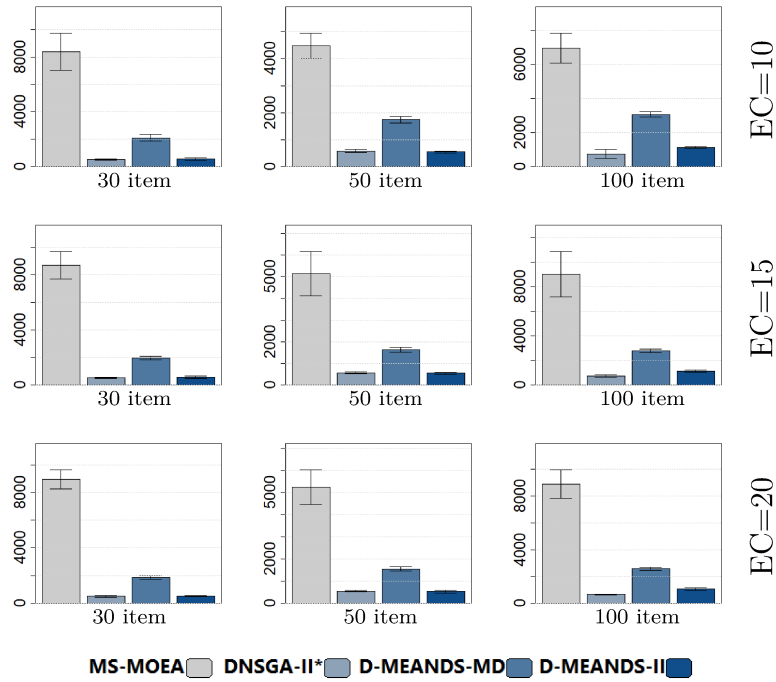


Figura 18 – Tempo de execução do D-MEANDS-II em relação aos demais algoritmos em instâncias com 6 objetivos e muitas mudanças de ambiente ($EC = 10, 15$ e 20).

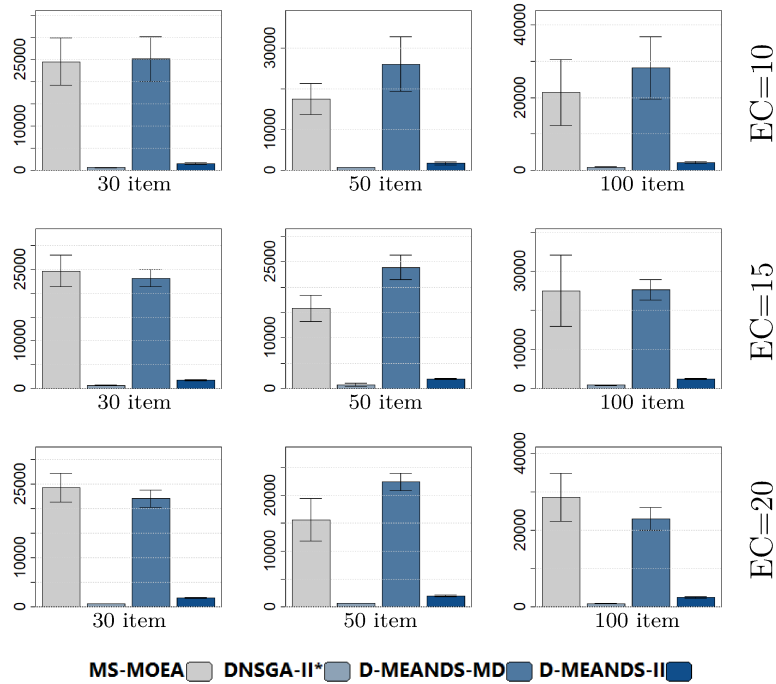


Figura 19 – Tempo de execução do D-MEANDS-II em relação aos demais algoritmos em instâncias com 8 objetivos e muitas mudanças de ambiente ($EC = 10, 15$ e 20).

4.6 Fase 5: Análise de Desempenho do Algoritmo D-MEANDS-III

O D-MEANDS-III é uma evolução do algoritmo anterior (D-MEANDS-II), que visa aprimorar o mecanismo de busca do modelo multi-populacional a partir de uma nova estratégia de seleção de pais e do ajuste do método de mudança de ambiente. A fim de validar a eficácia dessas modificações, o desempenho do novo algoritmo é comparado com as abordagens investigadas na fase 4 (MS-MOEA, DNSGA-II*, D-MEANDS-MD e D-MEANDS-II). Esse experimento foi conduzido a partir das mesmas instâncias (27 no total) e métricas de desempenho (AD e HV). As configurações dos MOEAs também foram mantidas inalteradas, com o D-MEANDS-III utilizando os mesmos parâmetros do seu antecessor (D-MEANDS-II).

Como nas fases anteriores, cada algoritmo foi executado 100 vezes para cada instância do problema. Os resultados experimentais dessas execuções são apresentados de forma consolidada (média e desvio padrão) e analisados no Apêndice B (Seção B.5).

Considerando que os melhores resultados para as métricas investigadas (AD e HV) foram obtidos pelo D-MEANDS-MD e D-MEANDS-II, empregou-se um teste de hipótese t , com 99% de confiança, para avaliar a significância do desempenho do D-MEANDS-III em relação a esses algoritmos. Os resultados são apresentados nas Tabelas 18, 19 e 20, devidamente agrupados pela quantidade de mudanças realizadas no ambiente (EC=10, 15 e 20). Como pode ser observado, em relação à métrica AD, o D-MEANDS-III foi significativamente pior que o D-MEANDS-MD em 13 instâncias, conseguindo superá-lo em apenas duas instâncias. Comparado ao D-MEANDS-II, o novo algoritmo foi significativamente melhor em 15 instâncias e perdeu em 4 instâncias. Nas 3 tabelas, nota-se que o desempenho do algoritmo nessa métrica (em relação aos demais) é afetado com o aumento no número de objetivos. Analisando a métrica HV, o D-MEANDS-III mostra-se significativamente superior aos seus antecessores, sendo melhor que o D-MEANDS-MD em 26 instâncias e que o D-MEANDS-II em 25 instâncias. Nessa métrica, o D-MEANDS-III não foi superado por nenhum dos dois algoritmos.

Tabela 18 – Teste de hipótese comparando o desempenho do D-MEANDS-III em relação aos seus antecessores, considerando instâncias com EC=10.

Objetivos	Metricas/Itens	D-MEANDS-MD			D-MEANDS-II		
		30	50	100	30	50	100
4	HV	>	>	>	=	>	>
	AD	=	=	=	=	<	<
6	HV	>	>	>	>	>	>
	AD	=	=	>	=	<	=
8	HV	=	>	>	=	>	>
	AD	=	>	>	>	<	=

Tabela 19 – Teste de hipótese comparando o desempenho do D-MEANDS-III em relação aos seus antecessores, considerando instâncias com EC=15.

Objetivos	Metricas/Itens	D-MEANDS-MD			D-MEANDS-II		
		30	50	100	30	50	100
4	HV	>	>	>	>	>	>
	AD	=	=	>	<	<	<
6	HV	>	>	>	>	>	>
	AD	<	>	>	=	<	=
8	HV	>	>	>	>	>	>
	AD	=	>	>	>	<	=

Tabela 20 – Teste de hipótese comparando o desempenho do D-MEANDS-III em relação aos seus antecessores, considerando instâncias com EC=20.

Objetivos	Metricas/Itens	D-MEANDS-MD			D-MEANDS-II		
		30	50	100	30	50	100
4	HV	>	>	>	>	>	>
	AD	=	=	>	<	<	<
6	HV	>	>	>	>	>	>
	AD	<	>	>	>	<	<
8	HV	>	>	>	>	>	>
	AD	=	>	>	>	<	=

Nas Figuras 20, 21 e 22 são mostrados os tempos médios de execução dos algoritmos em cada uma das 27 instâncias do DMKP. Como pode ser observado, o D-MEANDS-III acompanha o mesmo comportamento que o D-MEANDS-II no tempo de execução. Nas instâncias de 4 objetivos (Figura 20), esses algoritmos apresentam os melhores tempos de execução. Nas instâncias de 6 (Figura 21) e 8 objetivos (Figura 22), o DNSGA-II* reage e supera todos os demais algoritmos. Entretanto, tanto o D-MEANDS-II quanto o D-MEANDS-III conseguem manter um tempo de execução próximo ao do DNSGA-II*. Também é possível notar que, à medida que o número de mudanças de ambiente aumenta, o tempo de execução do D-MEANDS-III melhora em relação ao do D-MEANDS-II.

Resumidamente, o D-MEANDS-III incorpora novos mecanismos no processo de busca, introduzindo um novo modelo de mudança de ambiente que explora informações da característica multi-populacional presente desde seu predecessor (MEANDS). Esses novos mecanismos melhoraram a convergência e a diversidade da busca em comparação ao D-MEANDS-II, mantendo um tempo de execução similar.

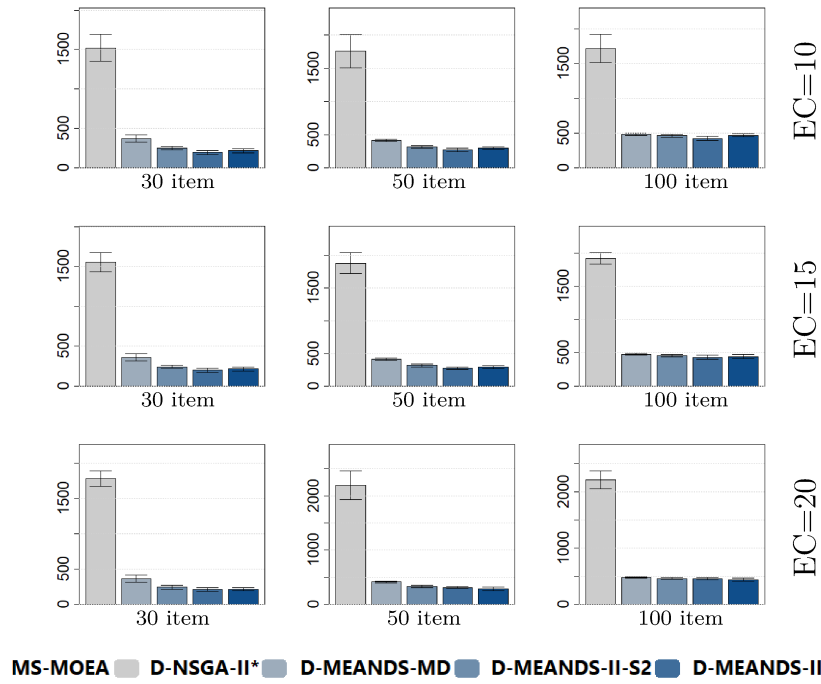


Figura 20 – Tempo de execução do D-MEANDS-III em relação aos demais algoritmos em instâncias com 4 objetivos e muitas mudanças de ambiente ($EC = 10, 15$ e 20).

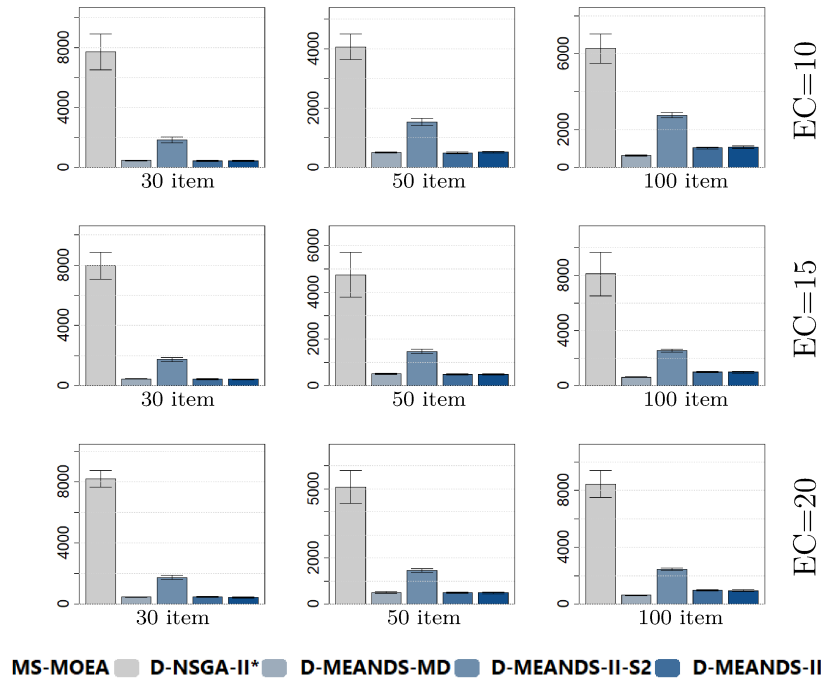


Figura 21 – Tempo de execução do D-MEANDS-III em relação aos demais algoritmos em instâncias com 6 objetivos e muitas mudanças de ambiente ($EC = 10, 15$ e 20).

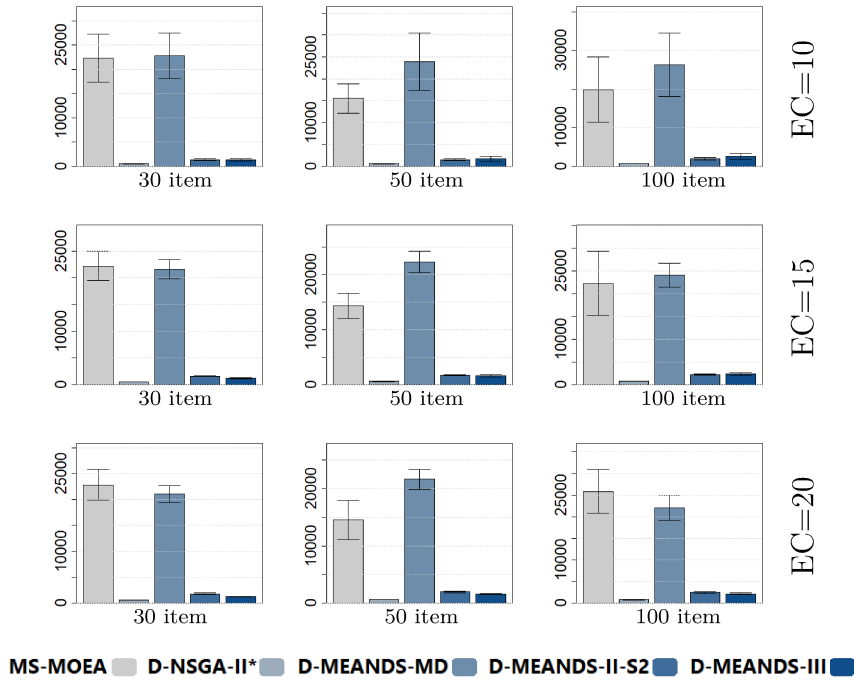


Figura 22 – Tempo de execução do D-MEANDS-III em relação aos demais algoritmos em instâncias com 8 objetivos e muitas mudanças de ambiente ($EC = 10, 15$ e 20).

4.7 Análises Complementares

4.7.1 Comparando os Conjuntos de Soluções retornados pelos MOEAs

Nos experimentos relatados nas seções anteriores, os MOEAs foram avaliados em relação às métricas não-paramétricas AD e HV, que fornecem valores absolutos para a análise comparativa entre vários algoritmos. Essas métricas traduzem algumas características da aproximação do conjunto Pareto encontradas por um algoritmo específico, como diversidade e convergência para o conjunto Pareto ótimo, em valores numéricos que possibilitam essas comparações múltiplas. No entanto, também é comum, em investigações multiobjetivo, realizar uma comparação direta de dois algoritmos quanto à qualidade do conjunto de soluções encontradas por cada um. Nesta seção, a métrica *Coverage of two Sets* (CS) (descrita na Seção 2) é utilizada para realizar uma comparação direta entre os algoritmos: MS-MOEA, D-NSGA-II*, D-MEANDS-MD, D-MEANDS-II e D-MEANDS-III.

Este experimento adicional foi realizado utilizando os cenários com o maior número de alterações avaliadas, ou seja, considerando apenas instâncias com $EC=20$. Cada resultado representa a média de 100 comparações entre o conjunto de Pareto de dois algoritmos (X e Y). As Tabelas apresentadas neste subcapítulo mostram os valores da métrica CS obtidos com os MOEAs em cenários com 30, 50 e 100 itens, respectivamente. A linha representa o conjunto de Pareto X e a coluna representa o conjunto Y . Quanto mais

próximo o valor estiver de 1, mais soluções do algoritmo de coluna (Y) são dominadas por soluções do algoritmo de linha (X). Cada algoritmo foi representado por uma sigla: ms =MS-MOEa, dns =DNSGA-II*, dmd =D-MEANDS-MD, $dm2$ =D-MEANDS-II e $dm3$ =D-MEANDS-III.

Na Tabela 21 (instâncias de 30 itens), o DNSGA-II* apresenta a pior convergência, tendo cerca de 40% das soluções do conjunto Pareto encontradas dominadas pelo conjunto Pareto dos outros algoritmos. O MS-MOEa apresenta situação semelhante quando comparado apenas com os algoritmos D-MEANDS-MD, D-MEANDS-II e D-MEANDS-III. O D-MEANDS-MD apresenta a melhor capacidade de convergência entre os algoritmos propostos (D-MEANDS-MD, D-MEANDS-II e D-MEANDS-III), pois o maior percentual de soluções do conjunto Pareto do D-MEANDS-MD dominadas por outros algoritmos é um pouco maior que 21%, enquanto para o D-MEANDS-II chega a ser um pouco maior que 35% e para o D-MEANDS-III, um pouco maior que 24%. Tanto a proposta D-MEANDS-II quanto a D-MEANDS-III adotaram estratégias que aumentaram a diversidade da busca, o que pode ter influenciado a convergência. No entanto, a capacidade de convergência do D-MEANDS-III é próxima à do D-MEANDS-MD neste conjunto de instâncias.

Na Tabela 22 (instâncias de 50 itens), o DNSGA-II* apresenta a pior convergência, com cerca de 50% das soluções do conjunto Pareto dominadas pelo conjunto Pareto dos outros algoritmos. O MS-MOEa apresenta cerca de 44% das soluções dominadas pelo conjunto Pareto dos algoritmos D-MEANDS-MD, D-MEANDS-II e D-MEANDS-III. Novamente, o D-MEANDS-MD apresenta a melhor capacidade de convergência entre os algoritmos propostos (D-MEANDS-MD, D-MEANDS-II e D-MEANDS-III) nas instâncias de 4 e 6 objetivos. O maior percentual de soluções do conjunto Pareto do D-MEANDS-MD dominadas por outros algoritmos é de 22%, enquanto para o D-MEANDS-II é um pouco maior que 51% e para o D-MEANDS-III, um pouco maior que 34%. Ainda assim, na instância de 8 objetivos, o D-MEANDS-III apresenta melhor convergência que o D-MEANDS-MD e o D-MEANDS-II, com o maior percentual de soluções dominadas sendo um pouco maior que 20%, contra 46% do D-MEANDS-II e 23% do D-MEANDS-MD.

Na Tabela 23 (instâncias de 100 itens), o DNSGA-II* apresenta a pior convergência, com cerca de 94% das soluções do conjunto Pareto dominadas pelo conjunto dos outros algoritmos. O MS-MOEa apresenta cerca de 80% das soluções dominadas pelo conjunto Pareto dos algoritmos D-MEANDS-MD, D-MEANDS-II e D-MEANDS-III. Novamente, o D-MEANDS-MD apresenta melhor convergência entre os algoritmos propostos nas instâncias de 4 e 6 objetivos, com 15% das soluções dominadas por outros algoritmos. Para o D-MEANDS-II, esse valor chega a 59% e para o D-MEANDS-III, 44%. Na instância de 8 objetivos, o maior percentual de soluções dominadas do D-MEANDS-III é um pouco maior que 17%, enquanto para o D-MEANDS-II chega a 58% e para o D-MEANDS-MD, 21%. Esses dados indicam que, na instância de 8 objetivos, o D-MEANDS-III apresenta melhor convergência que o D-MEANDS-MD e o D-MEANDS-II.

Tabela 21 – Comparação dos algoritmos através da métrica CS em instâncias de 30 itens e EC=20.

4 objetivos	X/Y	ms	dns	dmd	dm2	dm3
	MS-MOEA (ms)	-	0,405	0,09	0,135	0,114
	DNSGA-II* (dns)	0,222	-	0,051	0,081	0,069
	D-MEANDS-MD (dmd)	0,542	0,639	-	0,284	0,244
	D-MEANDS-II (dm2)	0,475	0,58	0,171	-	0,209
	D-MEANDS-III (dm3)	0,491	0,587	0,2	0,267	-
6 objetivos	X/Y	ms	dns	dmd	dm2	dm3
	MS-MOEA (ms)	-	0,492	0,093	0,155	0,102
	DNSGA-II* (dns)	0,126	-	0,032	0,059	0,038
	D-MEANDS-MD (dmd)	0,554	0,703	-	0,355	0,246
	D-MEANDS-II (dm2)	0,435	0,634	0,164	-	0,181
	D-MEANDS-III (dm3)	0,494	0,671	0,216	0,325	-
8 objetivos	X/Y	ms	dns	dmd	dm2	dm3
	MS-MOEA (ms)	-	0,532	0,071	0,129	0,08
	DNSGA-II* (dns)	0,065	-	0,014	0,029	0,016
	D-MEANDS-MD (dmd)	0,545	0,725	-	0,335	0,219
	D-MEANDS-II (dm2)	0,407	0,665	0,139	-	0,15
	D-MEANDS-III (dm3)	0,48	0,713	0,198	0,308	-

Tabela 22 – Comparação dos algoritmos através da métrica CS em instâncias de 50 itens e EC=20.

4 objetivos	X/Y	ms	dns	dmd	dm2	dm3
	MS-MOEA (ms)	-	0,524	0,09	0,174	0,137
	DNSGA-II* (dns)	0,161	-	0,027	0,068	0,054
	D-MEANDS-MD (dmd)	0,656	0,806	-	0,442	0,347
	D-MEANDS-II (dm2)	0,508	0,698	0,178	-	0,254
	D-MEANDS-III (dm3)	0,517	0,679	0,205	0,335	-
6 objetivos	X/Y	ms	dns	dmd	dm2	dm3
	MS-MOEA (ms)	-	0,507	0,044	0,149	0,051
	DNSGA-II* (dns)	0,118	-	0,012	0,054	0,018
	D-MEANDS-MD (dmd)	0,71	0,823	-	0,511	0,257
	D-MEANDS-II (dm2)	0,449	0,661	0,098	-	0,12
	D-MEANDS-III (dm3)	0,61	0,759	0,22	0,45	-
8 objetivos	X/Y	ms	dns	dmd	dm2	dm3
	MS-MOEA (ms)	-	0,53	0,033	0,112	0,03
	DNSGA-II* (dns)	0,068	-	0,005	0,024	0,006
	D-MEANDS-MD (dmd)	0,699	0,815	-	0,463	0,201
	D-MEANDS-II (dm2)	0,451	0,69	0,089	-	0,086
	D-MEANDS-III (dm3)	0,671	0,811	0,236	0,459	-

Tabela 23 – Comparação dos algoritmos através da métrica CS em instâncias de 100 itens e EC=20.

4 objetivos	X/Y	ms	dns	dmd	dm2	dm3
	MS-MOEA (ms)	-	0,607	0,036	0,116	0,113
	DNSGA-II* (dns)	0,133	-	0,003	0,022	0,027
	D-MEANDS-MD (dmd)	0,806	0,94	-	0,518	0,442
	D-MEANDS-II (dm2)	0,619	0,85	0,113	-	0,286
	D-MEANDS-III (dm3)	0,556	0,769	0,105	0,267	-
6 objetivos	X/Y	ms	dns	dmd	dm2	dm3
	MS-MOEA (ms)	-	0,674	0,036	0,195	0,055
	DNSGA-II* (dns)	0,034	-	0,001	0,022	0,005
	D-MEANDS-MD (dmd)	0,719	0,912	-	0,596	0,253
	D-MEANDS-II (dm2)	0,352	0,777	0,046	-	0,087
	D-MEANDS-III (dm3)	0,58	0,86	0,155	0,486	-
8 objetivos	X/Y	ms	dns	dmd	dm2	dm3
	MS-MOEA (ms)	-	0,56	0,009	0,092	0,008
	DNSGA-II* (dns)	0,04	-	0,001	0,012	0,001
	D-MEANDS-MD (dmd)	0,815	0,877	-	0,584	0,172
	D-MEANDS-II (dm2)	0,454	0,75	0,032	-	0,029
	D-MEANDS-III (dm3)	0,796	0,891	0,217	0,571	-

4.7.2 Aprofundando a Análise do Hipervolume

Podemos observar nos experimentos da seção 4.4 a superioridade do D-MEANDS-MD sobre os demais algoritmos em relação à métrica de hipervolume, quando submetidos ao mesmo número de avaliações de aptidão. Nesta seção, continuaremos a analisar o hipervolume, utilizando também informações visuais para aprofundar a análise numérica. As Figuras 23 e 24 mostram gráficos de coordenadas paralelas, comumente usados para visualizar conjuntos de soluções em problemas de muitos objetivos (mais de três objetivos). A Figura 23 apresenta dois conjuntos de soluções finais: as linhas e pontos pretos representam a aproximação da fronteira de Pareto final encontrada pelo D-MEANDS-MD na melhor execução em relação ao HV, enquanto as linhas e pontos azul-claro representam o conjunto final encontrado pela melhor execução do DDIS-MOEA/D-DE. A instância utilizada foi o DMKP com 4 objetivos, 100 itens e EC=20. Além disso, o gráfico também mostra o ponto de referência (em vermelho) utilizado para o cálculo do HV: $w_0 = (0, 0, 0, 0)$. Cada linha representa uma solução evoluída, conectando os valores encontrados nos quatro objetivos. Os eixos X , Y , W e Z correspondem às dimensões dos quatro objetivos do problema. O hipervolume calculado para o conjunto encontrado pelo D-MEANDS-MD é $HV = 4,4738 \times 10^7$, enquanto para o DDIS-MOEA/D-DE é $HV = 2,0549 \times 10^7$. Como os valores de hipervolume são bastante diferentes, é possível observar claramente no gráfico da Figura 24 as diferenças entre as duas aproximações da fronteira de Pareto. Além dos valores numéricos, o conjunto encontrado pelo D-MEANDS-MD apresenta desempenho visualmente superior.

