

Leonardo César Teonácio Bezerra

Uma colônia de formigas para o caminho mais curto multiobjetivo

Natal, Rio Grande do Norte

07 de Fevereiro de 2011

Leonardo César Teonácio Bezerra

Uma colônia de formigas para o caminho mais curto multiobjetivo

Dissertação de mestrado apresentada à Universidade Federal do Rio Grande do Norte, como requisito parcial para a obtenção do grau de Mestre em Sistemas e Computação. Área de concentração: Algoritmos Experimentais

Orientadora Elizabeth Ferreira Gouvêa Goldberg

Co-orientadora Luciana Salete Buriol

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO

Natal, Rio Grande do Norte

07 de Fevereiro de 2011

Dissertação de mestrado sob o título "*Uma colônia de formigas para o caminho mais curto multiobjetivo*", defendida por Leonardo César Teonácio Bezerra e aprovada em 07 de fevereiro de 2011, em Natal, Rio Grande do Norte, pela banca examinadora constituída pelos doutores:

Prof^a. Dr^a. Elizabeth Ferreira Gouvêa Goldbarg
Orientadora
Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte

Prof^a. Dr^a. Luciana Salete Buriol
Co-orientadora
Instituto de Informática
Universidade Federal do Rio Grande do Sul

Prof. Dr. Marco César Goldbarg
Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte

Prof^a. Dr^a. Anne Magaly de Paula Canuto
Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte

Agradecimentos

(...)

À minha mãe, por todo o investimento feito ao longo desses anos.

À minha irmã, pela revisão sempre bem humorada.

A Vanessa, por nos entendermos sem palavras. Obrigado pela paciência, compreensão, esforço e por me ajudar a redigitar essa dissertação. Te amo.

À professora Elizabeth, pela ajuda na caminhada. Nunca mais rodo um algoritmo exato dez vezes na mesma instância.

Ao professor Goldbarg, pelo convite para o mestrado.

À professora Luciana, pelas madrugadas em claro. Curta bem a Alice.

Ao Wagner, pela ajuda nesse tempo todo.

Ao Everton, pelos socorros bem presentes.

Ao Paulo, pelos churrascos experimentados. Bah, tchê.

A cada um que de certa forma eu atrapalhei menos que ajudei, apesar de serem poucos.

Resumo

Problemas de otimização combinatória multiobjetivo apresentam características peculiares que exigem que técnicas de otimização se adaptem a esse contexto. Como muitos desses problemas são NP-Árduos, o uso de metaheurísticas tem crescido nos últimos anos. Particularmente, muitas abordagens que utilizam a Otimização por Colônias de Formigas têm sido propostas. Neste trabalho, propõe-se um algoritmo baseado em colônias de formigas para o Problema do Caminho mais Curto Multiobjetivo, e compara-se o algoritmo proposto com dois otimizadores encontrados na literatura. Um conjunto de 18 instâncias oriundas de dois tipos de grafos é utilizado, além de uma metodologia específica para a avaliação de otimizadores multiobjetivo. Os experimentos iniciais mostram que o algoritmo proposto consegue gerar conjuntos de aproximação melhores que os demais otimizadores para todas as instâncias. Na segunda parte do trabalho, uma análise experimental de diferentes abordagens publicadas para colônias de formigas multiobjetivo é realizada, usando as mesmas instâncias. Os experimentos mostram que cada tipo de instância privilegia uma abordagem algorítmica diferente. Uma nova metáfora para o desenvolvimento deste tipo de metaheurística é então proposta. Geralmente, formigas possuem características comuns e poucos artigos abordam o uso de múltiplas espécies. Neste trabalho, uma abordagem com múltiplas espécies competindo por fontes de comida é proposta. Cada espécie possui sua própria estratégia de busca e diferentes espécies não tem acesso à informação dada pelo feromônio das outras. Como na natureza, as populações de formigas bem sucedidas tem a chance de crescer, enquanto as demais se reduzem. A abordagem apresentada aqui mostra-se capaz de herdar o comportamento de estratégias bem-sucedidas em diferentes tipos de instâncias. Resultados de experimentos computacionais são relatados e mostram que a abordagem proposta produz conjuntos de aproximação significativamente melhores que os outros métodos.

Palavras-chave: Metaheurísticas; Otimização por Colônias de Formigas; Otimização Multiobjetivo; Caminho mais curto; Múltiplas espécies; Regulação por disponibilidade de comida.