No entanto, quando os resultados são mais próximos, a análise visual torna-se menos clara, pois as soluções dos dois conjuntos podem se sobrepor no gráfico. Isso ocorreu ao comparar os conjuntos de soluções encontrados pelo D-MEANDS-MD e pelo D-MEANDS na mesma instância do DMKP (4 objetivos, 100 itens e $EC=20$). O hipervolume calculado para o conjunto do D-MEANDS é $HV = 3,9697 \times 10^7$ (para o D-MEANDS-MD, como mencionado, é $HV = 4,4738 \times 10^7$). Nesse caso, os resultados foram plotados em gráficos separados: Figura 24(a) para o D-MEANDS-MD e Figura 24(b) para o D-MEANDS. Além disso, observou-se que a inclusão do ponto $w_0 = (0, 0, 0, 0)$ dificulta a comparação dos conjuntos, pois altera as escalas dos gráficos e provoca sobreposição das soluções. Por esse motivo, w_0 não foi incluído na Figura 24.

Comparando os conjuntos de soluções das Figuras 24(a) e (b), nota-se que, nas dimensões x e w , ambos os conjuntos ocupam intervalos semelhantes. Na dimensão z , o conjunto do D-MEANDS-MD (Figura 24(a)) apresenta valores significativamente maiores que o conjunto do D-MEANDS (Figura 24(b)). Na dimensão y , o D-MEANDS-MD tende a ocupar um intervalo ligeiramente menor, mas a diferença é menos relevante que na dimensão z . Assim, a diferença na dimensão z teve maior influência no cálculo do HV, resultando em um valor maior para o conjunto do D-MEANDS-MD. Adicionalmente, o conjunto do D-MEANDS-MD apresenta maior dispersão de pontos nos eixos, o que aumenta o hipervolume, uma vez que o cálculo do HV também reflete a diversidade das soluções finais, reduzindo a sobreposição dos hipercubos. Dessa análise visual, conclui-se que ambas as versões do D-MEANDS possuem convergência semelhante à fronteira de Pareto, mas o D-MEANDS-MD apresenta maior diversidade, elevando os valores da métrica HV.

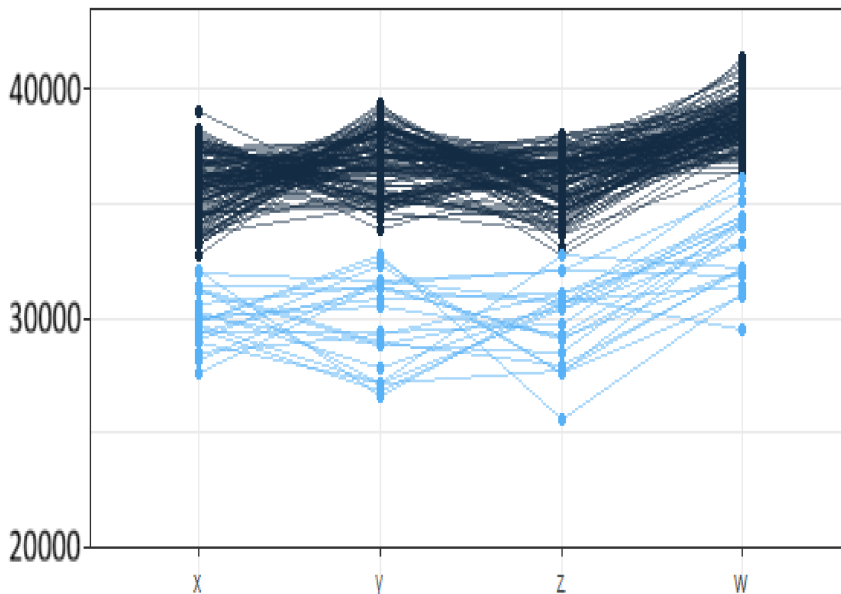


Figura 23 – Coordenadas paralelas das soluções encontradas por D-MEANDS-MD (linhas e pontos pretos) e DDIS-MOE/D-DE (linhas e pontos azuis) em suas melhores execuções para a instância DMKP com 4 objetivos, 100 itens e $EC=20$.

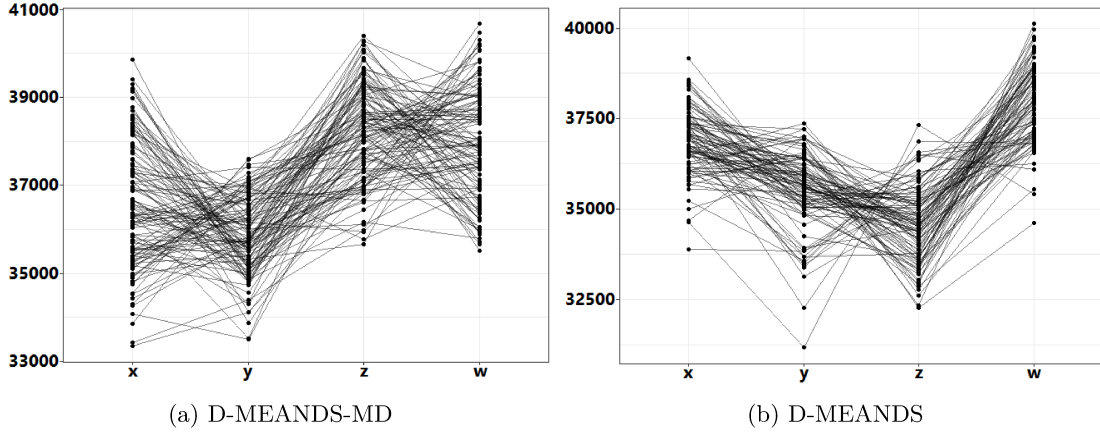


Figura 24 – Coordenadas paralelas das soluções encontradas por (a) D-MEANDS e (b) D-MEANDS-MD em suas melhores execuções para a instância DMKP com 4 objetivos, 100 itens e EC=20.

4.7.3 Comparando o Tempo de Execução do D-MEANDS-MD com D-MEANDS-II

Nesta subseção, apresentamos uma análise comparativa entre o tempo de execução do D-MEANDS-MD e do D-MEANDS-II. O D-MEANDS-MD é uma das propostas deste trabalho e apresentou excelentes resultados nas métricas HV e AD. No entanto, seu tempo de execução cresce de forma não linear à medida que o número de objetivos do problema (DMKP) aumenta. Com isso em mente, foi proposto o D-MEANDS-II, que adota uma estratégia visando reduzir o custo computacional do algoritmo, sem comprometer significativamente seu desempenho. A seção 4.5 apresenta uma análise comparativa entre D-MEANDS-MD e D-MEANDS-II. Os resultados mostraram que o D-MEANDS-II foi competitivo com o D-MEANDS-MD nas métricas avaliadas, mas com um tempo de execução significativamente menor.

A quantidade de subpopulações usadas nesses dois algoritmos varia conforme o número de objetivos do problema. O principal diferencial do D-MEANDS-II em relação ao D-MEANDS-MD é a redução do número de subpopulações. Para o D-MEANDS-MD, a quantidade de subpopulações geradas pode ser calculada pela Equação (11), onde m representa o número de objetivos.

$$Subpop_{D-MEANDS-MD} = \sum_{r=2}^{m-1} \frac{m!}{r!(m-r)!} = 2^m - m - 2 \quad (11)$$

sendo m o número de objetivos considerado.

A proposta do D-MEANDS-II é utilizar apenas as subpopulações que evoluem considerando dois objetivos do problema (ou seja, cada subpopulação evolui como se fosse um problema de dois objetivos). Para o D-MEANDS-II, a quantidade de subpopulações geradas pode ser calculada como:

$$Subpop_{D-MEANS-II} = \frac{m!}{2!(m-2)!} = \frac{m^2 - m}{2} \quad (12)$$

O fator de crescimento da quantidade de subpopulações geradas pelo D-MEANDS-MD em proporção ao D-MEANDS-II é dado por:

$$\frac{\sum_{r=2}^{m-1} \frac{m!}{r!(m-r)!}}{\frac{m!}{2!(m-2)!}} = \sum_{r=2}^{m-1} \frac{2!(m-2)!}{r!(m-r)!} = \frac{2^{m+1} - 4}{m^2 - m} \quad (13)$$

O Gráfico 25 apresenta a variação desse fator em função da quantidade de objetivos do problema. Como pode ser observado, a curva resultante apresenta um comportamento não linear, com o número de subpopulações do D-MEANDS-MD chegando a ser mais de 60 vezes maior do que a quantidade de subpopulações do D-MEANDS-II no problema com 12 objetivos.



Figura 25 – Fator de crescimento da quantidade de subpopulações entre D-MEANDS-MD e D-MEANDS-II.

Por meio de experimentos, aferiu-se o tempo de execução médio dos dois algoritmos em 100 execuções nas instâncias com 4, 6, 8, 10 e 12 objetivos, variando entre 30, 50 e 100 itens. Todas as execuções foram planejadas com 20 mudanças de ambiente (20 EC). Nas Figuras 26, 27 e 28, são comparados os tempos médios de execução do D-MEANDS-MD e do D-MEANDS-II em instâncias do problema da mochila com 30, 50 e 100 itens. Em cada figura, é possível notar o crescimento do tempo de execução dos algoritmos em relação ao número de objetivos. O gráfico à esquerda apresenta a dinâmica de crescimento considerando todas as instâncias avaliadas (de 4 até 12 objetivos), enquanto o gráfico à direita enfatiza os tempos obtidos nas instâncias menores (de 4 até 8 objetivos), a fim de destacar a diferença no comportamento de crescimento nesses casos. Em todas as figuras, é possível observar que a curva referente ao D-MEANDS-MD (linha azul) cresce

exponencialmente conforme o número de objetivos aumenta. Por outro lado, embora não seja tão evidente na figura devido à diferença considerável entre os tempos de execução dos dois algoritmos, o comportamento de crescimento do tempo de execução do D-MEANDS-II pode ser representado por uma função quadrática. Visto que o número de objetivos influencia diretamente o número de subpopulações dos algoritmos D-MEANDS-MD e D-MEANDS-II, os tempos de execução apresentados nos gráficos das Figuras 26, 27 e 28 refletem o impacto da quantidade de subpopulações utilizadas no custo computacional dos dois algoritmos. Isso demonstra que a redução na quantidade de subpopulações adotada pelo D-MEANDS-II foi uma estratégia eficaz.

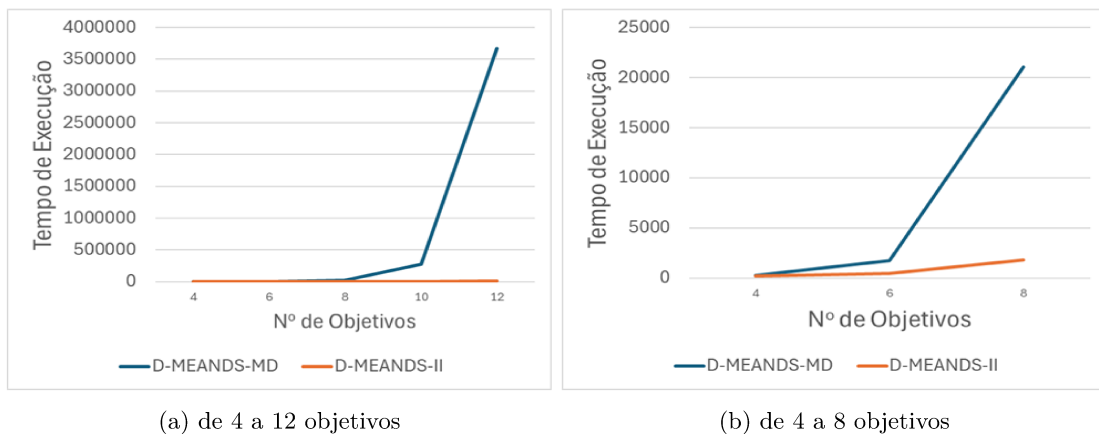


Figura 26 – Evolução do tempo de execução do D-MEANDS-MD e D-MEANDS-II em função do número de objetivos do problema, considerando instâncias de 30 itens e 20 mudanças de ambiente.

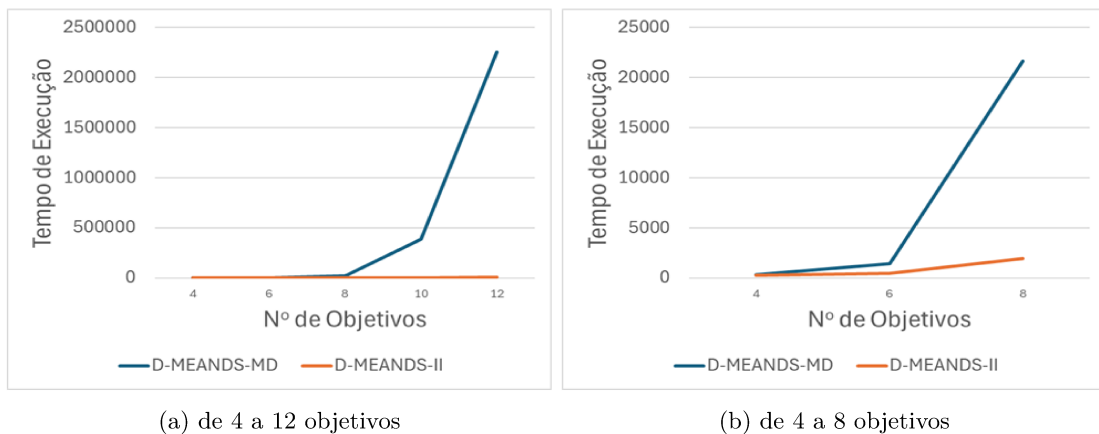


Figura 27 – Evolução do tempo de execução do D-MEANDS-MD e D-MEANDS-II em função do número de objetivos do problema, considerando instâncias de 50 itens e 20 mudanças de ambiente.

Em contrapartida, devido à sua estratégia evolutiva, o D-MEANDS-MD não possui limite de indivíduos em suas subpopulações. Dessa forma, o algoritmo acaba lidando com um grande volume de indivíduos, demandando um tempo maior de execução. Por outro lado, as subpopulações do D-MEANDS-II possuem limite de indivíduos. Por exemplo, nos

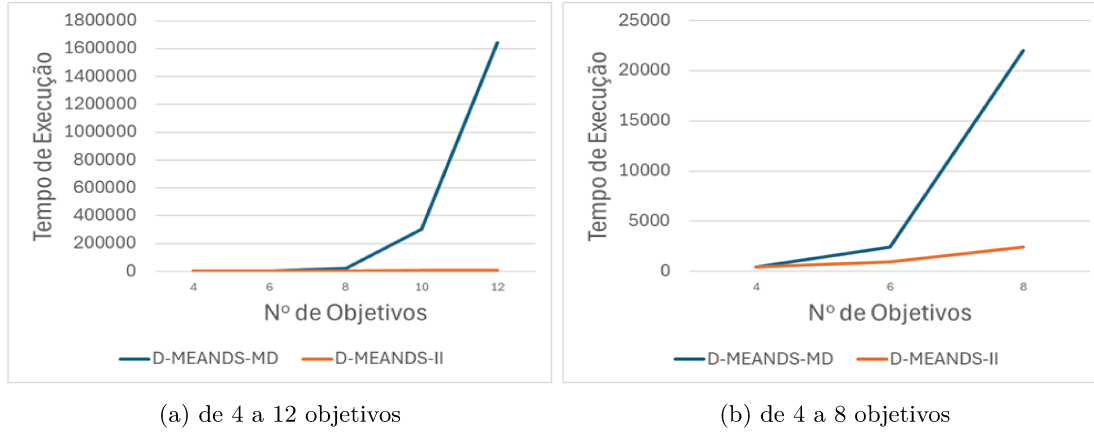


Figura 28 – Evolução do tempo de execução do D-MEANDS-MD e D-MEANDS-II em função do número de objetivos do problema, considerando instâncias de 100 itens e 20 mudanças de ambiente.

experimentos realizados nesta pesquisa, as subpopulações do D-MEANDS-II eram limitadas a 20 indivíduos. Destaca-se ainda que o D-MEANDS-II utiliza apenas subpopulações com 2 objetivos. Considerando que o conjunto de Pareto de um problema de 2 objetivos tende a ser menor que o de 3 objetivos (considerando o mesmo problema de otimização), a escolha de usar apenas subpopulações com 2 objetivos também afeta positivamente o tamanho (quantidade de indivíduos) das subpopulações, auxiliando na redução do tempo de execução do D-MEANDS-II. Como o tamanho da população do D-MEANDS-MD e do D-MEANDS-II varia de acordo com o número de objetivos, é possível mensurar o tamanho populacional do D-MEANDS-II, considerando que cada subpopulação consegue armazenar no máximo 20 indivíduos, como segue:

$$SizePop_{D-MEANDS-II} = 10(m^2 - m), \quad (14)$$

Como o D-MEANDS-MD não possui limite em suas subpopulações, não é possível determinar uma quantidade máxima de indivíduos nessas subpopulações. No entanto, o D-MEANDS-MD lida com uma grande quantidade de indivíduos, sendo que algumas subpopulações ultrapassaram 1.000 indivíduos nas instâncias de 6 e 8 objetivos. A fim de entender melhor a diferença do tamanho populacional entre o D-MEANDS-MD e o D-MEANDS-II, consideramos um limite de 100 indivíduos por subpopulação para o D-MEANDS-MD e comparamos seu tamanho populacional em relação ao D-MEANDS-II em função do aumento do número de objetivos. A Figura 29 mostra o crescimento populacional dos dois algoritmos. Pode-se observar que o crescimento populacional do D-MEANDS-MD é significativamente maior que o do D-MEANDS-II.

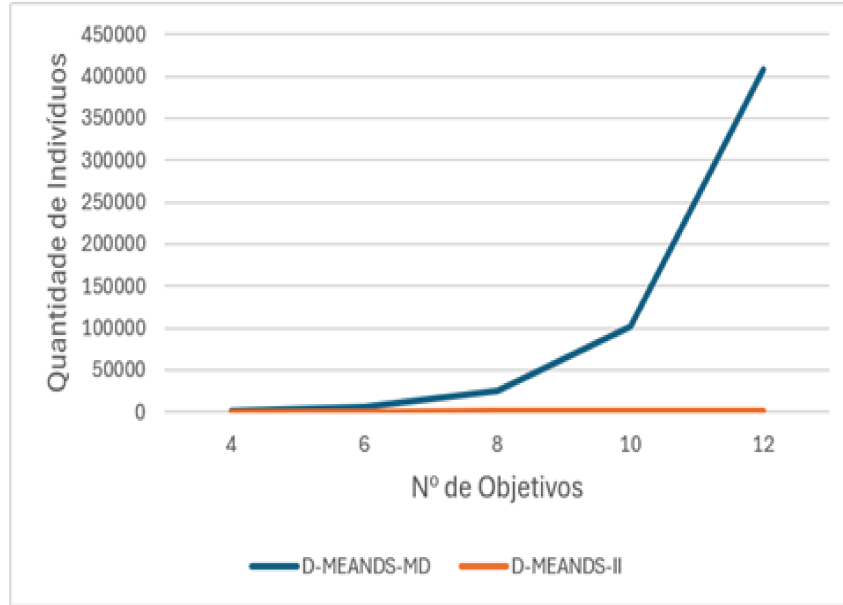


Figura 29 – Evolução do tamanho populacional entre D-MEANDS-MD e D-MEANDS-II, considerando um limite de 100 indivíduos por subpopulação para o D-MEANDS-MD e 20 para o D-MEANDS-II.

4.8 Síntese das Análises

Os experimentos foram divididos em fases. Na primeira fase (Seção 4.2) foi realizada uma análise comparativa do desempenho dos algoritmos evolutivos de otimização quando aplicados para resolver um problema dinâmico discreto com muitos objetivos: o *Dynamic Multiobjective Knapsack Problem 0-1* (DMKP). Três deles (DMSGA-II (KALYANMOY; BHASKARA; SINDHYA, 2007), MOEA/D-KF (MURUGANANTHAM; TAN; VADAKKEPAT, 2016) e MS-MOEA (WANG et al., 2010)) foram extraídos da literatura, sendo que anteriormente foram aplicados a problemas de otimização definidos sobre espaços contínuos. O DMSGA-III foi proposto neste trabalho, introduzindo ao NSGA-III (LI et al., 2014) a mesma estratégia dinâmica usada no DMSGA-II. Também foi proposta uma pequena alteração no DMSGA-II, introduzindo o uso de arquivo externo, o que resultou no DMSGA-II*.

Comparando os quatro algoritmos: MS-MOEA, DMSGA-II, DMSGA-III e MOEA/D-KF, em relação às métricas multiobjetivo utilizadas, HVR e $\overline{IGD}$, o MS-MOEA superou claramente os demais algoritmos. No entanto, o MS-MOEA apresentou um tempo de execução muito maior que os demais. Por outro lado, o DMSGA-II teve um bom desempenho (o segundo melhor), com um tempo de execução bem menor. Portanto, ambos os algoritmos podem ser considerados eficientes para este problema, dependendo se o tempo de execução é mais importante que um conjunto Pareto robusto e vice-versa. Se tanto o desempenho das métricas quanto o tempo de processamento forem igualmente importantes, o DMSGA-II é o mais recomendado. Acredita-se que o emprego de um arquivo externo seja responsável tanto pelo bom desempenho do MS-MOEA quanto pelo proces-

samento mais custoso. Essas informações ajudam o algoritmo a se adaptar às mudanças ambientais, mas consomem um tempo extra para verificar a dominância de cada nova solução em relação a todas as outras armazenadas no arquivo atual. Essa observação levou à proposta de uma modificação para o DNSGA-II usando arquivo externo. O algoritmo resultante foi chamado aqui DNSGA-II*, que retornou resultados melhores em relação ao original, com um pequeno custo computacional adicional. Concluímos que o DNSGA-II* é a melhor opção entre os algoritmos avaliados para as instâncias do DMKP investigadas neste trabalho. Ele retorna bons resultados nas métricas HVR e $\overline{IGD}$ com um tempo de processamento razoável. O DNSGA-III apresentou desempenho inferior ao DNSGA-II, embora se esperasse que a adaptação do NSGA-III oferecesse vantagens em relação aos demais algoritmos, já que o DMKP é um problema de muitos objetivos. Pelo contrário, seu desempenho diminui à medida que o número de objetivos aumenta.

Na segunda fase (Seção 4.3), um novo algoritmo chamado D-MEANDS foi proposto, baseado em seu precursor MEANDS (LAFETA et al., 2018), sendo o MEANDS uma proposta para problemas estáticos. Para auxiliar na avaliação de desempenho do algoritmo proposto, selecionamos apenas dois algoritmos da literatura que mostraram melhor desempenho na primeira fase dos experimentos: DNSGA-II e MS-MOEA. Duas métricas comumente utilizadas para avaliar DMOPs foram aplicadas nesta fase: *Hypervolume Ratio* (HVR) e *Mean Inverted Generational Distance* ($\overline{IGD}$). O tempo de execução de cada algoritmo também foi avaliado. Formulações com 4 e 6 objetivos foram aplicadas a instâncias do DMKP com 30 e 50 itens na análise comparativa. Experimentos foram realizados com 1, 2, 5 e 10 mudanças de ambiente (ECs) ao longo da busca evolutiva, distribuídas uniformemente ao longo das gerações. Como passamos a utilizar métricas não paramétricas (HVR e $\overline{IGD}$) nesta fase, a análise comparativa foi feita entre quatro algoritmos: D-MEANDS, MS-MOEA, DNSGA-II* e um algoritmo mais recente da literatura, o DDIS-MOEA/D-DE (LIU et al., 2020).

Nos experimentos com 1 e 2 mudanças de ambiente ($EC = 1$ e 2), o MS-MOEA foi claramente o melhor entre os dois algoritmos da literatura, se analisarmos apenas as métricas HVR e $\overline{IGD}$. O desempenho do algoritmo proposto D-MEANDS é muito semelhante ao do MS-MOEA em quase todas as instâncias, apresentando melhorias com o aumento do número de objetivos de 4 para 6. Além disso, o MS-MOEA apresentou alto custo de tempo de execução comparado ao DNSGA-II nos experimentos com 4 e 6 objetivos. O D-MEANDS, por sua vez, exigiu um tempo de processamento menor que o DNSGA-II na formulação com 4 objetivos, porém com custo maior nas instâncias com 6 objetivos. Nessas instâncias, aproximou-se do tempo de processamento do MS-MOEA. Podemos afirmar que, em instâncias de 4 objetivos, o algoritmo proposto D-MEANDS apresentou bom desempenho considerando as métricas HVR e $\overline{IGD}$, com baixo tempo de processamento. O aumento do número de objetivos fortalece o D-MEANDS em termos de convergência e diversidade, mas ao custo de um aumento considerável no tempo de processamento.

Logo, apresentamos os experimentos com 5 e 10 mudanças de ambiente ($EC = 5$ e 10). Considerando as instâncias com 4 objetivos, observam-se algumas mudanças significativas na análise comparativa dos algoritmos avaliados. O aumento do número de EC provocou uma diminuição no número de gerações de um ambiente para outro, visto que a quantidade total de gerações da busca não foi alterada, impactando o comportamento de todos os algoritmos. O MS-MOEA reduziu seu tempo de processamento em comparação aos experimentos com poucas mudanças de ambiente. Por outro lado, o DNSGA-II tornou-se mais competitivo em relação ao MS-MOEA nas métricas HVR e $\overline{IGD}$, embora tenha havido um aumento no seu tempo de processamento. No entanto, o DNSGA-II ainda é mais rápido que o MS-MOEA. O algoritmo proposto D-MEANDS apresentou melhoria significativa nas métricas de desempenho, embora seu tempo de processamento também tenha aumentado nas instâncias de 50 itens. O desempenho do D-MEANDS em ambas as métricas (HVR e $\overline{IGD}$) foi superior ao do DNSGA-II e MS-MOEA, com significância estatística comprovada pelo Teste de Hipótese T. Considerando as instâncias com 4 objetivos, os tempos de processamento de D-MEANDS e DNSGA-II são muito próximos para 30 itens, enquanto D-MEANDS demanda mais tempo quando o número de itens é aumentado para 50.

Na última seção da segunda fase, utilizamos métricas não paramétricas: Hipervolume (HV) e *Density* (AD). Nesse experimento, expandimos a quantidade de instâncias, usando instâncias de 4, 6 e 8 objetivos com 30, 50 e 100 itens. A quantidade de mudanças de ambiente variou entre 10, 15 e 20 EC. Além disso, o DDIS-MOEA/D-DE (LIU et al., 2020), que usa evolução diferencial como estratégia de busca evolutiva subjacente, foi utilizado nos experimentos. O D-MEANDS superou os algoritmos da literatura na maioria das instâncias, especialmente em instâncias de 8 objetivos. No entanto, o aumento da mudança de ambiente (EC) faz com que o desempenho do D-MEANDS decaia em relação aos demais algoritmos. Vale ressaltar que nossos experimentos mostram que, ao aumentar a complexidade das instâncias, o tempo de processamento do D-MEANDS aumenta relativamente mais que nos demais algoritmos. Esse aumento de custo é devido à característica do D-MEANDS de não ter um limite populacional em todos os subconjuntos de soluções não dominadas. Em instâncias de 4 objetivos, o D-MEANDS é a opção mais rápida, próximo ao tempo de execução do DNSGA-II*. No entanto, com o aumento do número de objetivos, o DNSGA-II* mostrou-se muito mais rápido que os outros algoritmos, incluindo o D-MEANDS.

Na terceira fase (Seção 4.4) foi proposto o D-MEANDS-MD. Este algoritmo é um refinamento do D-MEANDS, onde é empregada uma nova estratégia para inserir diversidade durante o processo evolutivo. Duas métricas não paramétricas utilizadas para avaliar DMOPs foram aplicadas na avaliação de desempenho: Hipervolume (HV) e *Density* (AD). O tempo de processamento de cada algoritmo também foi avaliado. Formulações com 4, 6 e 8 objetivos foram aplicadas a instâncias do DMKP com 30, 50 e 100 itens na análise

comparativa. Experimentos foram realizados aplicando 10, 15 e 20 mudanças de ambiente (ECs) ao longo da busca evolutiva, distribuídas uniformemente ao longo das gerações. Para avaliar o desempenho do D-MEANDS-MD, utilizamos os resultados dos algoritmos D-MEANDS e DNSGA-II*. Uma variação do DNSGA-II* foi proposta usando a mesma estratégia de inserção de diversidade do D-MEANDS-MD, denominada DNSGA-II⁺, com o objetivo de analisar o comportamento da estratégia de inserção de diversidade em outro algoritmo.

A proposta D-MEANDS-MD, que utiliza um mecanismo simples para inserir diversidade na população principal, foi analisada e superou todos os outros algoritmos em todas as 27 instâncias do DMKP quando consideradas as métricas HV e AD. No entanto, seu tempo de processamento é semelhante ao do D-MEANDS, sendo levemente maior.

Na quarta fase (Seção 4.5) dos experimentos, avaliamos uma nova proposta inspirada na estratégia do D-MEANDS. O D-MEANDS-II foi baseado no algoritmo MEANDS-II, que é um algoritmo para problemas estáticos e já foi aplicado no *Multiobjective Knapsack Problem* (MKP). O D-MEANDS-II utiliza a mesma estratégia dinâmica aplicada no D-MEANDS e D-MEANDS-MD, mas com muito menos subpopulações que os dois algoritmos. O objetivo é propor um algoritmo com tempo de execução bem menor que o D-MEANDS-MD, mantendo convergência e diversidade próximas. O D-MEANDS-II utiliza um processo evolutivo similar ao do D-MEANDS, porém com tamanho populacional limitado e menor quantidade de subpopulações. O D-MEANDS-II obteve um tempo de processamento muito menor que o D-MEANDS-MD, próximo ao tempo do DNSGA-II*. Ele supera o D-MEANDS-MD na métrica HV e é inferior na métrica AD, mas seus resultados não são tão distantes. O D-MEANDS-II consegue superar o MS-MOEA e o DNSGA-II* nas métricas, assim como o D-MEANDS-MD. Nesta fase, dentre as propostas deste trabalho, o algoritmo que melhor equilibra desempenho e tempo de processamento é o D-MEANDS-II.

A quinta fase (Seção 4.6) apresenta mais uma proposta da tese, o D-MEANDS-III. Este algoritmo é baseado na dinâmica do D-MEANDS-II, onde são empregadas duas novas estratégias evolutivas que utilizam melhor as informações do processo multi-populacional do D-MEANDS-II. A primeira estratégia propõe uma nova forma de seleção dos pais para o crossover, melhorando os resultados nas métricas HV e AD a custo de um leve aumento no tempo de execução. A segunda estratégia apresenta uma nova dinâmica aplicada durante a mudança de ambiente, mais elaborada que a usada pelo D-MEANDS-II. Essa nova dinâmica melhora o tempo de execução do algoritmo, sem grandes alterações nos resultados das métricas. Combinando essas duas estratégias com a dinâmica do D-MEANDS-II, foi proposto o D-MEANDS-III. Este novo algoritmo foi comparado com MS-MOEA, DNSGA-II*, D-MEANDS-MD e D-MEANDS-II. Nesta fase, também foram utilizadas métricas não paramétricas: Hipervolume (HV) e *Density* (AD). Assim como o D-MEANDS-MD e D-MEANDS-II, o D-MEANDS-III superou o MS-MOEA e o DNSGA-

II* em todas as métricas. Na métrica AD, o D-MEANDS-III supera o D-MEANDS-II, mas é superado pelo D-MEANDS-MD. Na métrica HV, o D-MEANDS-III supera ambos os algoritmos D-MEANDS-MD e D-MEANDS-II. Mostrando que a distribuição das soluções no conjunto Pareto encontrado pelo D-MEANDS-MD é melhor que a do D-MEANDS-III, mas a relação entre convergência e diversidade do D-MEANDS-III é superior à do D-MEANDS-MD e D-MEANDS-II.

Em uma análise complementar (Seção 4.7), comparamos os algoritmos par a par usando a métrica *Coverage of two Sets* (CS). Essa métrica avalia quantas soluções do algoritmo Y são dominadas pelo algoritmo X , dando uma ideia de qual dos dois algoritmos possui melhor convergência. Os algoritmos usados neste experimento foram: DNSGA-II*, MS-MOEA, D-MEANDS-MD, D-MEANDS-II e D-MEANDS-III. O D-MEANDS-MD apresentou melhor convergência comparado aos demais algoritmos; no entanto, nas instâncias de 8 objetivos com 50 e 100 itens, o D-MEANDS-III superou o D-MEANDS-MD em termos de convergência. Além disso, uma análise da métrica Hipervolume (HV) é apresentada nesta seção, mostrando a diferença de comportamento entre D-MEANDS e D-MEANDS-MD, justificando como a inserção de diversidade ajudou a encontrar valores melhores na métrica HV do que os apresentados pelo D-MEANDS. Por último, discutimos a comparação entre o tempo de execução do D-MEANDS-MD e D-MEANDS-II, apresentando uma análise que justifica como a redução da quantidade de subpopulações impactou no tempo de execução do D-MEANDS-II.

Conclusão

Nesta tese, investigamos novos algoritmos evolutivos capazes de lidar com problemas de otimização dinâmicos, discretos e com múltiplos objetivos. Para tal, adaptamos o algoritmo MEANDS (LAFETA et al., 2018), proposto anteriormente para problemas de otimização discreta com vários objetivos em um ambiente estático. Duas novas propostas surgiram durante essa investigação, nas quais adaptamos o MEANDS para ambientes dinâmicos: a versão inicial D-MEANDS e uma versão mais refinada, denominada D-MEANDS-MD. Posteriormente, adaptamos o predecessor do algoritmo MEANDS, o MEANDS-II, para ambientes dinâmicos, resultando em duas novas propostas descritas neste trabalho: D-MEANDS-II e D-MEANDS-III. Para avaliar a qualidade dessas propostas e, ao mesmo tempo, compreender as estratégias evolutivas empregadas em ambientes dinâmicos, foram realizados experimentos comparativos com algoritmos evolutivos propostos na literatura para lidar com os DMOPs (MS-MOEA, DNSGA-II, DDIS-MOEA/D-DE e MOEA/D-KF). Além das versões originais desses algoritmos, algumas variações também foram propostas neste trabalho: DNSGA-II*, DNSGA-II⁺ e DNSGA-III.

As métricas empregadas inicialmente nessas comparações são paramétricas (HVR (ZITZLER; THIELE, 1999) e $\overline{IGD}$ (WANG; LI, 2009)). Entretanto, devido à necessidade de calcular o conjunto de Pareto aproximado para obter essas medidas, apenas experimentos com 1, 2, 5 e 10 mudanças de ambiente (EC) foram viáveis de serem executados (EC = 1, 2, 5 e 10). Assim, para que experimentos mais dinâmicos pudessem ser empregados, utilizaram-se métricas não paramétricas (HV (ZITZLER; THIELE, 1998) e AD (ZHANG; QIAN, 2011)). Realizamos experimentos com 10, 15 e 20 mudanças de ambiente (EC). Nos últimos experimentos, a métrica CS (ZITZLER, 1999) também foi utilizada para realizar uma comparação entre os algoritmos melhor avaliados. Além disso, uma análise visual dos conjuntos de soluções não dominadas também foi apresentada para melhor compreensão dos resultados de Hipervolume. Por último, demonstramos como a redução do número de subpopulações do D-MEANDS-II, em relação ao D-MEANDS-MD, impactou o tempo de execução.

As principais conclusões deste trabalho são as seguintes: as propostas D-MEANDS e

D-MEANDS-MD, de forma geral, são superiores aos algoritmos da literatura no problema DMKP (e suas variações) em relação às métricas multiobjetivo empregadas, principalmente quando o número de objetivos aumenta de 6 para 8. Entretanto, essa melhoria nos resultados é acompanhada por um grande aumento no tempo de processamento, o que pode ser um impedimento para o uso do D-MEANDS e do D-MEANDS-MD a partir de 8 objetivos. Dentre as duas versões propostas do MEANDS, o D-MEANDS-MD apresenta o melhor desempenho nas métricas multiobjetivo, embora também demande tempo adicional de processamento, mantendo-se, ainda assim, próximo ao tempo do D-MEANDS. O D-MEANDS-II apresentou um tempo de processamento próximo ao do D-NSGA-II*, o algoritmo mais rápido investigado neste trabalho, sem grandes diferenças nas métricas quando comparado ao D-MEANDS-MD. O tempo de execução do D-MEANDS-MD e do D-MEANDS aumentava exponencialmente à medida que aumentava o número de objetivos do problema; já no D-MEANDS-II, o tempo de execução aumentava em ordem quártica em função do número de objetivos. Na análise complementar, verificamos que a quantidade de subpopulações é um fator que influencia o tempo de execução do D-MEANDS, do D-MEANDS-MD e do D-MEANDS-II. Como a quantidade de subpopulações do D-MEANDS-II é significativamente menor do que a do D-MEANDS e a do D-MEANDS-MD, seu tempo de execução também é consideravelmente menor. O D-MEANDS-III manteve um tempo de execução próximo ao do D-MEANDS-II e do D-NSGA-II*, ao mesmo tempo em que melhorou os resultados das métricas não paramétricas HV e AD, superando o D-MEANDS-MD na grande maioria das instâncias do DMKP na métrica HV. Além disso, o D-MEANDS-III supera o D-MEANDS-II em todas as instâncias do DMKP nas métricas HV e AD. Na análise complementar, em que comparamos os algoritmos MS-MOEA, D-NSGA-II*, D-MEANDS-MD, D-MEANDS-II e D-MEANDS-III utilizando a métrica CS, o D-MEANDS-MD apresentou melhor convergência nas instâncias com 4 e 6 objetivos. Visto que o D-MEANDS-MD obteve desempenho inferior na métrica HV em relação ao D-MEANDS-II e ao D-MEANDS-III nessas instâncias, conclui-se que esses dois algoritmos, D-MEANDS-II e D-MEANDS-III, apresentam melhor diversidade durante a busca evolutiva. Já nas instâncias com 8 objetivos, o D-MEANDS-III apresenta a melhor convergência, conforme indicado pela métrica CS.

5.1 Trabalhos Futuros

Uma evolução natural desta pesquisa é a realização de novos experimentos, ampliando a complexidade dos cenários avaliados, tanto em relação ao aumento no número de itens e/ou de objetivos, quanto ao acréscimo na quantidade de mudanças no ambiente (ex: 30 a 50 ECs). Esses experimentos possibilitarão uma análise mais aprofundada da eficiência dos MOEAs propostos em ambientes dinâmicos mais desafiadores. Pretendemos ainda explorar cenários nos quais a mudança de ambiente não seja controlada, tanto em relação

a quantidade quanto no momento das suas ocorrências, a fim de investigar o comportamento do algoritmo em cenários mais aleatórios (menos comportados), ou seja, analisar seu tempo de reação mínimo para ajustar seu mecanismo de busca ao novo ambiente. Além disso, os algoritmos propostos podem ser aplicados a outros problemas dinâmicos, como o problema de roteamento multiobjetivo dinâmico (VINODHINI; GOMATHY, 2020; VALENTINI et al., 2010; GUO et al., 2017), de modo a verificar sua eficácia no tratamento desses problemas. No caso específico do D-MEANDS-III, também é interessante investigar seu desempenho na resolução de DMOPs contínuos, uma vez que esse algoritmo trabalha com uma quantidade de subpopulações e seus respectivos tamanhos limitados, tornando-o propenso para lidar com esse tipo de problema. Outra linha a ser estudada é a adoção de novas estratégias evolutivas, capazes de explorar o comportamento multi-populacional dos algoritmos evolutivos, bem como o aprofundamento da análise sobre o conjunto de subpopulações utilizada pelo D-MEANDS-II e D-MEANDS-III, a fim de tentar reduzir o tempo de execução dos algoritmos sem grande impacto na sua convergência e diversidade. Outro aspecto importante que merece ser explorado em pesquisas futuras é a avaliação do desempenho de algoritmos multiobjetivo dinâmicos. Conforme discutido em (HELBIG; ENGELBRECHT, 2013), as métricas utilizadas para medir o desempenho dos algoritmos para DMOPs apresentam limitações significativas. Diante disso, torna-se relevante investigar novas métricas de desempenho mais robustas, que sejam menos vulneráveis a tais limitações e que possam ser utilizadas mesmo quando a fronteira de Pareto verdadeira é desconhecida.

5.2 Contribuições em Produção Bibliográfica

Em 2019, numa etapa preliminar dessa pesquisa e como continuidade à investigação iniciada na dissertação de mestrado (LAFETA, 2018), o algoritmo MEANDS-II foi proposto resultando na publicação (LAFETA; OLIVEIRA, 2019) no evento SMC2019.

Em 2020 publicamos no evento BRACIS2020 (*Brazilian Conference on Intelligent Systems* 2020) uma análise comportamental de diferentes algoritmos da literatura no *Dynamic Multiobjective Knapsack Problem* 0-1 (DMKP), que se refere basicamente aos experimentos descritos na Seção 4.2.

Em 2020, publicamos no evento SMC2020 (*International Conference on Systems, Man, and Cybernetics* 2020) propondo uma adaptação do algoritmo MEANDS (LAFETA et al., 2018) para problemas dinâmicos (D-MEANDS), que foi aplicada e avaliada no *Dynamic Multiobjective Knapsack Problem* 0-1 (DMKP). essa publicação está diretamente relacionada aos experimentos da Seção 4.3.

Em 2024, publicamos no periódico KER (*The Knowledge Engineering Review*) uma melhoria do algoritmo D-MEANDS, denominada D-MEANDS-MD, que insere diversidade no processo evolutivo. Essa publicação está relacionada às Seções 4.4 e 4.7.

Referências

- ARROYO, J. E. C. et al. Heurísticas e metaheurísticas para otimização combinatória multiobjetivo. **Biblioteca Digital da Unicamp**, p. 16, 2002.
- AZZOUZ, R.; BECHIKH, S.; SAID, L. B. Dynamic multi-objective optimization using evolutionary algorithms: a survey. In: **Recent advances in evolutionary multi-objective optimization**. [S.l.]: Springer, 2017. p. 31–70.
- BOSMAN, P. A.; THIERENS, D. The balance between proximity and diversity in multiobjective evolutionary algorithms. **IEEE transactions on evolutionary computation**, IEEE, v. 7, n. 2, p. 174–188, 2003.
- BRANKE, J. **Evolutionary optimization in dynamic environments**. [S.l.]: Springer Science & Business Media, 2012. v. 3.
- CAMARA, M.; ORTEGA, J.; TORO, F. J. Parallel processing for multi-objective optimization in dynamic environments. In: IEEE. **2007 IEEE International Parallel and Distributed Processing Symposium**. [S.l.], 2007. p. 1–8.
- CANTÚ-PAZ, E. Topologies, migration rates, and multi-population parallel genetic algorithms. In: **Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1**. [S.l.: s.n.], 1999. p. 91–98.
- COELLO, C. C. An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends. In: IEEE. **Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)**. [S.l.], 1999. v. 1, p. 3–13.
- COSTA, D. An evolutionary tabu search algorithm and the nhl scheduling problem. **INFOR: Information Systems and Operational Research**, Taylor & Francis, v. 33, n. 3, p. 161–178, 1995.
- DAS, I.; DENNIS, J. E. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. **SIAM journal on optimization**, SIAM, v. 8, n. 3, p. 631–657, 1998.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. **IEEE transactions on evolutionary computation**, IEEE, v. 6, n. 2, p. 182–197, 2002.

- ECKSTEIN, Z.; WOLPIN, K. I. The specification and estimation of dynamic stochastic discrete choice models: A survey. **The Journal of Human Resources**, JSTOR, v. 24, n. 4, p. 562–598, 1989.
- ESHELMAN, L. J. Preventing premature convergence in genetic algorithms by preventing incest. In: **Proc. of the 4th. Conf. on GA, July, 1991**. [S.l.: s.n.], 1991. p. 115–122.
- FARINA, M.; DEB, K.; AMATO, P. Dynamic multiobjective optimization problems: test cases, approximations, and applications. **IEEE Transactions on evolutionary computation**, IEEE, v. 8, n. 5, p. 425–442, 2004.
- FOGEL, D. B. **Artificial intelligence through simulated evolution**. [S.l.]: Wiley-IEEE Press, 1998.
- FONSECA, C. M.; FLEMING, P. J. Multiobjective genetic algorithms. In: IET. **IEE colloquium on genetic algorithms for control systems engineering**. [S.l.], 1993. p. 6–1.
- FONSECA, C. M.; FLEMING, P. J. et al. Genetic algorithms for multiobjective optimization: Formulation discussion and generalization. In: CITESEER. **Icga**. [S.l.], 1993. v. 93, n. July, p. 416–423.
- FRANÇA, T. P.; MARTINS, L. G.; OLIVEIRA, G. M. Maco/nds: Many-objective ant colony optimization based on non-dominated sets. In: IEEE. **2018 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2018. p. 1–8.
- GAO, Y.; SHI, L.; YAO, P. Study on multi-objective genetic algorithm. In: IEEE. **Proceedings of the 3rd World Congress on Intelligent Control and Automation (Cat. No. 00EX393)**. [S.l.], 2000. v. 1, p. 646–650.
- GLOVER, F. Tabu search for nonlinear and parametric optimization (with links to genetic algorithms). **Discrete Applied Mathematics**, Elsevier, v. 49, n. 1-3, p. 231–255, 1994.
- GLOVER, F.; KELLY, J. P.; LAGUNA, M. Genetic algorithms and tabu search: hybrids for optimization. **Computers & Operations Research**, Elsevier, v. 22, n. 1, p. 111–134, 1995.
- GOH, C.-K.; TAN, K. C. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 13, n. 1, p. 103–127, 2008.
- GOLDBERG, D. E. A note on boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing. **Complex Systems**, Complex Systems Publications, Inc., v. 4, p. 445–460, 1990.
- GOLDBERG, D. E.; KORB, B.; DEB, K. Messy genetic algorithms: Motivation, analysis, and first results. **Complex systems**, Complex Systems Publications, Champaign, IL, USA, v. 3, n. 5, p. 493–530, 1989.
- GU, Q. et al. Surrogate-assisted evolutionary algorithm for expensive constrained multi-objective discrete optimization problems. **Complex & Intelligent Systems**, Springer, v. 8, n. 4, p. 2699–2718, 2022.

- GUO, Y.-N. et al. Robust dynamic multi-objective vehicle routing optimization method. **IEEE/ACM transactions on computational biology and bioinformatics**, IEEE, v. 15, n. 6, p. 1891–1903, 2017.
- HATZAKIS, I.; WALLACE, D. Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach. In: **Proceedings of the 8th annual conference on Genetic and evolutionary computation**. [S.l.: s.n.], 2006. p. 1201–1208.
- HELBIG, M.; ENGELBRECHT, A. P. Performance measures for dynamic multi-objective optimisation algorithms. **Information Sciences**, Elsevier, v. 250, p. 61–81, 2013.
- HELBIG, M. et al. **Solving dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation**. Tese (Doutorado) — University of Pretoria, 2012.
- HIROYASU, T.; MIKI, M.; WATANABE, S. Divided range genetic algorithms in multiobjective optimization problems. **Proc. of IWES**, Citeseer, v. 99, p. 57–65, 1999.
- HOLLAND, J. Adaptation in natural and artificial systems, univ. of mich. press. **Ann Arbor**, 1975.
- HORN, J. rey; NAFPLIOTIS, N.; GOLDBERG, D. E. Multiobjective optimization using the niched pareto genetic algorithm. **IlliGAL report**, v. 93005, p. 61801–2296, 1993.
- IGLESIAS, A.; GÁLVEZ, A. Genetic algorithm applications for challenging real-world problems: Some recent advances and future trends. **Applied Genetic Algorithm and Its Variants: Case Studies and New Developments**, Springer, p. 21–50, 2023.
- ISHIBUCHI, H.; AKEDO, N.; NOJIMA, Y. A study on the specification of a scalarizing function in moea/d for many-objective knapsack problems. In: SPRINGER. **International Conference on Learning and Intelligent Optimization**. [S.l.], 2013. p. 231–246.
- ISHIBUCHI, H.; TSUKAMOTO, N.; NOJIMA, Y. Evolutionary many-objective optimization: A short review. In: IEEE. **2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)**. [S.l.], 2008. p. 2419–2426.
- JIANG, M. et al. Solving dynamic multi-objective optimization problems via support vector machine. In: IEEE. **2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)**. [S.l.], 2018. p. 819–824.
- JIANG, S.; ZOU, J.; YAO, X. Evolutionary dynamic multi-objective optimisation: A survey. **ACM Computing Surveys**, v. 55, 03 2022.
- JIN, Y.; BRANKE, J. Evolutionary optimization in uncertain environments-a survey. **IEEE Transactions on evolutionary computation**, IEEE, v. 9, n. 3, p. 303–317, 2005.
- JONG, K. A. D.; SPEARS, W. M. et al. Using genetic algorithms to solve np-complete problems. In: **ICGA**. [S.l.: s.n.], 1989. p. 124–132.

- KALMAN, R. E.; BUCY, R. S. New results in linear filtering and prediction theory. 1961.
- KALYANMOY, D.; BHASKARA, R. N. U.; SINDHYA, K. Dynamic multi-objective optimization and decision-making using modified nsga-ii: A case study on hydro-thermal power scheduling. In: SPRINGER. **International conference on evolutionary multi-criterion optimization**. [S.l.], 2007. p. 803–817.
- KARAMI, F.; DARIANE, A. B. A review and evaluation of multi and many-objective optimization: Methods and algorithms. **Global Journal of Ecology**, v. 7, n. 2, p. 104–119, 2022.
- KATOCH, S.; CHAUHAN, S. S.; KUMAR, V. A review on genetic algorithm: past, present, and future. **Multimedia tools and applications**, Springer, v. 80, p. 8091–8126, 2021.
- KELLERER, H.; PFERSCHY, U.; PISINGER, D. Multidimensional knapsack problems. In: **Knapsack problems**. [S.l.]: Springer, 2004. p. 235–283.
- KITANO, H. Empirical studies on the speed of convergence of neural network training using genetic algorithms. In: **AAAI**. [S.l.: s.n.], 1990. v. 90, p. 789–795.
- KNOWLES, J.; CORNE, D. On metrics for comparing nondominated sets. In: IEEE. **Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)**. [S.l.], 2002. v. 1, p. 711–716.
- KOO, W. T.; GOH, C. K.; TAN, K. C. A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment. **Memetic Computing**, Springer, v. 2, n. 2, p. 87–110, 2010.
- LAFETA, T. et al. Meands: A many-objective evolutionary algorithm based on non-dominated decomposed sets applied to multicast routing. **Applied Soft Computing**, Elsevier, v. 62, p. 851–866, 2018.
- LAFETA, T. F. d. Q. **Algoritmos evolutivos many objectives aplicados ao problema de roteamento Multicast com qualidade de servico**. 148 p. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Santa Monima, Uberlândia, MG, Brasil, 2018.
- LAFETÁ, T. F. d. Q.; OLIVEIRA, G. M. B. d. Applying dynamic evolutionary optimization to the multiobjective knapsack problem. In: SPRINGER. **Brazilian Conference on Intelligent Systems**. [S.l.], 2020A. p. 49–63.
- LAFETÁ, T. F. de Q. et al. Many-objective evolutionary algorithms for multicast routing with quality of service problem. In: IEEE. **2016 5th Brazilian Conference on Intelligent Systems (BRACIS)**. [S.l.], 2016. p. 187–192.
- LAFETÁ, T. F. de Q.; MARTINS, L. G.; OLIVEIRA, G. M. D-meands-md: an improved evolutionary algorithm with memory and diversity strategies applied to a discrete, dynamic, and many-objective optimization problem. **The Knowledge Engineering Review**, Cambridge University Press, v. 39, p. e9, 2024.