Abstract

Multi-objective combinatorial optimization problems have peculiar characteristics that require optimization methods to adapt for this context. Since many of these problems are NP-Hard, the use of metaheuristics has grown over the last years. Particularly, many different approaches using Ant Colony Optimization (ACO) have been proposed. In this work, an ACO is proposed for the Multi-objective Shortest Path Problem, and is compared to two other optimizers found in the literature. A set of 18 instances from two distinct types of graphs are used, as well as a specific multiobjective performance assessment methodology. Initial experiments showed that the proposed algorithm is able to generate better approximation sets than the other optimizers for all instances. In the second part of this work, an experimental analysis is conducted, using several different multiobjective ACO proposals recently published and the same instances used in the first part. Results show each type of instance benefits a particular type of instance benefits a particular algorithmic approach. A new metaphor for the development of multiobjective ACOs is, then, proposed. Usually, ants share the same characteristics and only few works address multi-species approaches. This work proposes an approach where multi-species ants compete for food resources. Each specie has its own search strategy and different species do not access pheromone information of each other. As in nature, the successful ant populations are allowed to grow, whereas unsuccessful ones shrink. The approach introduced here shows to be able to inherit the behavior of strategies that are successful for different types of problems. Results of computational experiments are reported and show that the proposed approach is able to produce significantly better approximation sets than other methods.

Keywords: Metaheuristics; Ant Colony Optimization; Multiobjective Optimization; Shortest Path Problem; Multi-species; Food regulation.

Lista de figuras

| | | |
|----|---|-------|
| 1 | As soluções não-suportadas de um problema biobjetivo encontram-se nas regiões triangulares delimitadas por soluções suportadas consecutivas (RAITH; EHRGOTT, 2009). | p. 21 |
| 2 | Classificação dos algoritmos exatos para o BSP (SKRIVER, 2000). | p. 25 |
| 3 | Exemplo de uma iteração da 3-Logos. | p. 37 |
| 4 | Exemplo de ponderações e colônias para a versão PGRan. Com $d = 2$, seis vetores são gerados, resultando em quatro colônias. | p. 49 |
| 5 | Violin plots para o indicador $I_{\epsilon+}$ da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO. | p. 83 |
| 6 | Violin plots para o indicador $I_{\epsilon+}$ da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO. | p. 84 |
| 7 | Violin plots para o indicador I_H^- da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO. | p. 85 |
| 8 | Violin plots para o indicador I_H^- da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO. | p. 86 |
| 9 | Violin plots para a distância geracional da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO. | p. 87 |
| 10 | Violin plots para a distância geracional da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO. | p. 88 |
| 11 | Violin plots para a taxa de erro da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO. | p. 89 |
| 12 | Violin plots para a taxa de erro da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO. | p. 90 |
| 13 | Violin plots para a métrica \mathcal{C}_\prec da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO. | p. 91 |

| | | |
|----|---|--------|
| 14 | Violin plots para a métrica \mathcal{C}_{\prec} da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO. | p. 92 |
| 15 | Violin plots para o indicador $I_{\epsilon+}$ da comparação entre a versão R&B-FRd, o NSGA-II e o HACO. | p. 93 |
| 16 | Violin plots para o indicador $I_{\epsilon+}$ da comparação entre a versão R&B-FRd, o NSGA-II e o HACO. | p. 94 |
| 17 | Violin plots para o indicador I_H^- da comparação entre a versão R&B-FRd, o NSGA-II e o HACO. | p. 95 |
| 18 | Violin plots para o indicador I_H^- da comparação entre a versão R&B-FRd, o NSGA-II e o HACO. | p. 96 |
| 19 | Violin plots para a distância geracional da comparação entre a versão R&B-FRd, o NSGA-II e o HACO. | p. 97 |
| 20 | Violin plots para a distância geracional da comparação entre a versão R&B-FRd, o NSGA-II e o HACO. | p. 98 |
| 21 | Violin plots para a taxa de erro da comparação entre a versão R&B-FRd, o NSGA-II e o HACO. | p. 99 |
| 22 | Violin plots para a taxa de erro da comparação entre a versão R&B-FRd, o NSGA-II e o HACO. | p. 100 |
| 23 | Violin plots para a métrica \mathcal{C}_{\prec} da comparação entre a versão R&B-FRd, o NSGA-II e o HACO. | p. 101 |
| 24 | Violin plots para a métrica \mathcal{C}_{\prec} da comparação entre a versão R&B-FRd, o NSGA-II e o HACO. | p. 102 |