- LAFETA, T. F. de Q.; OLIVEIRA, G. M. B. An improved version of a many-objective evolutionary algorithm based on non-dominated decomposed sets (meands-ii). In: IEEE. **2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)**. [S.l.], 2019. p. 3673–3678.
- LAFETÁ, T. F. Q.; OLIVEIRA, G. M. B. D-meands: a novel evolutionary approach to dynamic many-objective optimization problems. In: IEEE. **2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)**. [S.l.], 2020B. p. 1129–1134.
- LAUMANN, M. et al. Combining convergence and diversity in evolutionary multiobjective optimization. **Evolutionary computation**, MIT Press, v. 10, n. 3, p. 263–282, 2002.
- LI, H.; ZHANG, Q. Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. **IEEE transactions on evolutionary computation**, IEEE, v. 13, n. 2, p. 284–302, 2008.
- LI, K. et al. An evolutionary many-objective optimization algorithm based on dominance and decomposition. **IEEE transactions on evolutionary computation**, IEEE, v. 19, n. 5, p. 694–716, 2014.
- LIU, R. et al. A diversity introduction strategy based on change intensity for evolutionary dynamic multiobjective optimization. **Soft Computing**, Springer, v. 24, n. 17, p. 12789–12799, 2020.
- MALEK, M. et al. Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem. **Annals of Operations Research**, Springer, v. 21, n. 1, p. 59–84, 1989.
- _____. Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem. **Annals of Operations Research**, Springer, v. 21, n. 1, p. 59–84, 1989.
- MANKIEWICZ, R. **Story Of Mathematics**. [S.l.]: Cassell, 2000.
- MANNION, P. et al. Reward shaping for knowledge-based multi-objective multi-agent reinforcement learning. **The Knowledge Engineering Review**, Cambridge University Press, v. 33, 2018.
- MARTINS, L. G.; FRANCA, T. P.; OLIVEIRA, G. M. de. Bio-inspired algorithms for many-objective discrete optimization. In: IEEE. **2019 8th Brazilian Conference on Intelligent Systems (BRACIS)**. [S.l.], 2019. p. 515–520.
- MIETTINEN, K. **Nonlinear multiobjective optimization**. [S.l.]: Springer Science & Business Media, 1999. v. 12.
- MÜHLENBEIN, H. Parallel genetic algorithms in combinatorial optimization. In: **Computer science and operations research**. [S.l.]: Elsevier, 1992. p. 441–453.
- MÜHLENBEIN, H.; GORGES-SCHLEUTER, M.; KRÄMER, O. Evolution algorithms in combinatorial optimization. **Parallel computing**, Elsevier, v. 7, n. 1, p. 65–85, 1988.

- MURATA, T.; ISHIBUCHI, H. et al. Moga: multi-objective genetic algorithms. In: IEEE PISCATAWAY, NJ, USA. **IEEE international conference on evolutionary computation**. [S.l.], 1995. v. 1, p. 289–294.
- MURUGANANTHAM, A.; TAN, K. C.; VADAKKEPAT, P. Solving the ieece cec 2015 dynamic benchmark problems using kalman filter based dynamic multiobjective evolutionary algorithm. In: **Intelligent and evolutionary systems**. [S.l.]: Springer, 2016. p. 239–252.
- NELDER, J. A.; MEAD, R. A simplex method for function minimization. **The computer journal**, Oxford University Press, v. 7, n. 4, p. 308–313, 1965.
- PANICHELLA, A. An improved pareto front modeling algorithm for large-scale many-objective optimization. In: **Proceedings of the genetic and evolutionary computation conference**. [S.l.: s.n.], 2022. p. 565–573.
- PEREIRA, G. W. Aplicação da técnica de recozimento simulado em problemas de planejamento florestal multiobjetivo. Universidade Federal de Minas Gerais, 2004.
- POWELL, D. J.; TONG, S. S.; SKOLNICK, M. M. Engeneous domain independent, machine learning for design optimization. In: **Proceedings of the third international conference on Genetic algorithms**. [S.l.: s.n.], 1989. p. 151–159.
- RAQUEL, C.; YAO, X. Dynamic multi-objective optimization: a survey of the state-of-the-art. In: **Evolutionary computation for dynamic optimization problems**. [S.l.]: Springer, 2013. p. 85–106.
- RICHTER, H. Dynamic fitness landscape analysis. In: **Evolutionary computation for dynamic optimization problems**. [S.l.]: Springer, 2013. p. 269–297.
- SATO, H.; AGUIRRE, H. E.; TANAKA, K. Controlling dominance area of solutions and its impact on the performance of moeas. In: SPRINGER. **International conference on evolutionary multi-criterion optimization**. [S.l.], 2007. p. 5–20.
- SCHAFFER, J. D. Multiple objective optimization with vector evaluated genetic algorithms. In: LAWRENCE ERLBAUM ASSOCIATES. INC., PUBLISHERS. **Proceedings of the first international conference on genetic algorithms and their applications, 1985**. [S.l.], 1985.
- SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. **Evolutionary computation**, MIT Press, v. 2, n. 3, p. 221–248, 1994.
- STIEFEL, E. Methods of conjugate gradients for solving linear systems. **J. Res. Nat. Bur. Standards**, v. 49, p. 409–435, 1952.
- TAMAKI, H.; KITA, H.; KOBAYASHI, S. Multi-objective optimization by genetic algorithms: A review. In: IEEE. **Proceedings of IEEE international conference on evolutionary computation**. [S.l.], 1996. p. 517–522.
- TOFFOLO, A.; BENINI, E. Genetic diversity as an objective in multi-objective evolutionary algorithms. **Evolutionary computation**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 11, n. 2, p. 151–167, 2003.

- TSUTSUI, S.; FUJIMOTO, Y.; HAYASHI, I. Extended forking genetic algorithm for order representation (o-fga). In: IEEE. **Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence**. [S.l.], 1994. p. 639–644.
- TSUTSUI, S.; FUJIMOTO, Y. et al. Forking genetic algorithm with blocking and shrinking modes (fga). In: CITESEER. **ICGA**. [S.l.], 1993. p. 206–215.
- ULDER, N. L. et al. Genetic local search algorithms for the traveling salesman problem. In: SPRINGER. **International Conference on Parallel Problem Solving from Nature**. [S.l.], 1990. p. 109–116.
- VALENTINI, G. et al. Dynamic multi-objective routing algorithm: a multi-objective routing algorithm for the simple hybrid routing protocol on wireless sensor networks. **IET communications**, IET, v. 4, n. 14, p. 1732–1741, 2010.
- VINODHINI, R.; GOMATHY, C. Momhr: A dynamic multi-hop routing protocol for wsn using heuristic based multi-objective function. **Wireless Personal Communications**, Springer, v. 111, p. 883–907, 2020.
- WANG, X. et al. A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 51, n. 5, p. 757–767, 2010.
- WANG, Y.; LI, B. Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment. In: IEEE. **2009 IEEE Congress on Evolutionary Computation**. [S.l.], 2009. p. 630–637.
- ZBIGNIEW, M. Genetic algorithms+ data structures= evolution programs. **Comput Stat**, p. 372–373, 1996.
- ZHANG, Q.; LI, H. Moea/d: A multiobjective evolutionary algorithm based on decomposition. **IEEE Transactions on evolutionary computation**, IEEE, v. 11, n. 6, p. 712–731, 2007.
- ZHANG, Z.; QIAN, S. Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems. **Soft Computing**, Springer, v. 15, n. 7, p. 1333–1349, 2011.
- ZHENG, B. A new dynamic multi-objective optimization evolutionary algorithm. In: IEEE. **Third international conference on natural computation (ICNC 2007)**. [S.l.], 2007. v. 5, p. 565–570.
- ZITZLER, E. **Evolutionary algorithms for multiobjective optimization: Methods and applications**. [S.l.]: Citeseer, 1999. v. 63.
- ZITZLER, E.; LAUMANN, M.; THIELE, L. Spea2: Improving the strength pareto evolutionary algorithm. **TIK-report**, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische ..., v. 103, 2001.
- ZITZLER, E.; THIELE, L. Multiobjective optimization using evolutionary algorithms—a comparative case study. In: SPRINGER. **International conference on parallel problem solving from nature**. [S.l.], 1998. p. 292–301.

_____. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. **IEEE transactions on Evolutionary Computation**, IEEE, v. 3, n. 4, p. 257–271, 1999.

Apêndices

Experimentos Preliminares com os Algoritmos D-MEANDS-II e D-MEANDS-III

A.1 Estudo da Quantidade de Subpopulações do D-MEANDS-II

A proposta de limitar a quantidade de subpopulações surgiu a partir da seguinte regra: utilizar apenas subpopulações que agrupam 2 objetivos. Logo em seguida, surgiu a seguinte dúvida: por que subpopulações com 3 objetivos não poderiam apresentar desempenho superior? Quanto maior o número de objetivos agrupados por uma subpopulação, maior tende a ser sua fronteira de Pareto; dessa forma, maior também é a variedade de informações que pode ser explorada no processo evolutivo. Com base nessa observação, elaboramos um experimento para comparar o D-MEANDS-II, que gera todas as subpopulações produzidas pelo MEANDS-II, com versões que geram apenas grupos específicos de subpopulações. Denominamos D-MEANDS-II-S2 a versão em que são geradas apenas subpopulações de 2 objetivos, D-MEANDS-II-S3 a versão em que são geradas apenas subpopulações de 3 objetivos, D-MEANDS-II-S4 a versão em que são geradas apenas subpopulações de 4 objetivos, e assim sucessivamente.

Neste experimento, utilizamos 9 instâncias do DMKP, divididas de acordo com o número de itens e de objetivos. Em todas as 9 instâncias, o número de mudanças de ambiente (EC) é igual a 20. No presente trabalho, utilizamos instâncias com 30, 50 e 100 itens, combinadas com 4, 6 e 8 objetivos. Foram utilizados os seguintes parâmetros para o D-MEANDS-II e suas variações: 50.000 gerações, taxa de mutação de 10% e 1.000 indivíduos para inicialização da população. Cada algoritmo foi executado 100 vezes para cada instância. Os gráficos apresentam a média e o desvio padrão dessas 100 execuções, considerando as métricas não paramétricas *Density* (AD) (ZHANG; QIAN, 2011),

Hipervolume (HV) (ZITZLER; THIELE, 1998), descritas no Capítulo 2, e o tempo de execução.

Os resultados apresentados neste experimento foram divididos de acordo com o número de objetivos das instâncias:

1. Para as instâncias com 4 objetivos, comparamos o D-MEANDS-II com o D-MEANDS-II-S2 e o D-MEANDS-II-S3.
2. Para as instâncias com 6 objetivos, comparamos o D-MEANDS-II com o D-MEANDS-II-S2, o D-MEANDS-II-S3, o D-MEANDS-II-S4 e o D-MEANDS-II-S5.
3. Para as instâncias com 8 objetivos, comparamos o D-MEANDS-II com o D-MEANDS-II-S2, o D-MEANDS-II-S3, o D-MEANDS-II-S4, o D-MEANDS-II-S5, o D-MEANDS-II-S6 e o D-MEANDS-II-S7.

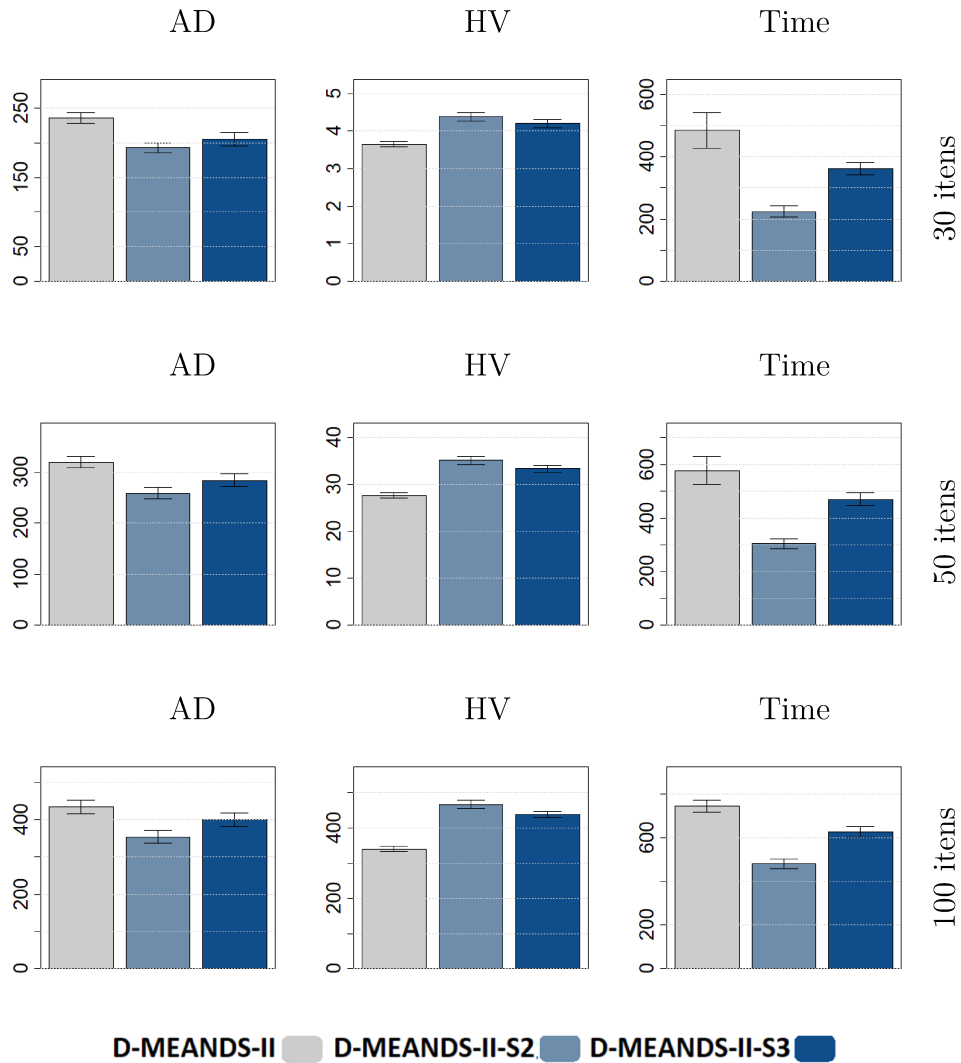


Figura 30 – Comparação de desempenho do D-MEANDS-II e suas variantes para instâncias do DMKP com 4 objetivos (EC=20).

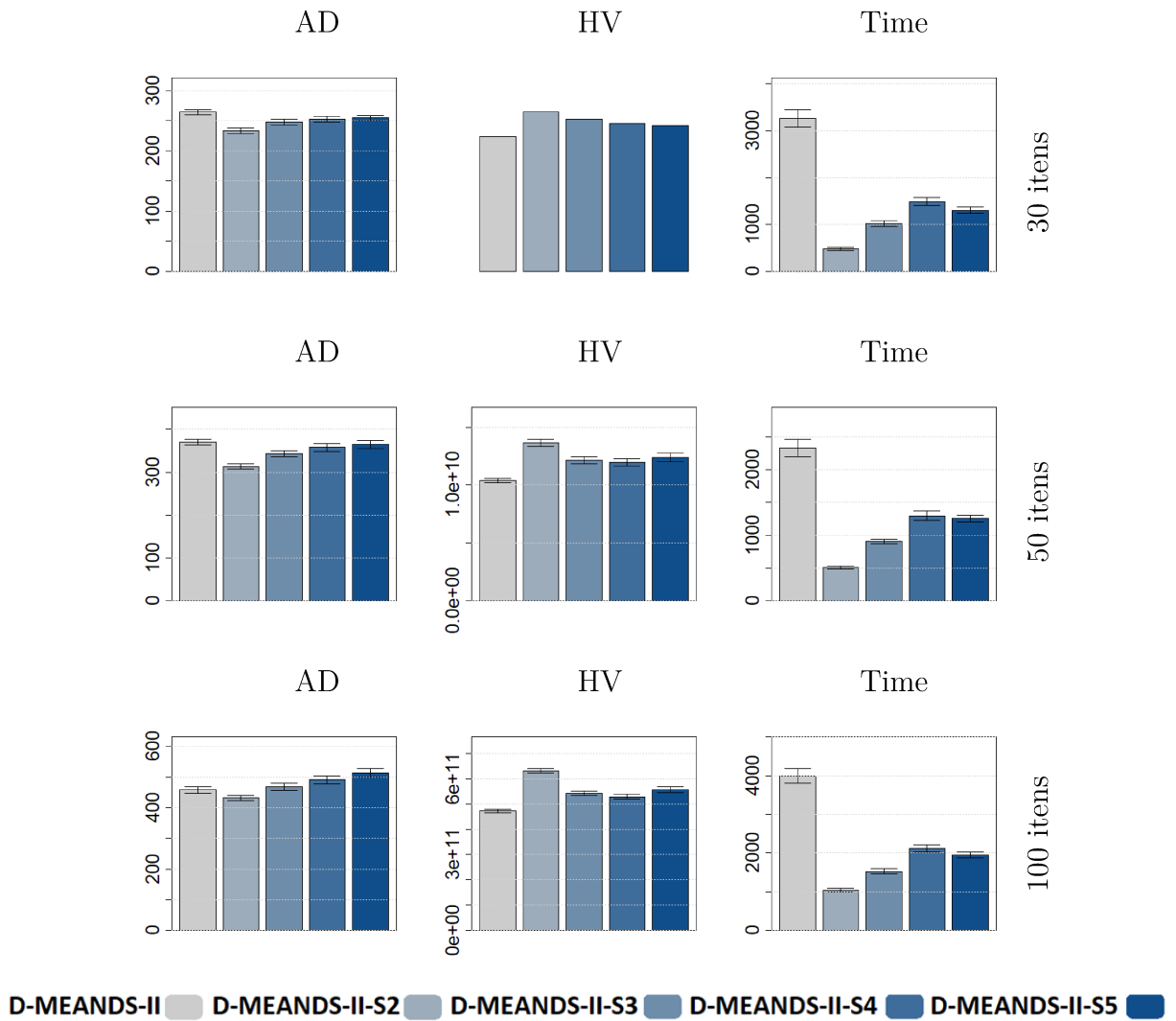


Figura 31 – Comparação de desempenho do D-MEANDS-II e suas variantes para instâncias do DMKP com 6 objetivos (EC=20).

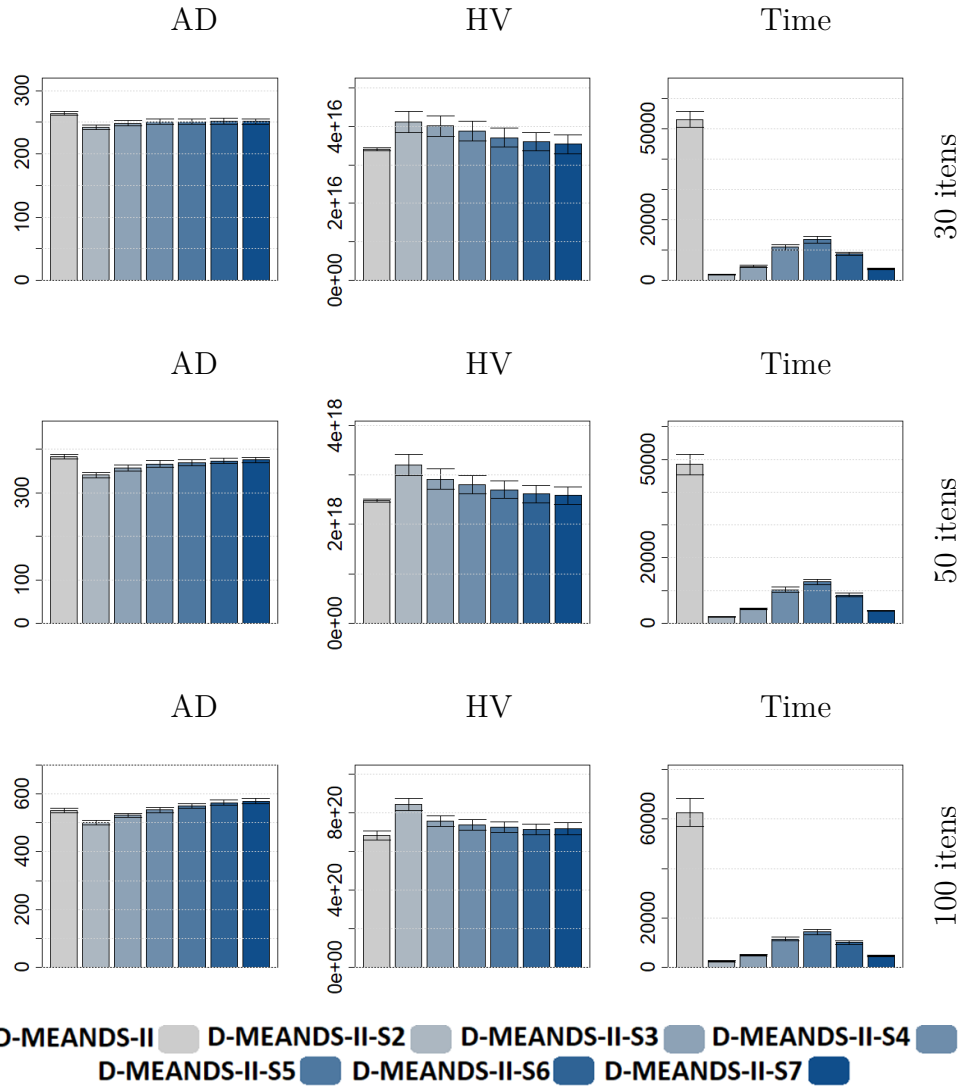


Figura 32 – Comparação de desempenho do D-MEANDS-II e suas variantes para instâncias do DMKP com 8 objetivos (EC=20).

A.2 Tour4Ind: Uma Nova Dinâmica para o Cruzamento

No D-MEANDS-II-S2 e em suas versões anteriores, o *crossover* seguia quatro passos:

i) Selecionar uma subpopulação usando o método de torneio quádruplo baseado no grau de convergência; ii) Selecionar outra subpopulação usando o método de torneio quádruplo, também baseado no grau de convergência das subpopulações. Essa não pode ser a mesma subpopulação selecionada anteriormente; iii) Selecionar aleatoriamente um indivíduo pai de cada subpopulação; iv) Efetuar o *crossover* com esses dois pais.

- i. Selecionar as 2 subpopulações (distintas entre si) que participarão do cruzamento, usando o *contribution score*;
- ii. Selecionar aleatoriamente um indivíduo pai de cada subpopulação;
- iii. Efetuar o *crossover* com esses dois pais.

Podemos observar que o algoritmo investe em uma estratégia mais complexa para selecionar as subpopulações do que para selecionar os indivíduos pais dentro dessas subpopulações. Propomos alterar o passo "iii" para uma metodologia mais elaborada, na qual, após a seleção das subpopulações, são escolhidos os indivíduos pais que se encontram mais próximos do centro de cada subpopulação.

Para verificar o impacto dessa mudança, elaboramos um experimento. Nesse experimento, comparamos o D-MEANDS-II-S2 com a proposta que utiliza o Tour4Ind, denominada D-MEANDS-II-T4I. Foram utilizadas 9 instâncias do DMKP, divididas de acordo com o número de itens e de objetivos. Em todas as 9 instâncias, o número de mudanças de ambiente (EC) é igual a 20. No presente trabalho, utilizamos instâncias com 30, 50 e 100 itens, combinadas com 4, 6 e 8 objetivos. Foram utilizados os seguintes parâmetros para o D-MEANDS-II-S2 e sua variação: 50.000 gerações, taxa de mutação de 10% e 1.000 indivíduos para inicialização da população. Cada algoritmo foi executado 100 vezes para cada instância. Os gráficos apresentam a média e o desvio padrão dessas 100 execuções, considerando as métricas não paramétricas *Density* (AD) (ZHANG; QIAN, 2011) e Hipervolume (HV) (ZITZLER; THIELE, 1998), descritas no Capítulo 2, além do tempo de execução.

Observando a Figura 33, vemos os resultados das instâncias com 4 objetivos. É fácil notar que o D-MEANDS-II-T4I apresentou melhora em todas as instâncias nas métricas HV e AD, principalmente na métrica HV. No entanto, seu tempo de execução foi pior do que o do D-MEANDS-II-S2. Concluímos, portanto, que a estratégia Tour4Ind melhora os resultados das métricas ao custo de um aumento no tempo de execução. Nas instâncias com 6 e 8 objetivos, apresentadas nas Figuras 34 e 35, observamos o mesmo

comportamento verificado nas instâncias com 4 objetivos. O D-MEANDS-II-T4I apresentou melhora nas métricas, mas obteve desempenho inferior em tempo de execução quando comparado ao D-MEANDS-II-S2. A única exceção foi a instância com 100 itens e 8 objetivos, na qual o tempo de execução do D-MEANDS-II-T4I foi melhor. A diferença no tempo de execução entre os dois algoritmos não é discrepante. Considerando que a estratégia Tour4Ind melhorou os resultados das métricas, ela se apresenta como um bom acréscimo ao D-MEANDS-II-S2.

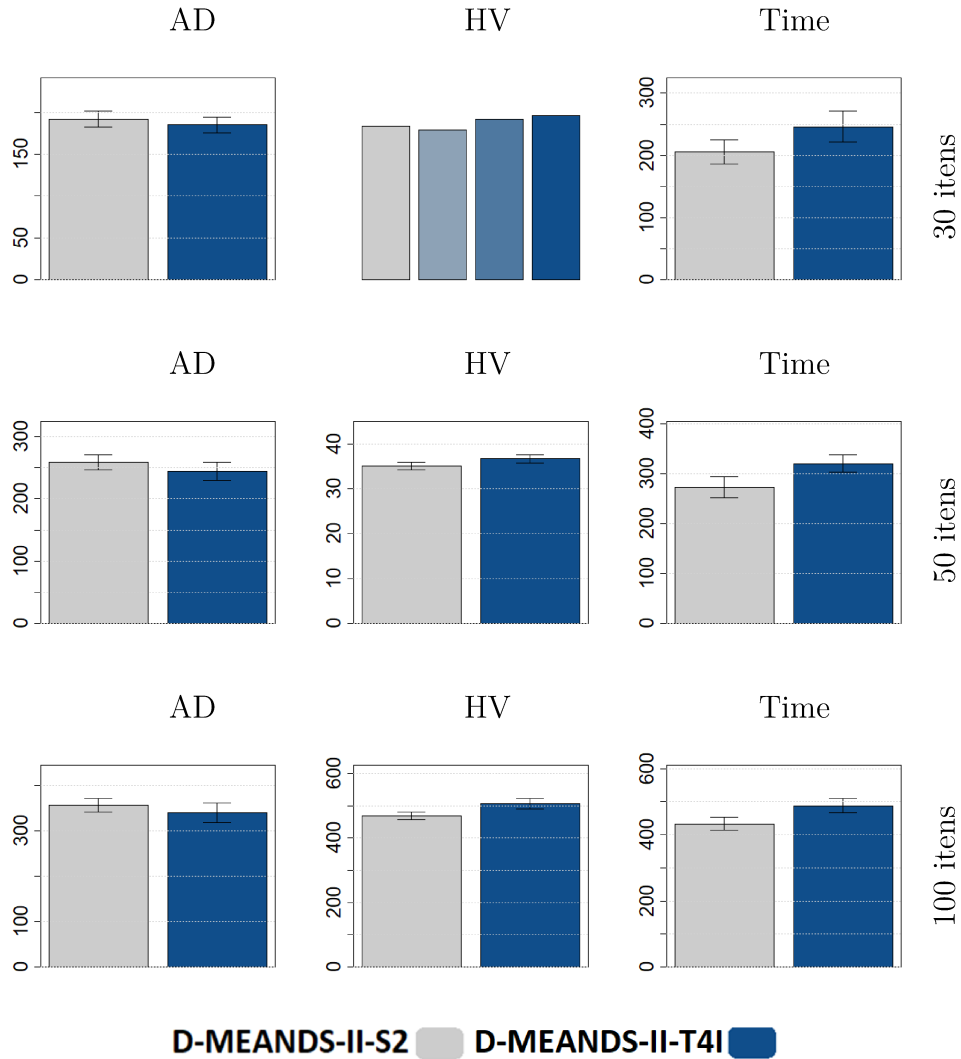


Figura 33 – Comparação de desempenho do D-MEANDS-II-S2 com D-MEANDS-II-T4I para instâncias do DMKP com 4 objetivos (EC=20).

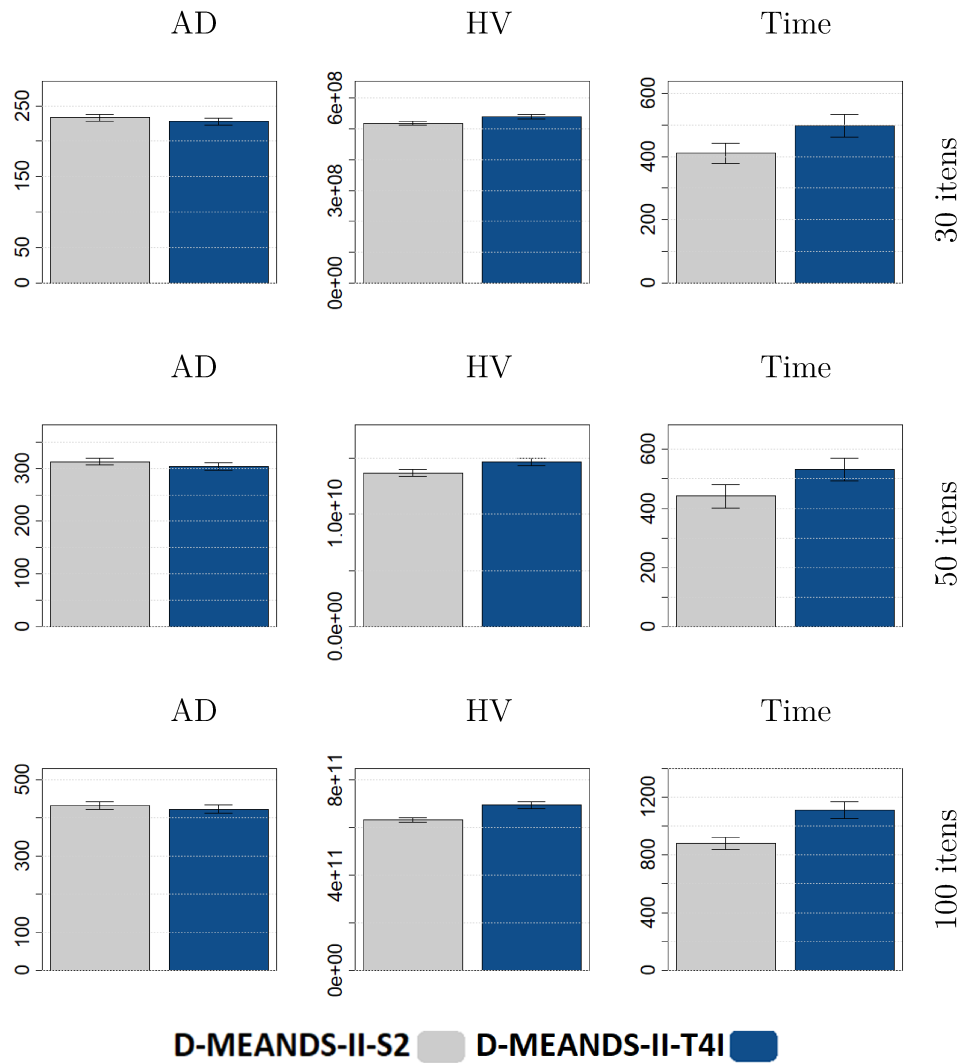


Figura 34 – Comparação de desempenho do D-MEANDS-II-S2 com D-MEANDS-II-T4I para instâncias do DMKP com 6 objetivos (EC=20).

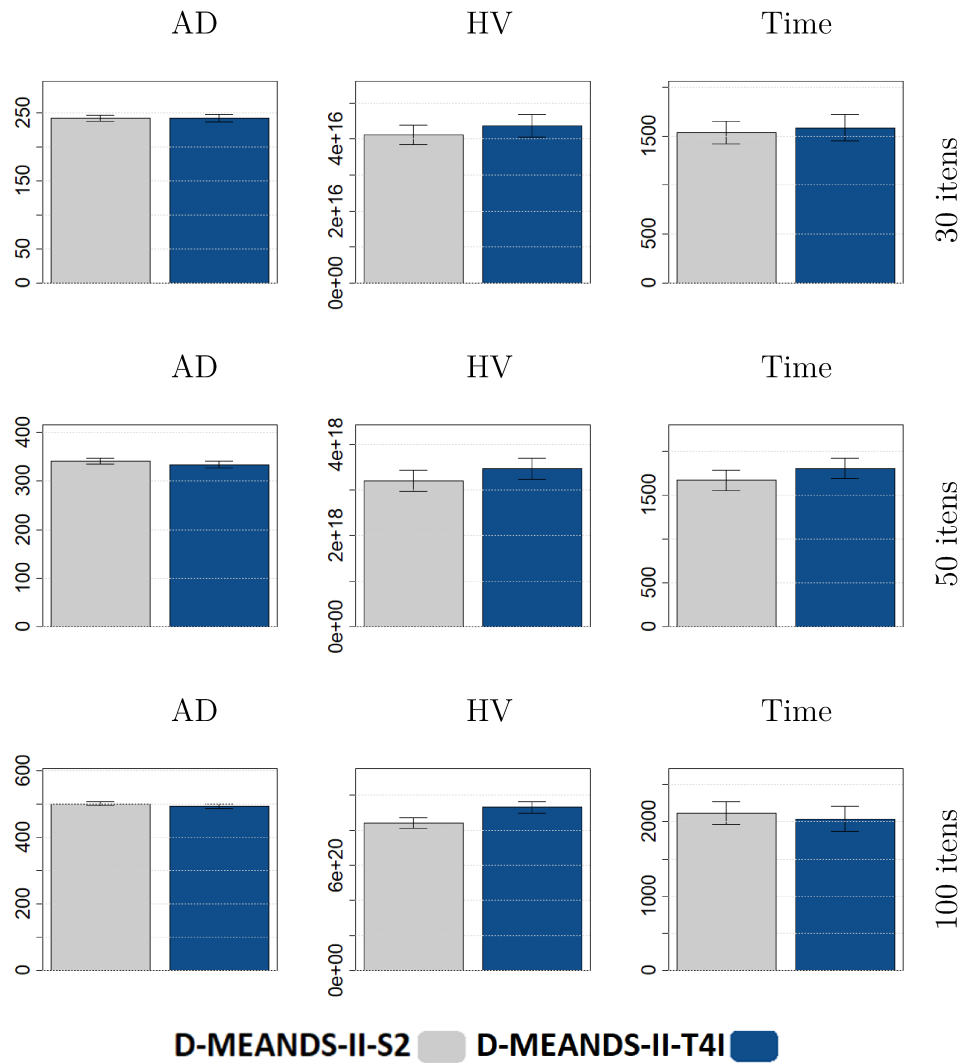


Figura 35 – Comparação de desempenho do D-MEANDS-II-S2 com D-MEANDS-II-T4I para instâncias do DMKP com 8 objetivos (EC=20).

A.3 Filtro-PR: Novo Método de Mudança de Ambiente

A mudança de ambiente do D-MEANDS e de suas versões propostas neste trabalho utiliza um método baseado em memória (descrito na Seção 3.3). A cada mudança de ambiente, a população é reconstruída. Estratégias baseadas em memória resgatam informações do ambiente anterior para reconstruir a população no novo ambiente. No caso do D-MEANDS e de suas demais versões, são resgatados os indivíduos armazenados no arquivo externo para essa reconstrução, sendo que esse arquivo armazena todas as soluções da primeira fronteira de Pareto do problema.

O novo método de mudança de ambiente foi implementado no D-MEANDS-II-S2 com o objetivo de filtrar as soluções do arquivo externo que serão introduzidas no novo ambiente. O intuito é resgatar soluções que estejam centralizadas ou localizadas nas extremidades do arquivo externo, considerando seus respectivos posicionamentos no espaço de objetivos. Dessa forma, busca-se fortalecer a diversidade da busca. Vale lembrar que as subpopulações (conjuntos) do D-MEANDS-II-S2 armazenam apenas soluções não dominadas; por isso, algumas soluções são descartadas na tentativa de inserção nessas subpopulações. Assim, mesmo que uma solução passe pelo filtro, não há garantia de que ela será mantida na população.

Para verificar o impacto dessa mudança, elaboramos um experimento. Nesse experimento, comparamos o D-MEANDS-II-S2 com a proposta que utiliza o *Filtro – PR*, denominada D-MEANDS-II-PR. Foram utilizadas 9 instâncias do DMKP, divididas de acordo com o número de itens e de objetivos. Em todas as 9 instâncias, o número de mudanças de ambiente (EC) é igual a 20. No presente trabalho, utilizamos instâncias com 30, 50 e 100 itens, combinadas com 4, 6 e 8 objetivos. Foram utilizados os seguintes parâmetros para o D-MEANDS-II-S2 e sua variação: 50.000 gerações, taxa de mutação de 10% e 1.000 indivíduos para inicialização da população. Cada algoritmo foi executado 100 vezes para cada instância. Os gráficos apresentam a média e o desvio padrão dessas 100 execuções, considerando as métricas não paramétricas *Density* (AD) (ZHANG; QIAN, 2011) e Hipervolume (HV) (ZITZLER; THIELE, 1998), descritas no Capítulo 2, além do tempo de execução.

A Figura 36 mostra os resultados comparativos entre o D-MEANDS-II-S2 e o D-MEANDS-II-PR utilizando as instâncias com 4 objetivos. Podemos observar que os valores das métricas são bastante próximos em todas as instâncias; no entanto, o D-MEANDS-II-PR apresentou um tempo de execução melhor que o do D-MEANDS-II-S2. Nas instâncias com 6 e 8 objetivos, apresentadas nas Figuras 37 e 38, o comportamento se mantém, com valores das métricas muito próximos e tempo de execução do D-MEANDS-II-PR inferior ao do D-MEANDS-II-S2. Vale observar que a diferença no tempo de execução entre os dois algoritmos, na instância mais complexa (100 itens e 8 objetivos), foi

menor do que nas demais instâncias.

Durante a mudança de ambiente, o D-MEANDS-II-S2 sempre tenta inserir 1.000 soluções no novo ambiente, enquanto o D-MEANDS-II-PR insere apenas as soluções que passam pelo filtro. Por essa razão, seu tempo de execução tende a ser menor, já que o D-MEANDS-II-PR tende a selecionar menos de 1.000 soluções. No entanto, vale lembrar que, nas instâncias com 8 objetivos, o D-MEANDS-II-PR chega a selecionar aproximadamente 1.000 soluções e ainda assim mantém um tempo de execução menor que o do D-MEANDS-II-S2. Visto que os valores das métricas não apresentam grande diferença entre os dois algoritmos, podemos concluir que o novo método de mudança de ambiente apresentou uma melhora significativa no tempo de execução e se adaptou bem à dinâmica do D-MEANDS-II-S2.

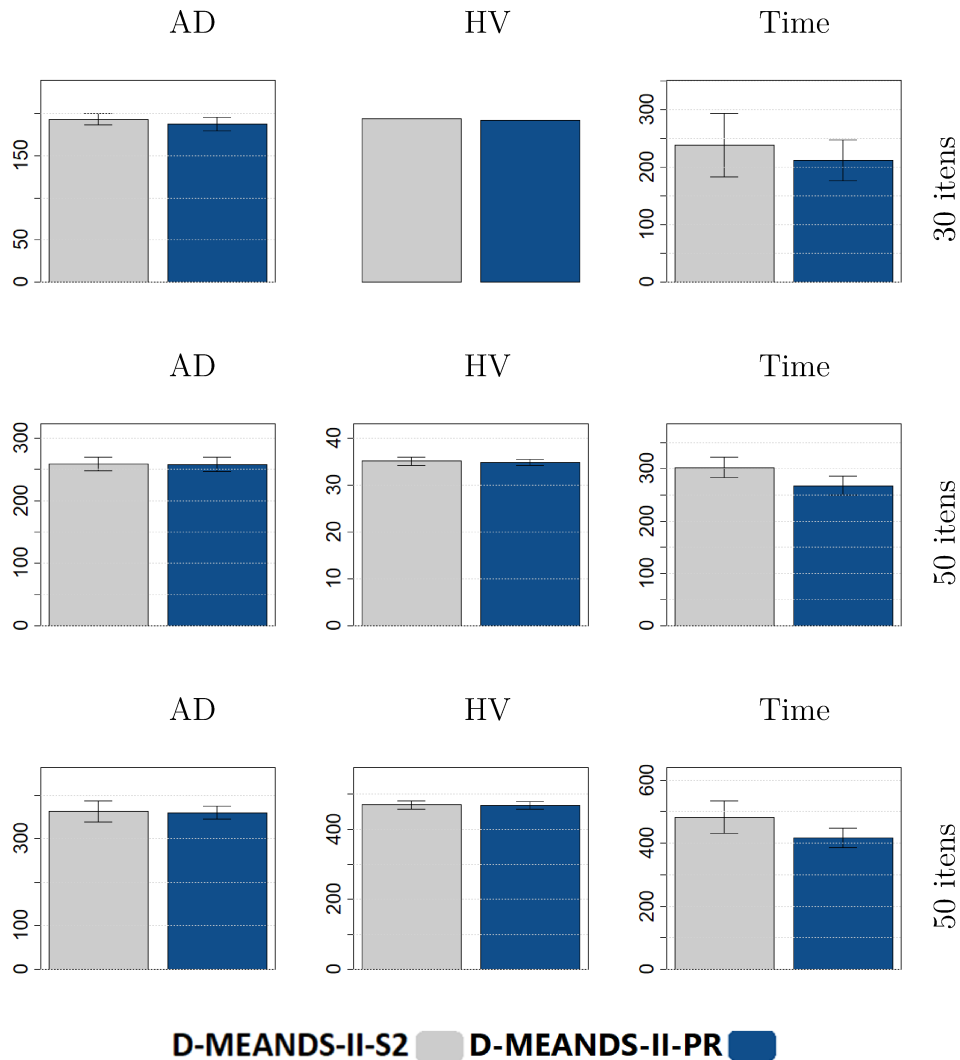


Figura 36 – Comparação de desempenho do D-MEANDS-II-S2 com D-MEANDS-II-PR para instâncias do DMKP com 4 objetivos (EC=20).

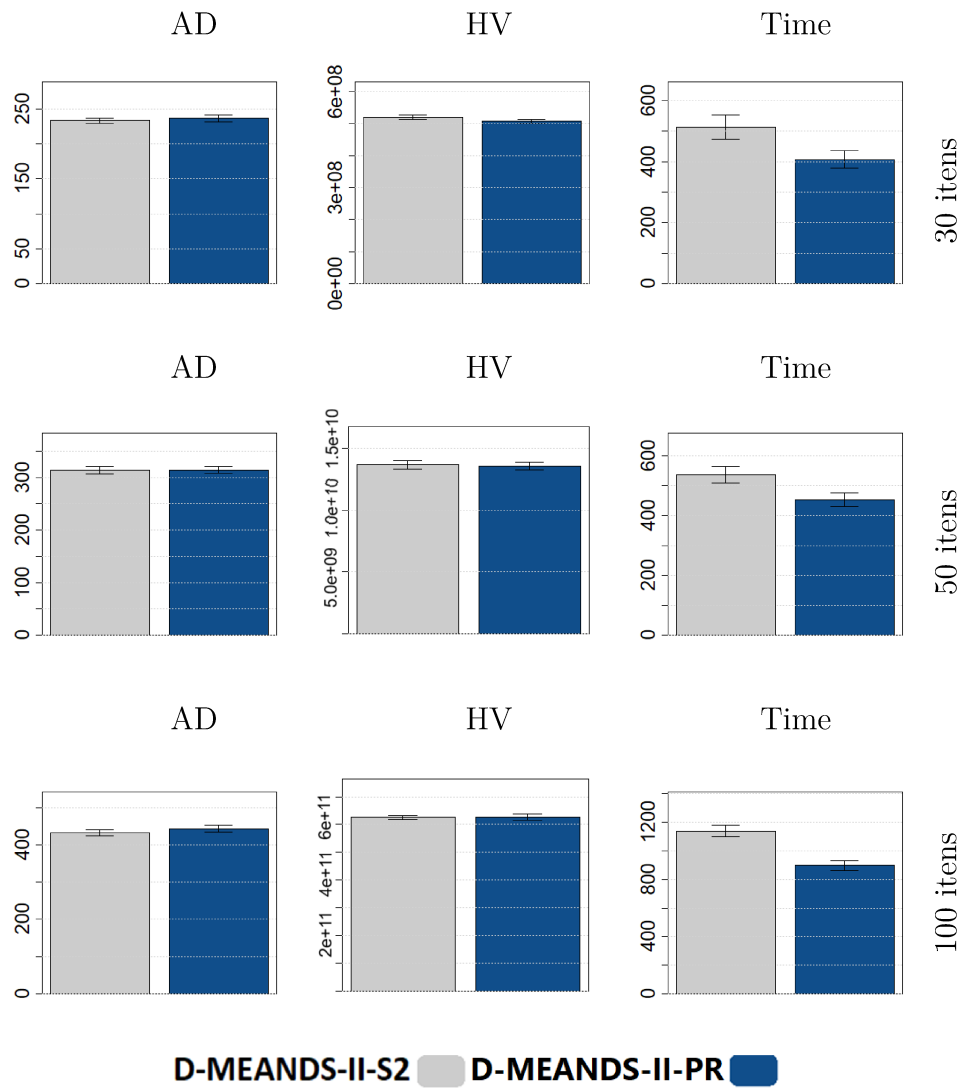


Figura 37 – Comparação de desempenho do D-MEANDS-II-S2 com D-MEANDS-II-PR para instâncias do DMKP com 6 objetivos (EC=20).

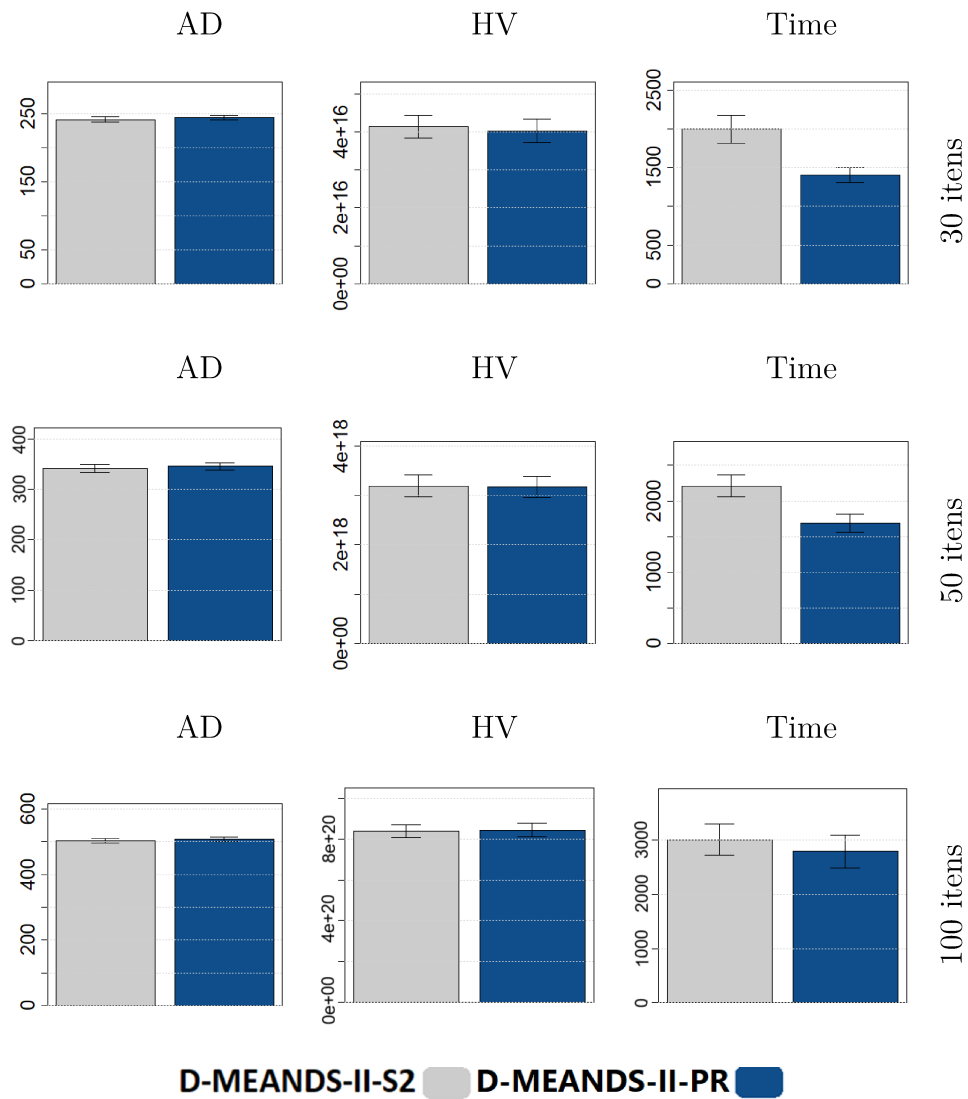


Figura 38 – Comparação de desempenho do D-MEANDS-II-S2 com D-MEANDS-II-PR para instâncias do DMKP com 8 objetivos (EC=20).

Dados Complementares dos Experimentos

Esta seção apresenta com mais detalhes os resultados alcançados pelos algoritmos evolutivos para as diferentes instâncias investigadas nos experimentos realizados nesta pesquisa. Os gráficos ilustram a média e o desvio padrão de cada métrica, considerando 100 execuções para cada algoritmo avaliado. Os parâmetros adotados em cada algoritmo, bem como a descrição das instâncias do problema utilizado (DMKP) encontram-se descritos na Seção 4.

B.1 Fase 1: Análise de Desempenho de Métodos Evolutivos da Literatura

Nesta seção, procede-se à avaliação e comparação do desempenho de distintos algoritmos evolutivos reportados na literatura, empregando-se as métricas *Mean Inverted Generational Distance* ($\overline{IGD}$) (WANG; LI, 2009) e *Hypervolume Ratio* (*HVR*) (ZITZLER, 1999), conforme descritas no Capítulo 2. Cada algoritmo considerado adota uma estratégia dinâmica própria. Destaca-se que todas as implementações foram desenvolvidas na mesma linguagem de programação e executadas sob condições computacionais equivalentes. Os algoritmos analisados compreendem: DNSGA-II, DNSGA-III, MOEA/D-KF e MS-MOEA.

B.1.1 Análise Comparativa dos Métodos da Literatura em Cenários com uma Mudança de Ambiente

Primeiramente, analisaremos os resultados das métricas utilizando instâncias em que a mudança de ambiente ocorre uma única vez, na geração 51 (metade da execução do AG). A Figura 39 apresenta os resultados das instâncias com 30 itens. Nas instâncias

com $m = 4, 6$ e 8 , o MS-MOEA superou todos os demais algoritmos em ambas as métricas multiobjetivo (HVR e IGD). À medida que o número de objetivos aumenta, a superioridade do MS-MOEA torna-se mais evidente. O DNSGA-II alcançou o segundo melhor desempenho, superando inclusive o DNSGA-III, cujo a formulação estática (NSGA-III) foi proposta para problemas de muitos objetivos. O MOEA/D-KF, por sua vez, apresenta os piores valores para as métricas multiobjetivo, embora seu desempenho em HVR melhore à medida que o número de objetivos aumenta. A Figura 40 apresenta os resultados das instâncias com 50 itens e apenas uma mudança de ambiente. Pode-se observar que o comportamento não muda com o aumento no número de itens, ou seja, a dinâmica de desempenho apresentada pelos algoritmos é muito similar àquela descrita para as instâncias com 30 itens.