Lista de tabelas

| | | |
|----|--|-------|
| 1 | Relações de dominância entre soluções (KNOWLES; THIELE; ZITZLER, 2006). | p. 20 |
| 2 | Relações de dominância entre conjuntos de soluções em um problema multiobjetivo (KNOWLES; THIELE; ZITZLER, 2006). | p. 20 |
| 3 | Lista de instâncias. Somente os três primeiros objetivos foram usados em instâncias <i>large</i> . | p. 40 |
| 4 | Valores de parâmetros testados pro NSGA-II. | p. 41 |
| 5 | p -valores do teste de Kruskal-Wallis usando rankings de dominância de todos os otimizadores. | p. 42 |
| 6 | p -valores da comparação de proporções: GRACE, NSGA-II (DEB et al., 2002) e HACO (HÄCKEL et al., 2008). | p. 42 |
| 7 | Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias completas. | p. 43 |
| 8 | Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias completas (continuação). | p. 44 |
| 9 | Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias <i>grid</i> . | p. 45 |
| 10 | Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias <i>grid</i> (continuação). | p. 46 |
| 11 | Resultados do teste de Kruskal-Wallis nos rankings de dominância e nos indicadores de qualidade gerados para a GRan usando 1, 3 e 5 colônias. NaN significa que as amostras comparadas são iguais. | p. 49 |
| 12 | p -valores da comparação de proporções para a versão GRan. | p. 50 |
| 13 | p -valores do teste de Kruskal-Wallis: GRan com uma colônia contra FUN. | p. 50 |
| 14 | p -valores para a comparação de proporções da GRan contra a FUN. | p. 51 |

| | | |
|----|---|-------|
| 15 | Análise do parâmetro m para a FMu com o teste de Kruskal-Wallis: $d = 6$, $m \in \{1, 3, 5\}$. NaN significa que as amostras comparadas são iguais. | p. 52 |
| 16 | Resultados da comparação de proporções para as instâncias <i>grid</i> . | p. 52 |
| 17 | Resultados para a comparação da PGRan usando $d \in \{1, 2, 3, 4, 5\}$. | p. 52 |
| 18 | p -valores para a comparação de proporções da PGRan com $d \in \{1, 2, 3, 4, 5\}$. | p. 53 |
| 19 | Teste de Kruskal-Wallis para versões com uma colônia: GRan, Fix, FMu e PGRan. | p. 53 |
| 20 | p -valores para comparação de proporções entre versões com uma colônia: GRan, Fix, FMu e PGRan. | p. 54 |
| 21 | Comparação geral das versões com três colônias usando o teste de Kruskal- Wallis: GRan, Fix, FMu e PGRan. | p. 54 |
| 22 | Comparação de proporções entre as versões GRan, Fix, FMu e PGRan ($m = 3$). | p. 55 |
| 23 | Resultados do teste bicaudal de Wilcoxon para a versão Mix: $m = 1$ e $m = 3$ ($d = 6$). | p. 57 |
| 24 | Resultados do teste de Kruskal-Wallis para a versão Obj com diferentes métodos de atualização de feromônio: por origem, por região ou o uso de ambas simultaneamente. | p. 58 |
| 25 | Comparação de proporções da versão Obj com diferentes estratégias de atualização de feromônio. | p. 58 |
| 26 | Resultados do teste de Kruskal-Wallis para a versão OMix com $d = 3$: atualização por origem, atualização por região e o uso de ambas simul- taneamente. | p. 59 |
| 27 | p -valores para o teste de comparação de proporções da versão OMix com as diferentes estratégias de atualização de feromônio. | p. 59 |
| 28 | Resultados do teste de Kruskal-Wallis ($m = d = 3$): GRan, FMu, Mix, Obj e OMix (as duas últimas versões usando a atualização por origem). | p. 60 |
| 29 | Comparação de proporções com $m = d = 3$: GRan, FMu, Mix, Obj e OMix (as duas últimas com atualização por origem). | p. 60 |