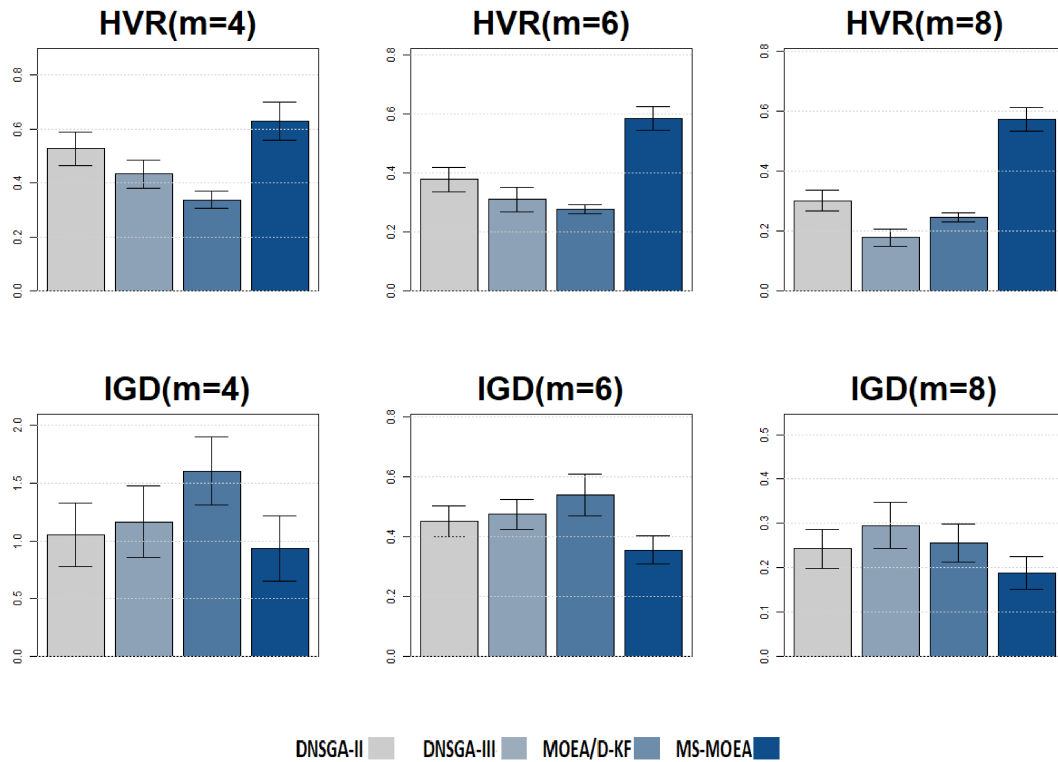


Figura 39 – Resultados HVR e $\overline{IGD}$ para instâncias do DMKP com 30 itens ($EC=1$).

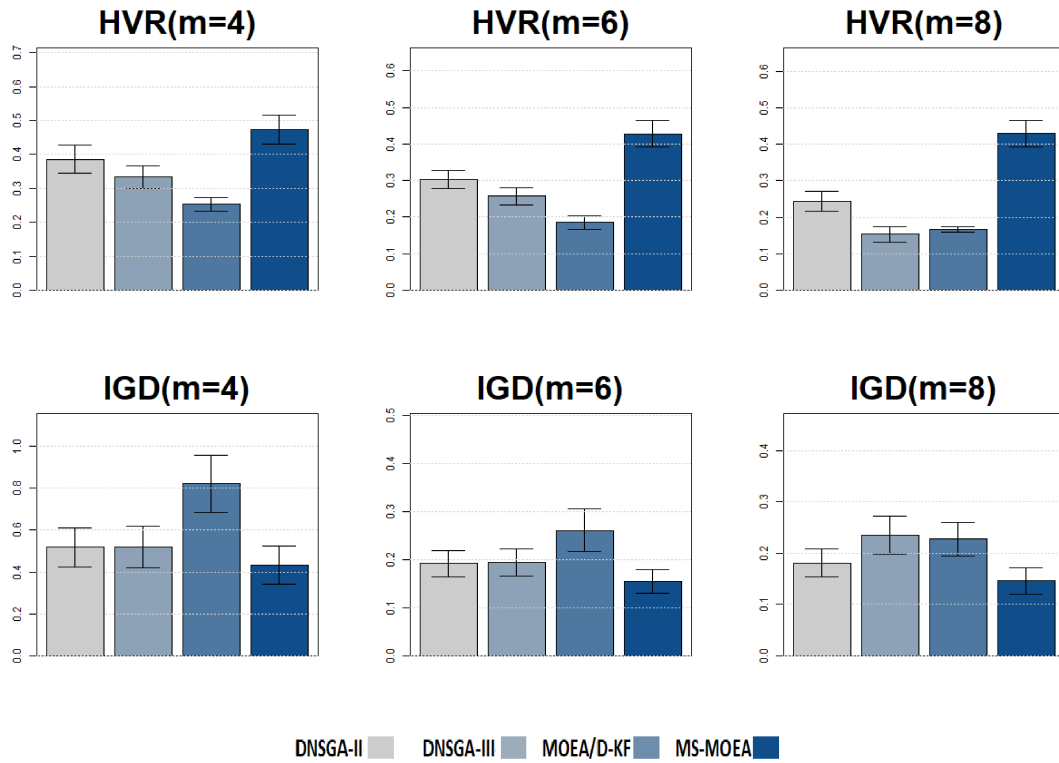


Figura 40 – Resultados HVR e $\overline{IGD}$ para instâncias do DMKP com 50 itens ($EC=1$).

B.1.2 Análise Comparativa de Métodos da Literatura em Cenários com duas Mudanças de Ambiente

Nesta seção, serão analisados os resultados dos experimentos utilizando instâncias do DMKP em que o ambiente realiza duas mudanças ao longo da execução dos MOEAs. Essas mudanças ocorrem nas gerações 34 e 67, ou seja, a cada um terço da execução. As Figuras 41 e 42 apresentam os resultados das instâncias com 30 e 50 itens, respectivamente. A análise comparativa dos quatro algoritmos é muito semelhante àquela observada nos experimentos com $EC = 1$.

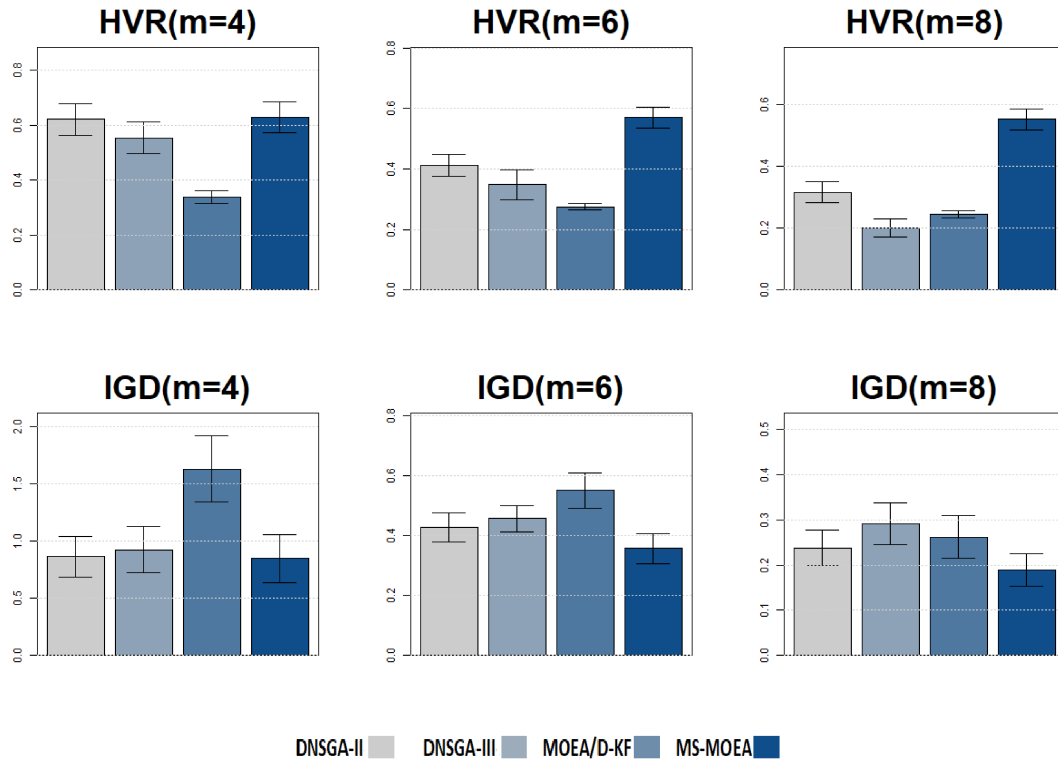


Figura 41 – Resultados HVR e $\overline{IGD}$ para instâncias do DMKP com 30 itens ($EC=2$).

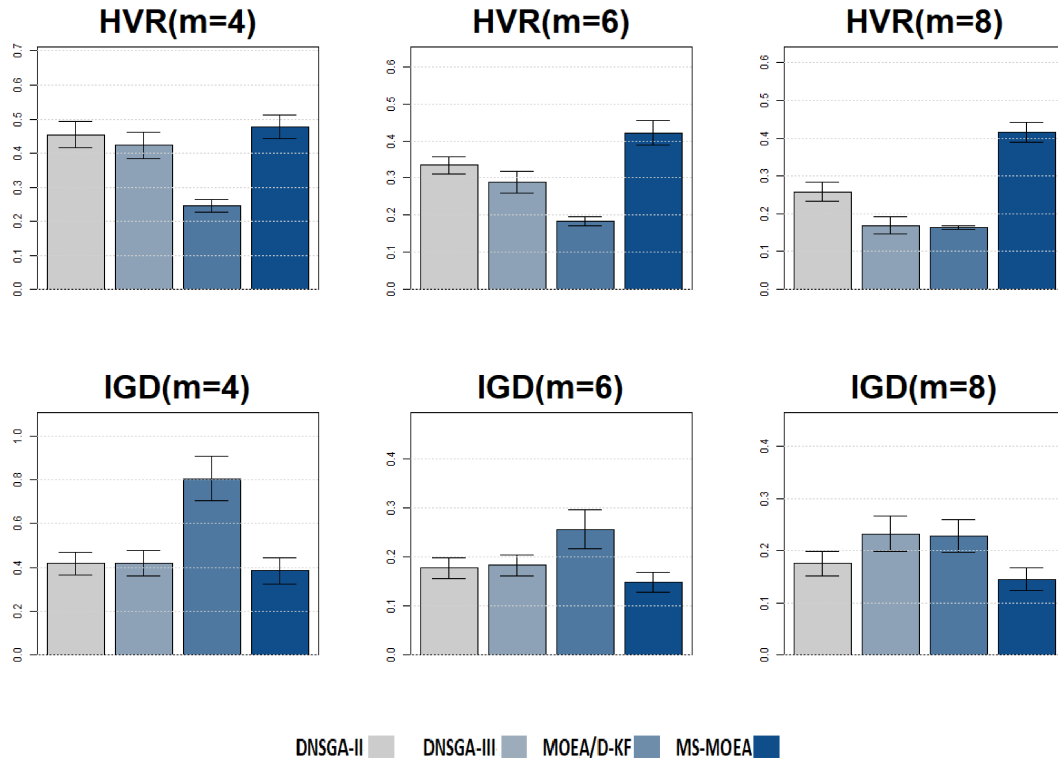


Figura 42 – Resultados HVR e $\overline{IGD}$ para instâncias do DMKP com 50 itens ($EC=2$).

B.2 Fase 2: Análise de Desempenho do Algoritmo D-MEANDS

Com base na análise dos algoritmos DNSGA-II, DNSGA-II* e MS-MOEA, foi proposto o D-MEANDS (LAFETÁ; OLIVEIRA, 2020B), um algoritmo evolutivo multiobjetivo que incorpora estratégias dinâmicas ao algoritmo MEANDS (LAFETA et al., 2018), originalmente desenvolvido para problemas estáticos. O método foi aplicado ao DMKP com o objetivo de avaliar seu desempenho em problemas de otimização multiobjetivo dinâmica e compará-lo aos algoritmos de referência DNSGA-II e MS-MOEA.

B.2.1 Comparação dos Algoritmos Considerando um Única Mudança de Ambiente

Inicialmente, foi analisado o desempenho dos MOEAs em cenários com uma única mudança de ambiente. A Figura 43 apresenta os resultados das métricas HVR e $\overline{IGD}$ para instâncias com 30 itens e 4 ou 6 objetivos. O MS-MOEA se mostra superior em relação ao DNSGA-II, considerando as métricas. Enquanto o D-MEANDS apresentou desempenho competitivo e menor desvio padrão comparado ao MS-MOEA. Para as instâncias com 50 itens (Figura 44), o MS-MOEA obteve os melhores resultados para $m = 4$, ao passo que o D-MEANDS se tornou mais competitivo com o aumento do número de objetivos.

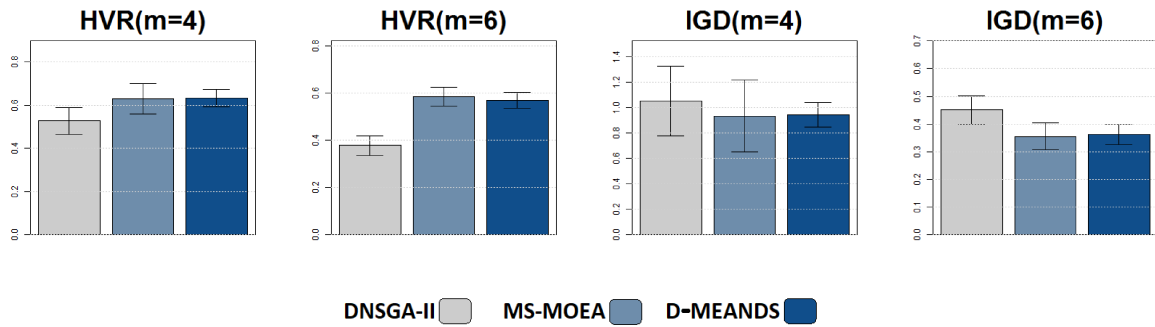


Figura 43 – Resultados de HVR e $\overline{IGD}$ para instâncias DMKP com 30 itens, objetivos $m = 4$ e 6 ($EC=1$). Avaliando o D-MEANDS.

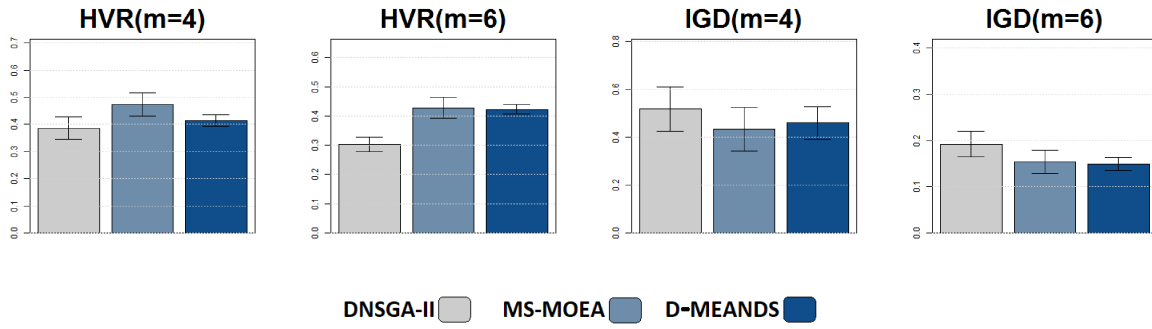


Figura 44 – Resultados de HVR e $\overline{IGD}$ para instâncias DMKP com 50 itens, objetivos $m = 4$ e 6 ($EC=1$). Avaliando o D-MEANDS.

B.2.2 Comparação dos Algoritmos Considerando Duas Mudança de Ambiente

Nos experimentos com duas alterações de ambiente, as modificações foram introduzidas em dois momentos distintos da execução, correspondentes a $1/3$ e $2/3$ do total de gerações previstas. Dessa forma, para os algoritmos DNSGA-II e MS-MOEA, as mudanças ocorreram nas gerações 33 e 66, respectivamente, enquanto, para o D-MEANDS, foram aplicadas nas gerações 3.333 e 6.666.

A Figura 45 apresenta os resultados das métricas HVR e $\overline{IGD}$ para instâncias com 30 itens. A Figura 46 apresenta os resultados para as instâncias com 50 itens. As conclusões são semelhantes às obtidas no experimento anterior (com $EC = 1$), embora o DNSGA-II se mostre mais competitivo em relação ao MS-MOEA nas instâncias com 4 objetivos. De forma geral, os algoritmos MS-MOEA e D-MEANDS superam o DNSGA-II, exceto na instância com 50 itens e $m = 4$, considerando a métrica HVR .

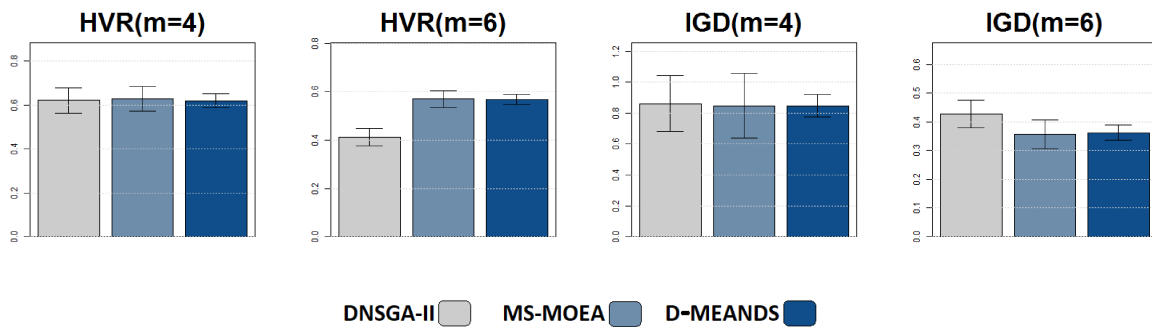


Figura 45 – Resultados de HVR e $\overline{IGD}$ para instâncias DMKP com 30 itens, objetivos $m = 4$ e 6 ($EC=2$). Avaliando o D-MEANDS.

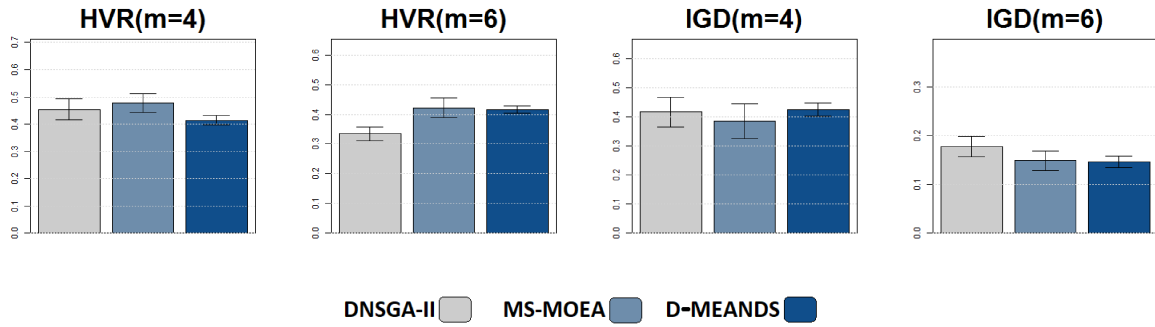


Figura 46 – Resultados de HVR e $\overline{IGD}$ para instâncias DMKP com 50 itens, objetivos $m = 4$ e 6 ($EC=2$). Avaliando o D-MEANDS.

B.2.3 Comparação dos Algoritmos Considerando Cinco e Dez Mudanças de Ambiente

Aqui os três MOEAs foram avaliados em cenários caracterizados por maior dinamismo ambiental, considerando configurações com cinco ($EC = 5$) e dez ($EC = 10$) alterações ao longo da execução do algoritmo genético. Em virtude do elevado custo computacional associado aos experimentos, a análise foi restrita às instâncias com 4 objetivos.

Ao analisar os resultados com $EC = 5$, apresentados na Figura 47, pode-se observar que o DNSGA-II e o D-MEANDS melhoraram seu desempenho em ambas as métricas, HVR e $\overline{IGD}$, quando comparados ao MS-MOEA. Além disso, nesses experimentos conduzidos em um ambiente mais dinâmico, o D-MEANDS superou os outros dois algoritmos em ambas as métricas. A análise da Figura 48, que apresenta os resultados com $EC = 10$, indica que os algoritmos mantêm o mesmo comportamento observado nos experimentos com $EC = 5$.

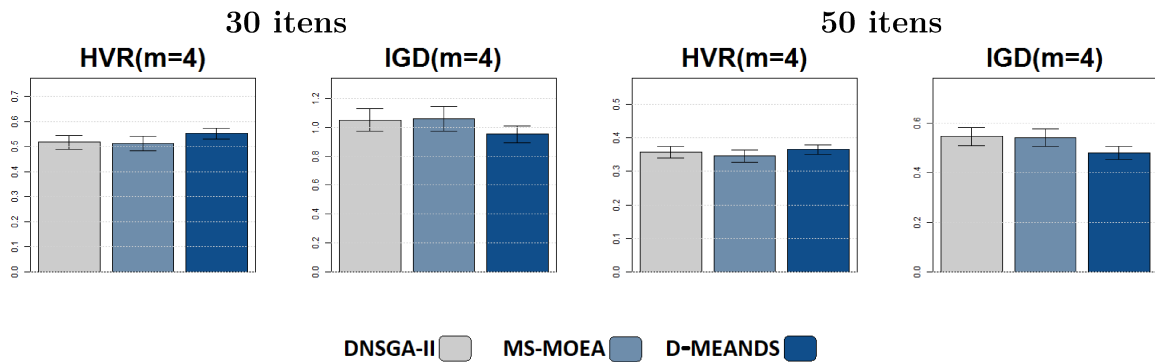


Figura 47 – Resultados de HVR e $\overline{IGD}$ para instâncias DMKP com 30 e 50 itens, 4 objetivos ($m=4$) e 5 mudanças de ambiente ($EC=5$).

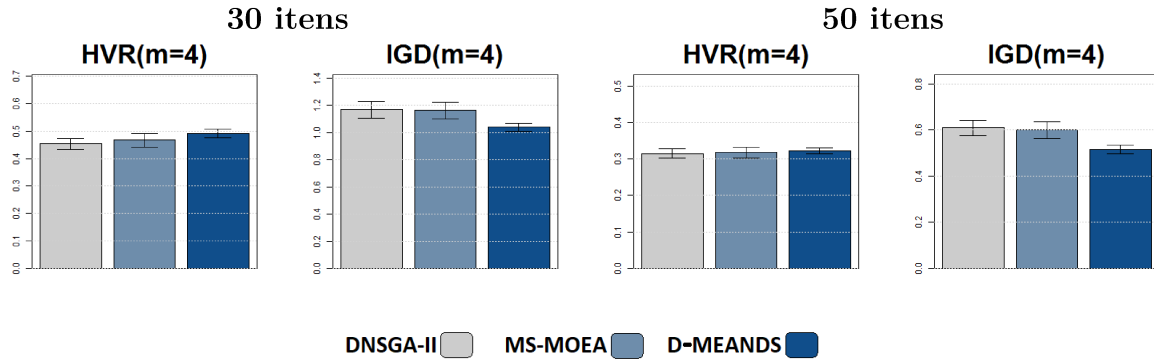


Figura 48 – Resultados de HVR e $\overline{IGD}$ para instâncias DMKP com 30 e 50 itens, 4 objetivos ($m=4$) e 10 mudanças de ambiente ($EC=10$).

B.2.4 Comparação entre os Algoritmos Considerando Muitas Mudanças de Ambiente

Nesta parte do experimento é realizada uma análise do desempenho dos algoritmos DNSGA-II*, MS-MOEA e D-MEANDS em cenários com diferentes quantidades de mudanças de ambiente ($EC = 10, 15$ e 20). Adicionalmente, foi incorporado à avaliação o algoritmo *MOEA/D based on Differential Evolution and Dynamic Diversity Introduction Strategy* (DDIS-MOEA/D-DE) (LIU et al., 2020), proposto recentemente na literatura.

As Figuras 49, 50 e 51 apresentam os resultados da métrica AD para os 27 novos cenários analisados. Como pode ser observado, o D-MEANDS apresenta os melhores valores em quase todas as instâncias. As exceções são três cenários com 30 itens: um envolvendo 6 objetivos ($EC = 15$) e dois com 8 objetivos ($EC = 15$ e $EC = 20$). Mesmo nesses três cenários, o D-MEANDS encontrou resultados próximos aos do MS-MOEA, que alcançou os menores (melhores) valores de AD. Ao comparar o MS-MOEA e o DNSGA-II*, observa-se que o segundo supera o primeiro em oito cenários; no entanto, o MS-MOEA apresenta melhor desempenho do que o DNSGA-II* na maioria dos 27 cenários. O DDIS-MOEA/D-DE apresentou o pior desempenho em todas as instâncias analisadas. Com base nos resultados de AD, é possível concluir que o D-MEANDS encontra, em geral, conjuntos de soluções melhor distribuídos (isto é, com menor densidade).

As Figuras 52, 53 e 54 apresentam os resultados da métrica HV. Analisando os gráficos, observa-se que, em 9 das 27 instâncias, o DNSGA-II* supera os demais algoritmos. Esses cenários envolvem 4 ou 6 objetivos. No entanto, na maioria dos cenários (incluindo todos os 9 cenários com 8 objetivos), o D-MEANDS obteve o melhor desempenho. Foi possível perceber que o aumento do número de objetivos evidencia o desempenho do D-MEANDS. Por outro lado, o aumento do número de mudanças de ambiente enfraquece, ainda que levemente, o desempenho do D-MEANDS. Essa característica é mais visível nas instâncias com 4 objetivos. Além disso, observa-se nos gráficos que o aumento do número de itens das instâncias favorece o DNSGA-II*. Comparando o MS-MOEA e o DNSGA-II*, verifica-se

que o primeiro supera o segundo na maioria dos resultados de HV. Já o DDIS-MOEA/D-DE apresentou os piores resultados em todas as instâncias.

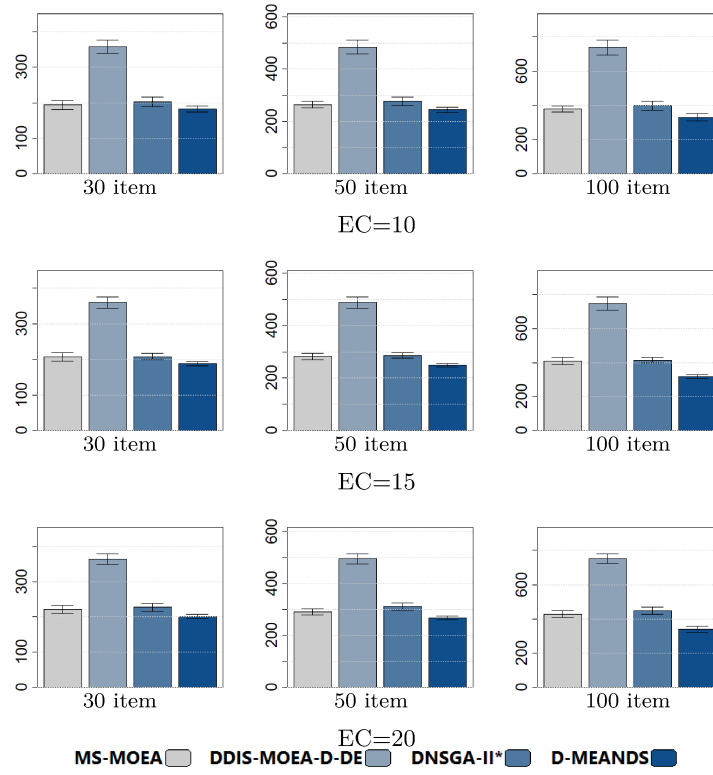


Figura 49 – Resultados da métrica Density (AD) em instâncias de 4 objetivos.

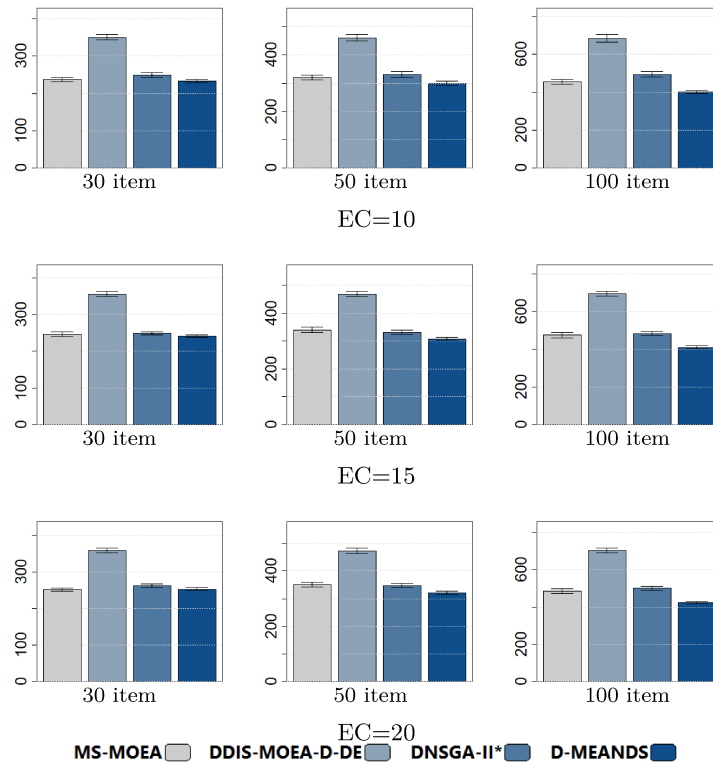


Figura 50 – Resultados da métrica Density (AD) em instâncias de 6 objetivos.

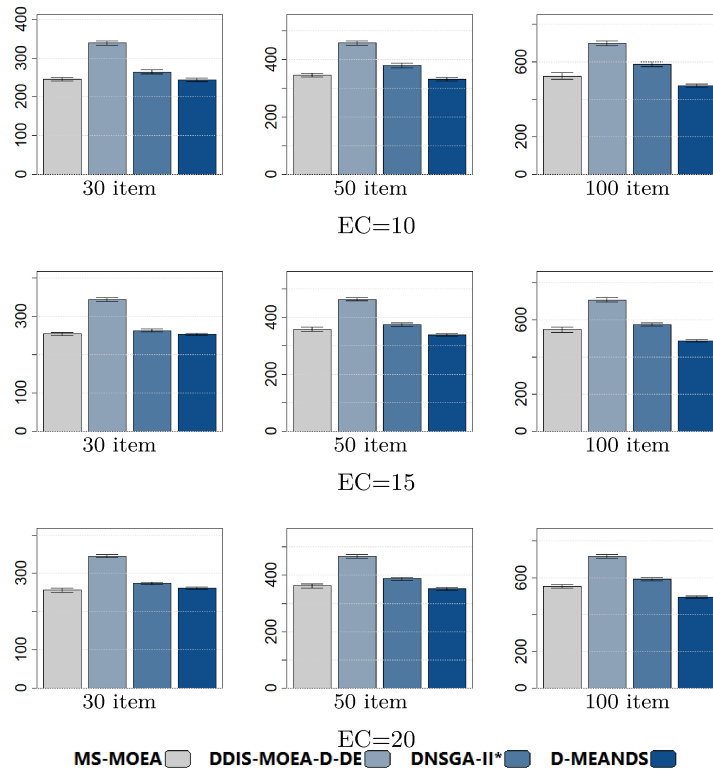


Figura 51 – Resultados da métrica Density (AD) em instâncias de 8 objetivos.

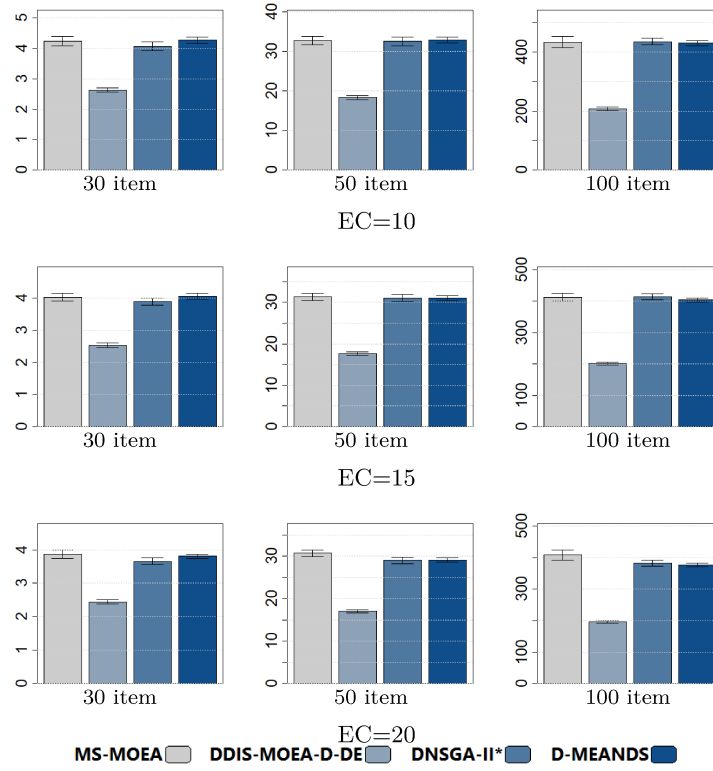


Figura 52 – Resultados da métrica Hipervolume (HV) em instâncias de 4 objetivos.

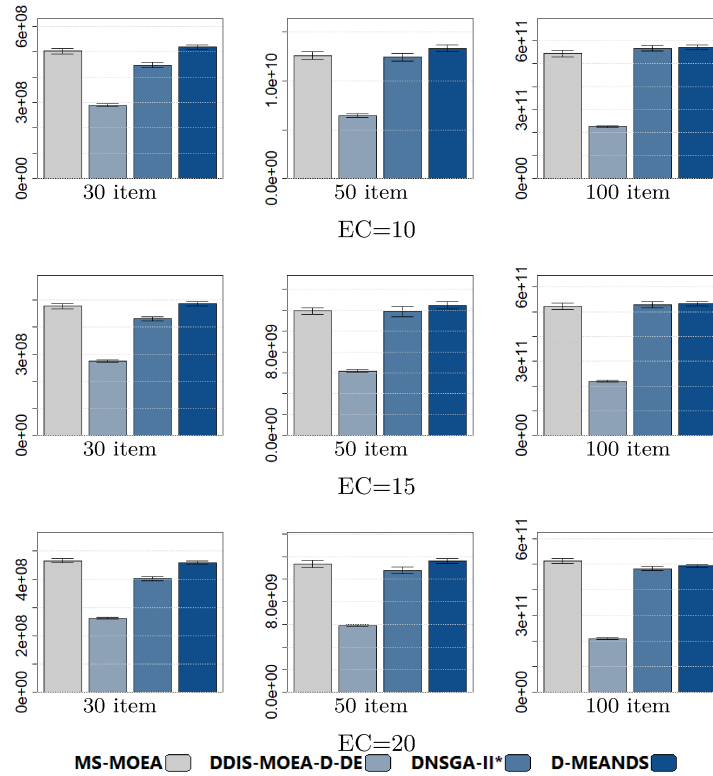


Figura 53 – Resultados da métrica Hipervolume (HV) em instâncias de 6 objetivos.

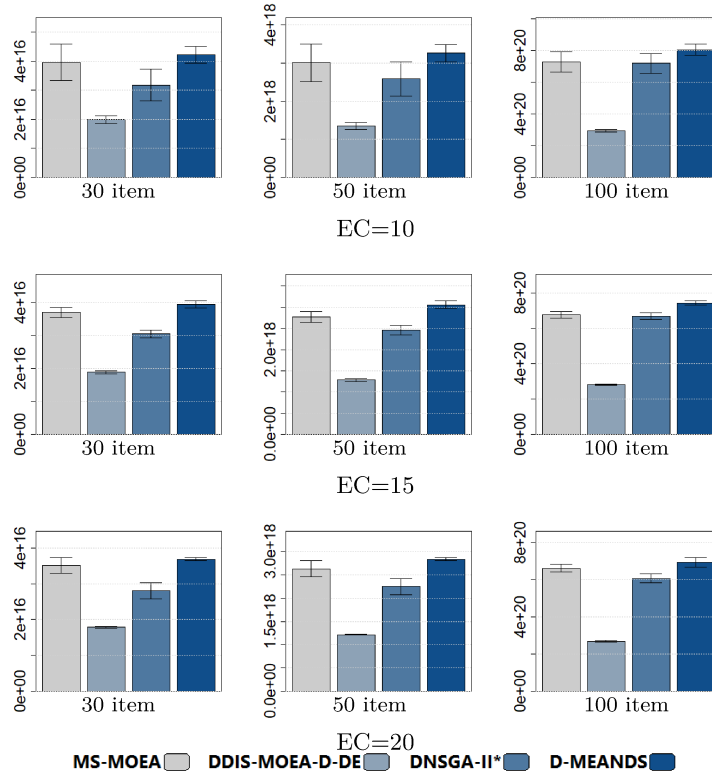


Figura 54 – Resultados da métrica Hipervolume (HV) em instâncias de 8 objetivos.

B.3 Fase 3: Análise de Desempenho do Algoritmo D-MEANDS-MD

Nesta seção, a proposta D-MEANDS-MD é comparada aos algoritmos DNSGA-II* e D-MEANDS, selecionados em função dos resultados obtidos na Seção 4.3.4. Adicionalmente, foi implementada uma variante do DNSGA-II*, denominada DNSGA-II⁺, que incorpora o mesmo mecanismo de diversidade do D-MEANDS-MD, utilizando um pai aleatório em 10% dos cruzamentos. A avaliação dos MOEAs na resolução do DMKP foi realizada por meio das métricas AD e HV.

As Figuras 55, 56 e 57 apresentam os resultados relacionados à métrica AD. O D-MEANDS-MD apresentou desempenho superior ao de todos os demais algoritmos, seguido pelo D-MEANDS original. O DNSGA-II⁺ não apresentou melhora em comparação com o DNSGA-II*. Analisando os resultados da métrica HV nas Figuras 58, 59 e 60, observa-se que o D-MEANDS-MD também superou todos os outros algoritmos, novamente seguido pelo D-MEANDS. Assim como observado na outra métrica, o DNSGA-II⁺ não apresentou melhora em relação ao DNSGA-II*. Portanto, a introdução de um mecanismo de aleatoriedade ao DNSGA-II* não resultou em nenhuma melhoria significativa. Por outro lado, esse mecanismo teve um impacto positivo no D-MEANDS, uma vez que a proposta D-MEANDS-MD melhorou os resultados em todas as instâncias e em ambas as métricas. Como consequência, o novo algoritmo superou todos os demais algoritmos.

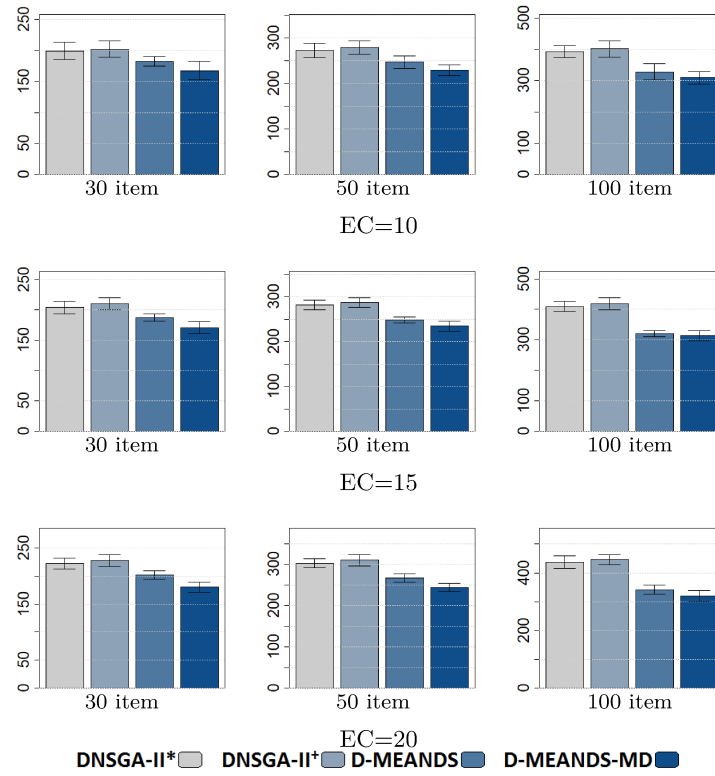


Figura 55 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS-MD em instâncias de 4 objetivos.

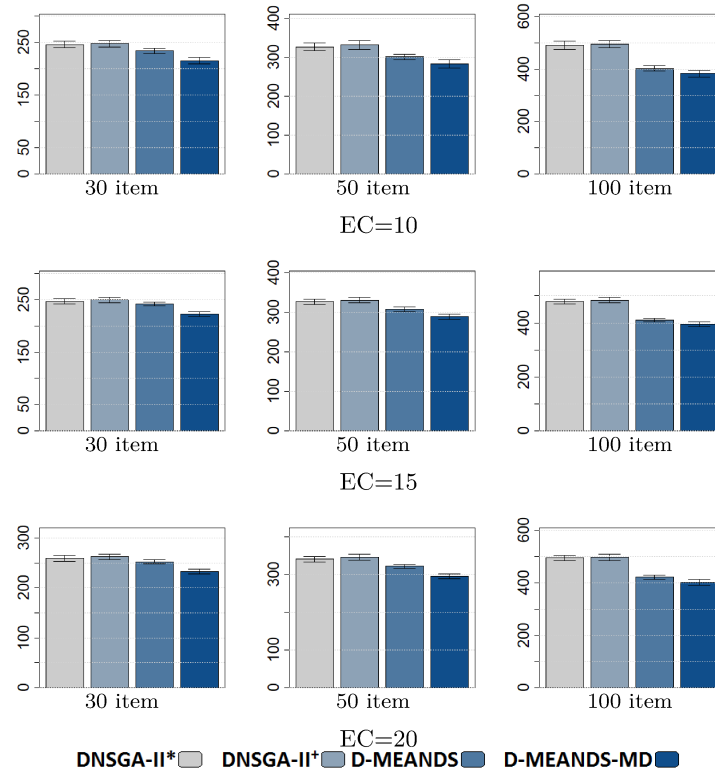


Figura 56 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS-MD em instâncias de 6 objetivos.

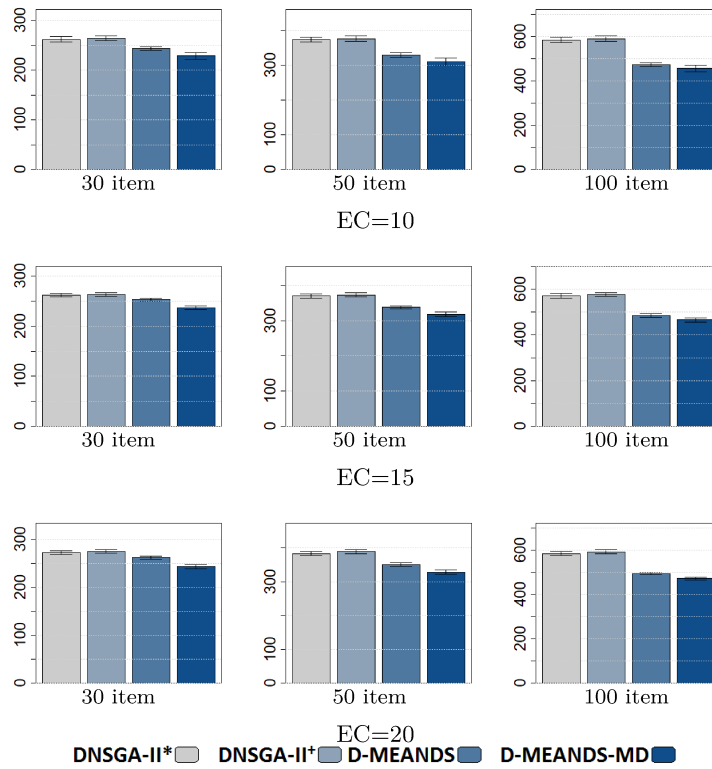


Figura 57 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS-MD em instâncias de 8 objetivos.

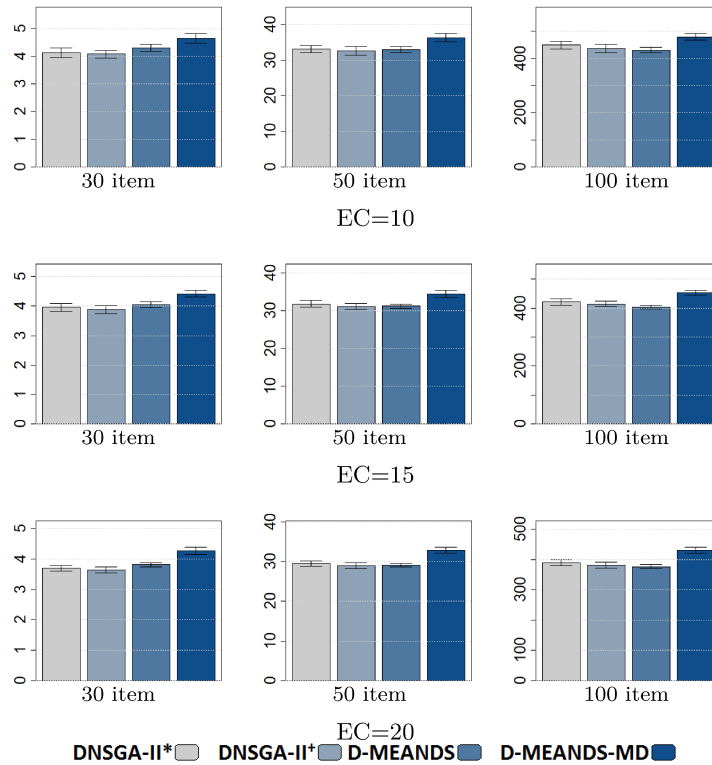


Figura 58 – Resultados da métrica Hipervolume (HV). Investigando a proposta D-MEANDS-MD em instâncias de 4 objetivos.

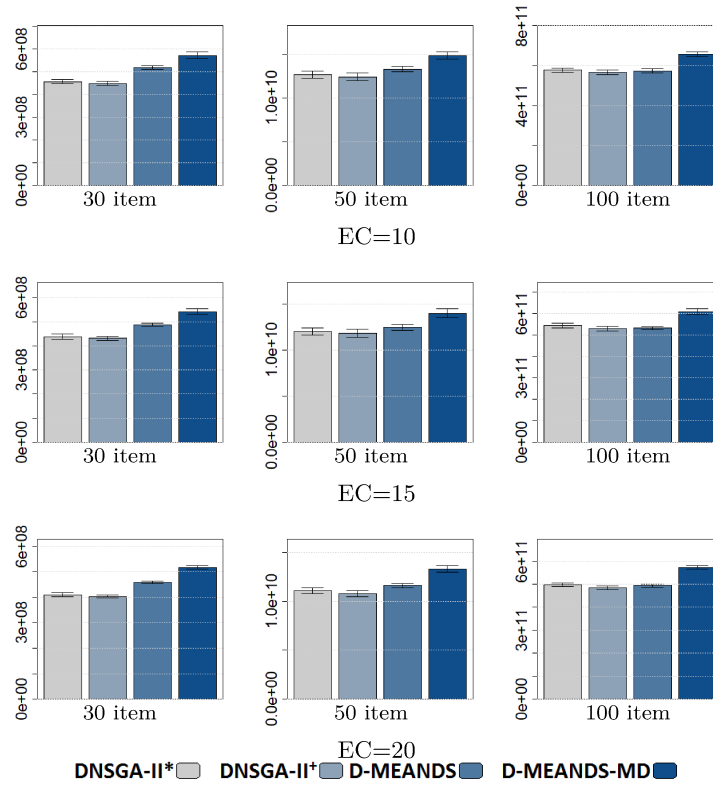


Figura 59 – Resultados da métrica Hipervolume (HV). Investigando a proposta D-MEANDS-MD em instâncias de 6 objetivos.

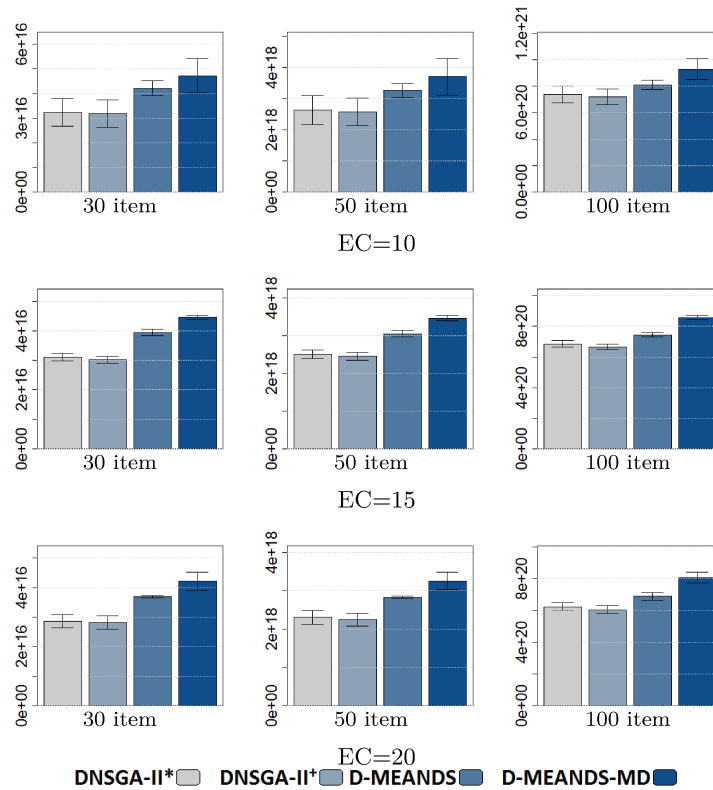


Figura 60 – Resultados da métrica Hipervolume (HV). Investigando a proposta D-MEANDS-MD em instâncias de 8 objetivos.

B.4 Fase 4: Análise de Desempenho do Algoritmo D-MEANDS-II

Nesta Fase, o algoritmo D-MEANDS-II é avaliado em comparação ao MS-MOEA e DNSGA-II*. Além disso, apenas o D-MEANDS-MD foi considerado nos experimentos, em virtude de seu desempenho superior ao D-MEANDS nas avaliações anteriores. A análise dos MOEAs foi conduzida com base nas métricas não paramétricas AD e HV.

As Figuras 61, 62 e 63 apresentam os resultados da métrica AD para as 27 instâncias do DMKP. Em 21 das 27 instâncias, o D-MEANDS-MD apresentou o melhor valor para a métrica AD, ficando o D-MEANDS-II em segundo lugar. Nas 6 instâncias restantes, o D-MEANDS-II superou os demais algoritmos na métrica AD. Ambos os algoritmos, D-MEANDS-MD e D-MEANDS-II, alcançaram resultados próximos nessa métrica. O DNSGA-II* apresentou os piores valores de AD, enquanto o MS-MOEA ficou em penúltimo lugar.

Nas Figuras 64, 65 e 66, podem ser observados os resultados da métrica HV nas 27 instâncias do DMKP. Nota-se que o D-MEANDS-II ocupa a primeira posição em 21 instâncias, deixando o segundo lugar para o D-MEANDS-MD, incluindo todas as 9 instâncias com 4 objetivos. Nas 6 instâncias restantes, o D-MEANDS-MD apresentou o melhor resultado na métrica HV, ficando o D-MEANDS-II em segundo lugar. O MS-MOEA manteve desempenho superior ao do DNSGA-II*, embora ambos tenham sido superados pelo D-MEANDS-II.

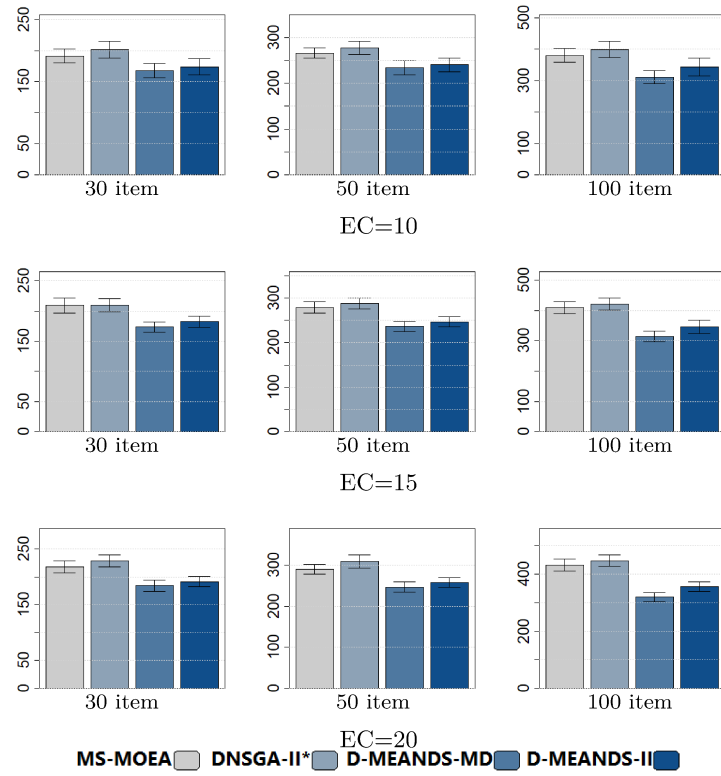


Figura 61 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS-MD em instâncias de 4 objetivos.

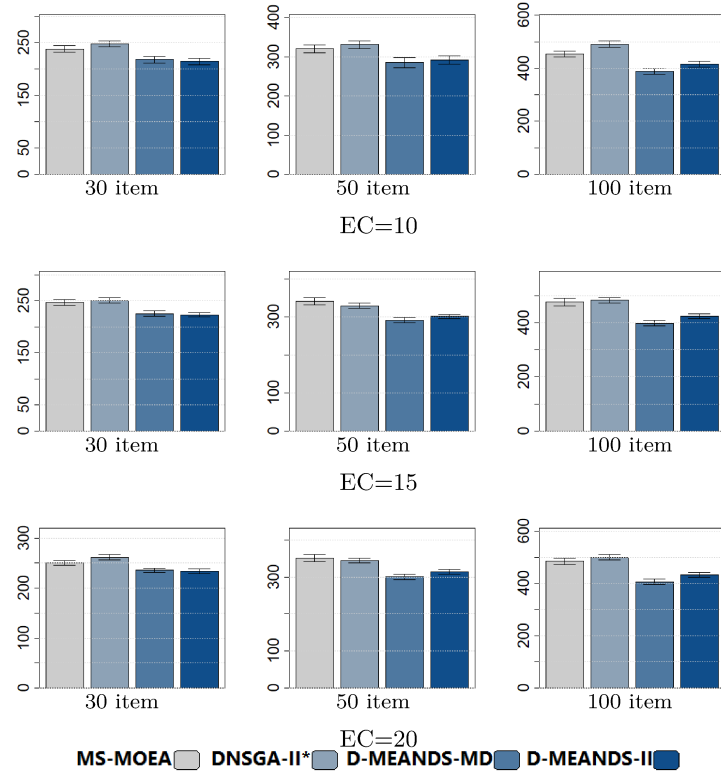


Figura 62 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS-MD em instâncias de 6 objetivos.

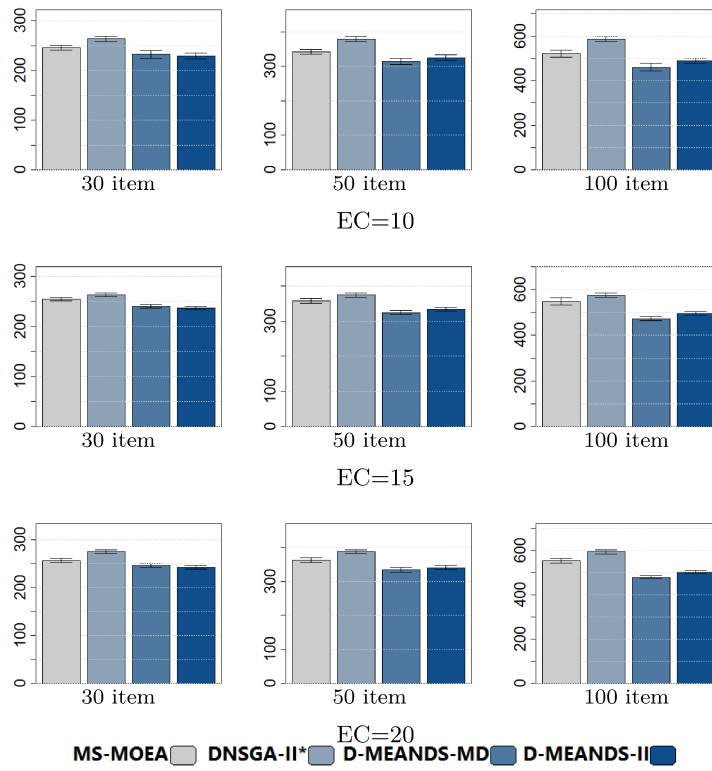


Figura 63 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS-MD em instâncias de 8 objetivos.

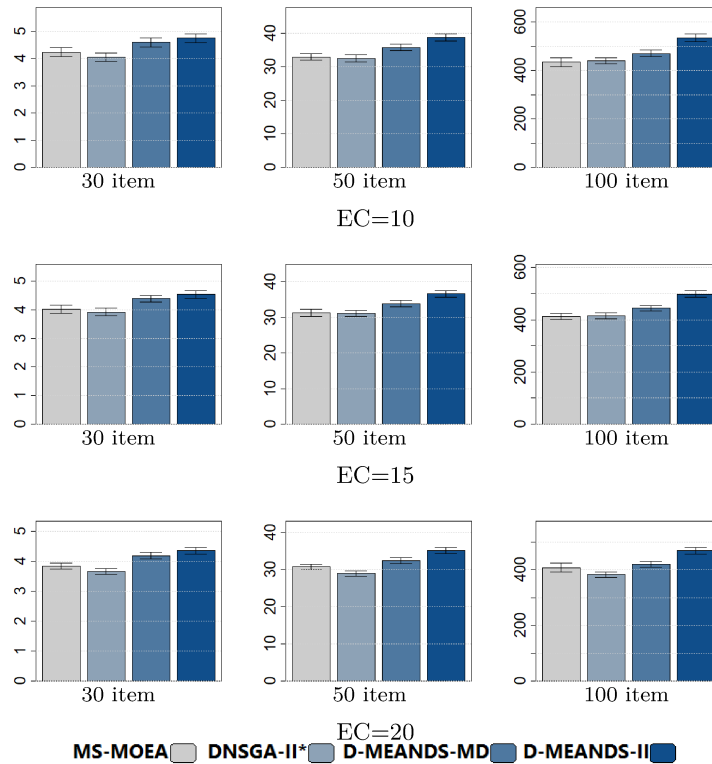


Figura 64 – Resultados da métrica Hipervolume (HV). Investigando a proposta D-MEANDS-MD em instâncias de 4 objetivos.

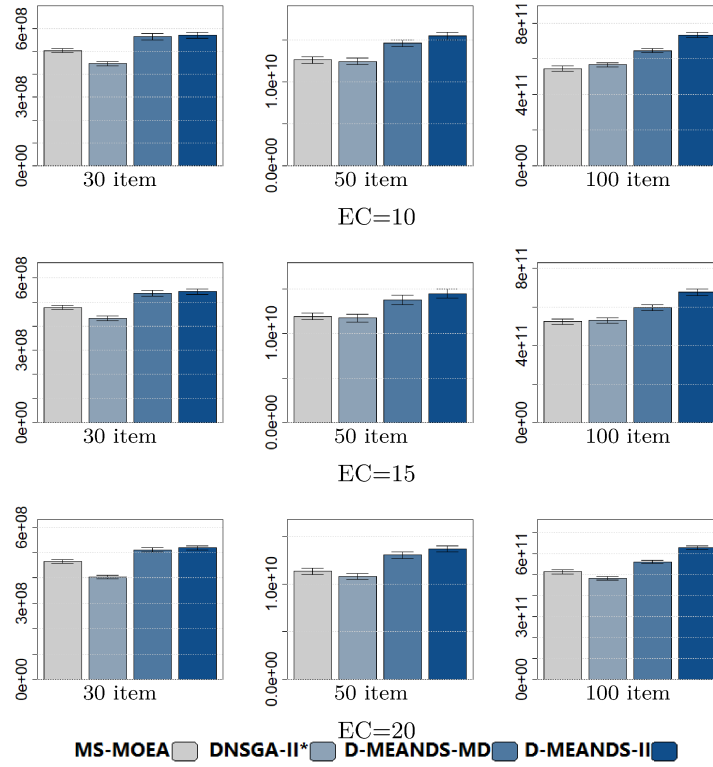


Figura 65 – Resultados da métrica Hipervolume (HV). Investigando a proposta D-MEANDS-MD em instâncias de 6 objetivos.

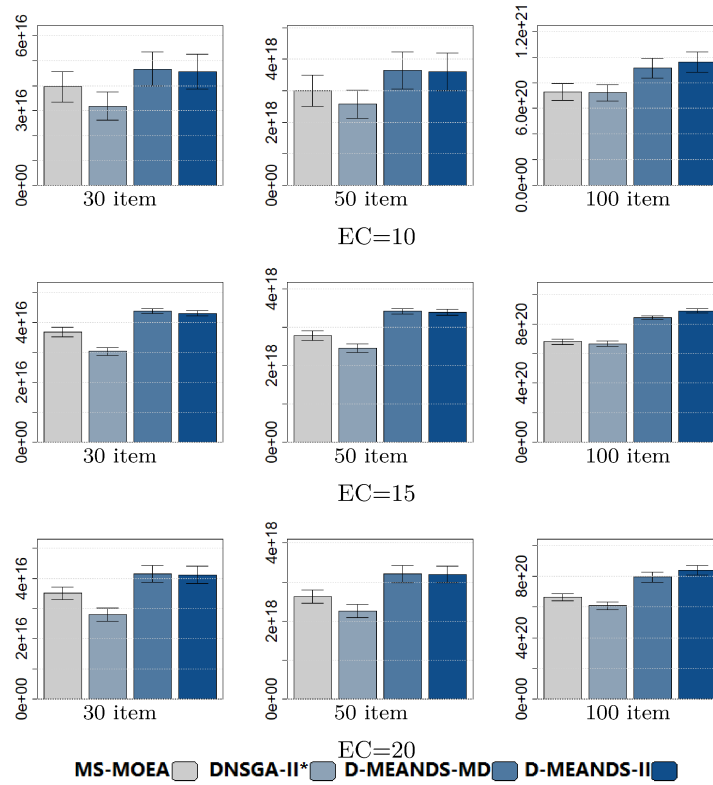


Figura 66 – Resultados da métrica Hipervolume (HV). Investigando a proposta D-MEANDS-MD em instâncias de 8 objetivos.

B.5 Fase 5: Análise de Desempenho do Algoritmo D-MEANDS-III

A avaliação conduzida nesta etapa tem como objetivo analisar o desempenho do D-MEANDS-III em relação aos algoritmos MS-MOEA, DNSGA-II*, D-MEANDS-MD e D-MEANDS-II. Para a comparação entre os diferentes MOEAs, foram empregadas as métricas não paramétricas AD e HV.

As Figuras 67, 68 e 69 apresentam os resultados da métrica AD para as 27 instâncias do DMKP. O D-MEANDS-III mostrou-se bastante competitivo em relação ao D-MEANDS-MD e ao D-MEANDS-II nessa métrica, além de ter superado claramente o MS-MOEA e o DNSGA-II*. O D-MEANDS-III supera o D-MEANDS-II em 18 instâncias, embora seja importante destacar que os valores obtidos por ambos são próximos. Em comparação com o D-MEANDS-MD, o D-MEANDS-III apresenta desempenho inferior na grande maioria das instâncias; no entanto, também nesse caso, os valores observados são bastante próximos. O aumento do número de mudanças de ambiente (EC) favorece o D-MEANDS-MD em comparação ao D-MEANDS-III. Por outro lado, o aumento de EC favorece o D-MEANDS-III em comparação ao D-MEANDS-II.

As Figuras 70, 71 e 72 apresentam os resultados da métrica HV para as 27 instâncias do DMKP. Nessa métrica, o D-MEANDS-III superou o D-MEANDS-MD e o D-MEANDS-II na grande maioria das instâncias. Também é fácil observar que o D-MEANDS-III superou o MS-MOEA e o DNSGA-II* em todas as instâncias. Esses resultados evidenciam um ganho, em termos de convergência e diversidade, da proposta D-MEANDS-III em relação ao D-MEANDS-II.

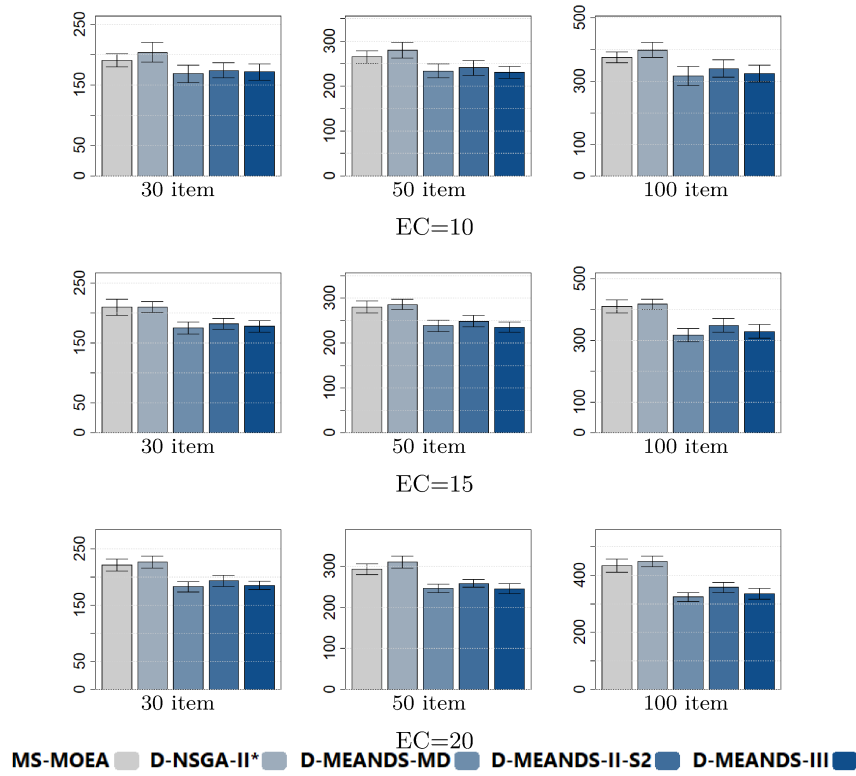


Figura 67 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS-III em instâncias de 4 objetivos.

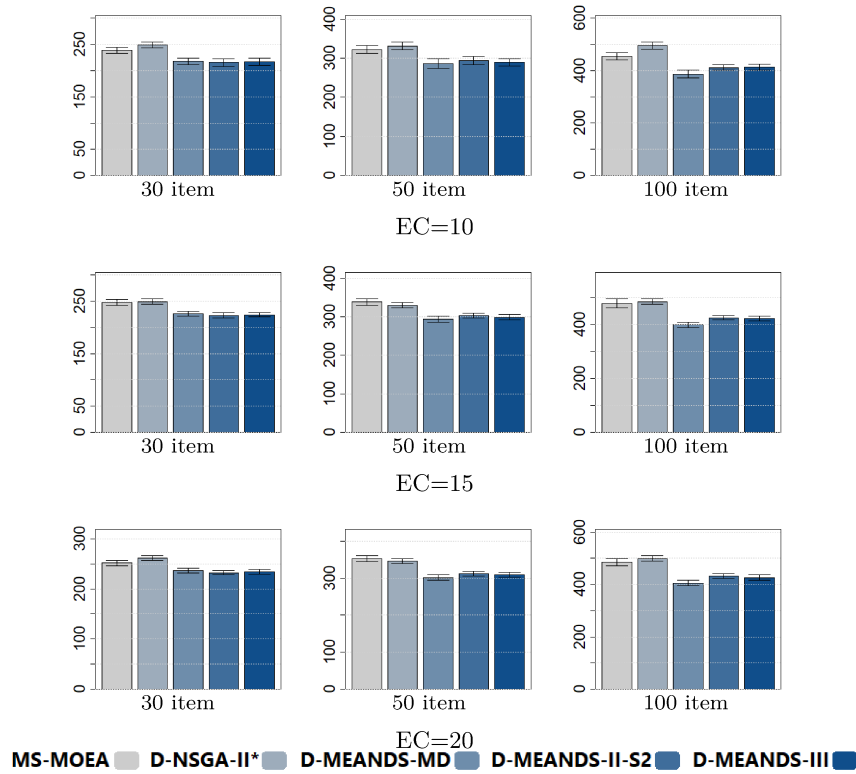


Figura 68 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS-III em instâncias de 6 objetivos.

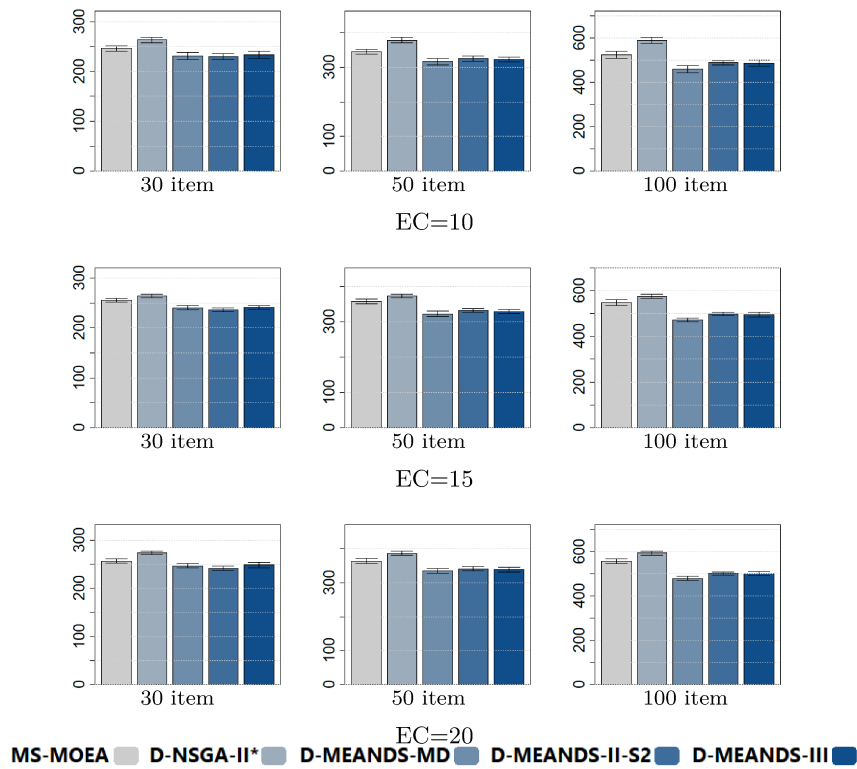


Figura 69 – Resultados da métrica Density (AD). Investigando a proposta D-MEANDS-III em instâncias de 8 objetivos.

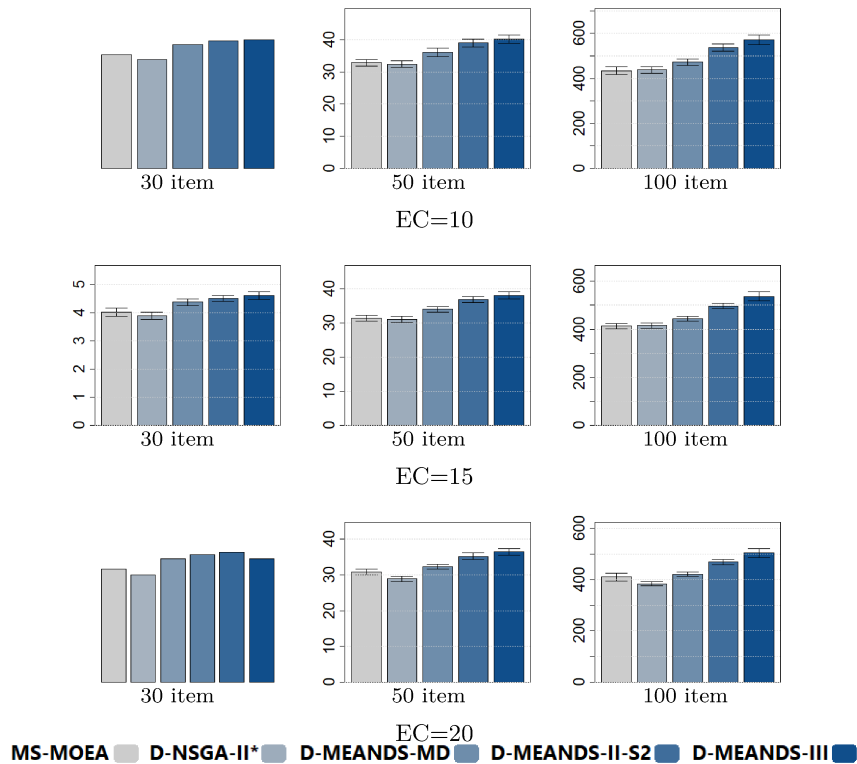


Figura 70 – Resultados da métrica Hipervolume (HV). Investigando a proposta D-MEANDS-III em instâncias de 4 objetivos.

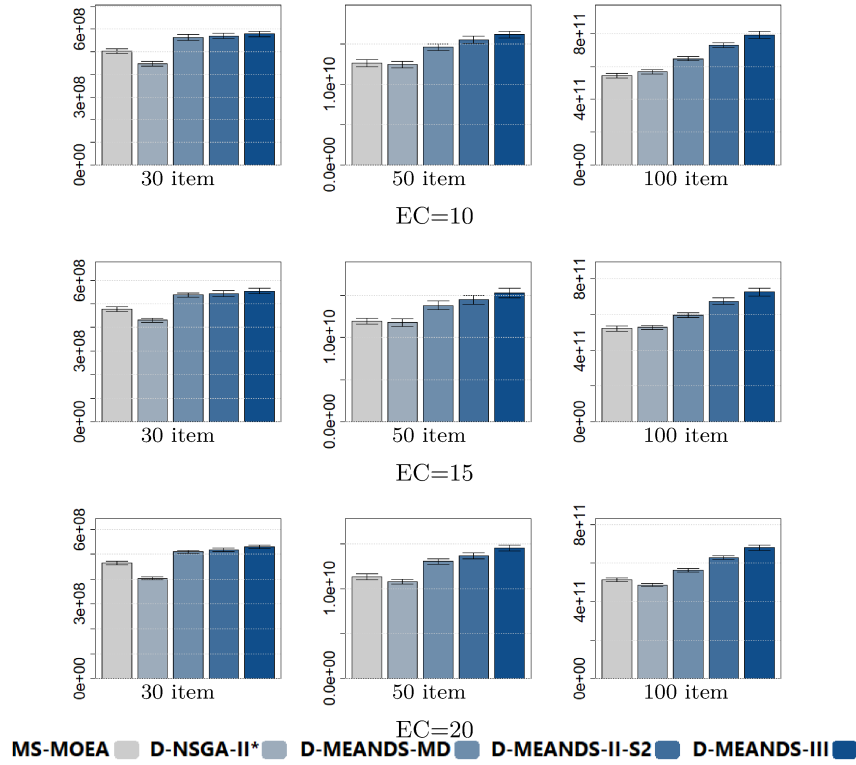


Figura 71 – Resultados da métrica Hipervolume (HV). Investigando a proposta D-MEANDS-III em instâncias de 6 objetivos.

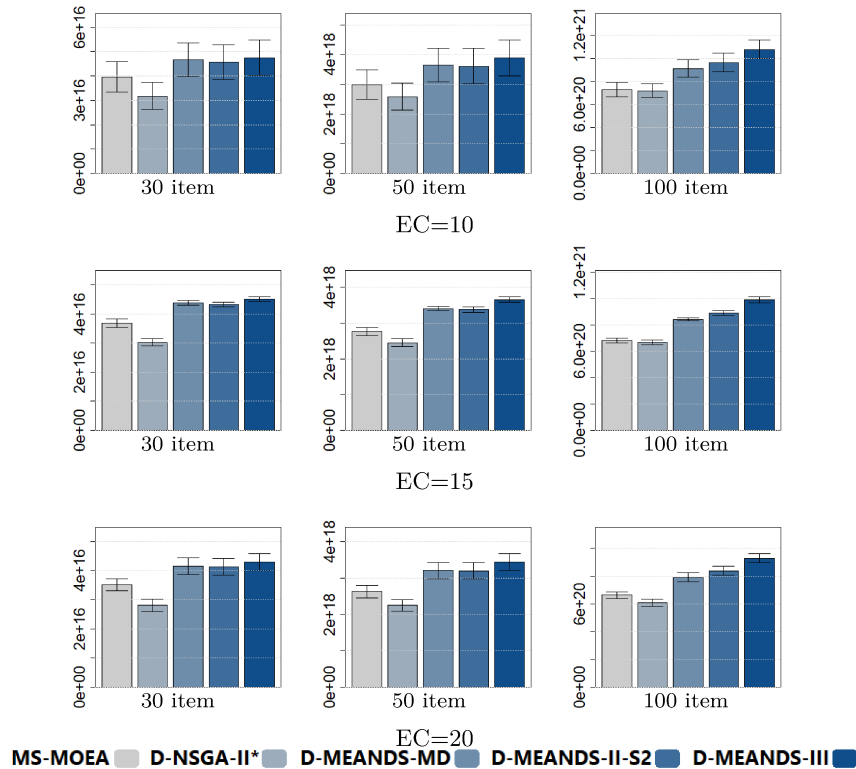


Figura 72 – Resultados da métrica Hipervolume (HV). Investigando a proposta D-MEANDS-III em instâncias de 8 objetivos.