| | | |
|----|--|-------|
| 30 | Comparação de proporções com $m = d = 3$: GRan, FMu, Mix, Obj e OMix (as duas últimas com atualização por origem e por região simultaneamente). | p. 61 |
| 31 | p -valores dos teste de Kruskal-Wallis: comparando as versões GRan, FMu e Mix com apenas uma colônia, e as versões Obj e OMix com 3 colônias e atualização por origem de região simultaneamente. | p. 61 |
| 32 | p -valores da comparação dois-a-dois: versões GRan, FMu e Mix com apenas uma colônia, e versões Obj e OMix com 3 colônias e atualização por origem e região simultaneamente. | p. 62 |
| 33 | Comparação efetuada na experimentação do parâmetro d da versão R&B com $m = 1$ | p. 66 |
| 34 | p -valores obtidos com o teste bicaudal de Wilcoxon para a versão R&B com $m \in \{1, 3\}$ e $d = 3$ | p. 66 |
| 35 | p -valores obtidos com o teste de Kruskal-Wallis para as versões com três colônias GRan, FMu, R&B, Obj e OMix, com $d = 3$ | p. 67 |
| 36 | Resultados da comparação de proporções dois-a-dois entre as versões com três colônias R&B, GRan, FMu, Obj e OMix (com $d = 3$). | p. 67 |
| 37 | Comparação de proporções: versões R&B, GRan e FMu com apenas uma colônia, e versões Obj e OMix com 3 colônias e atualização por origem e região simultaneamente. | p. 68 |
| 38 | p -valores obtidos com o teste de Kruskal-Wallis na comparação das variantes com regulação por disponibilidade de comida em relação à versão R&B. | p. 69 |
| 39 | Teste de comparação de proporções usando as versões com uma colônia R&B-FRd, GRan e FMu, e as versões com 3 colônias Obj e OMix (estas usando atualizações por origem e região simultaneamente). | p. 70 |
| 40 | p -valores da comparação de proporções: R&B-FRd com uma colônia, NSGA-II (DEB et al., 2002) e HACO (HÄCKEL et al., 2008). | p. 71 |
| 41 | Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias completas. | p. 72 |

| | | |
|----|--|-------|
| 42 | Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias completas (continuação). | p. 73 |
| 43 | Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias <i>grid</i> | p. 74 |
| 44 | Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias <i>grid</i> (continuação). | p. 75 |

Lista de abreviaturas e siglas

BSP – *Bi-objective Shortest Path Problem*

ACO – *Ant Colony Optimization*

MSP – *Multi-objective Shortest Path Problem*

NSGA-II – *Non-dominated Sorting Genetic Algorithm II*

HACO – ACO proposta por Häckel et al. (2008)

GRACE – *Generational Randomized Ant Colony Enhancement*

SPEA2 – *Strength Pareto Evolutionary Algorithm 2*

PISA – *Platform and Programming Language Independent Interface for Search Algorithm*

SPEA – *Strength Pareto Evolutionary Algorithm*

rwGA – *Random Weight Genetic Algorithm*

LAH – *Look-Ahead Heuristic*

AS – *Ant System*

ACS – *Ant Colony System*

MOACO – *Multi-Objective Ant Colony Optimization*

P-ACO – *Population Based Ant Colony Optimization*

ACO – *Ant Colony Optimization*

NSGA – *Non-dominated Sorting Genetic Algorithm*

PSA – *Pareto Simulated Annealing*

MOTS – *Multi-objective Tabu Search*

bQAP – *Bi-objective Quadratic Assignment Problem*

MMAS – *MIN-MAX Ant System*

w-RoTS – *Robust Tabu Search*

bTSP – *Bi-objective Traveling Salesman Problem*

Logos – *Logarithmic Objective Space Search*

DR – *Dominance ranking*

QI – *Quality indicators*

GRan – *Global-Random*

FUn – *Fixo-Único*

FMu – *Fixo-Múltiplos*

PGRan – *Pseudo-Global-Random*

Mix – *Mixed*

Obj – *Objective-Driven Colonies*

ObjMixed – *Objective-Driven Mixed Colonies*

R&B – *Red&Black*

R&B-FR – *Red&Black Food Regulated*

R&B-FRd – *Red&Black Food Regulated* com Divisão Proporcional

R&B-FRl – *Red&Black Food Regulated* com Crescimento Limitado

R&B-FRld – *Red&Black Food Regulated* com Divisão Proporcional e Crescimento Limitado

Lista de trabalhos produzidos a partir desta dissertação

BEZERRA, L. C. T. et al. GRACE: A generational randomized ACO for the multi-objective shortest path problem. In: TAKAHASHI, R. et al. (Ed.). *Evolutionary Multi-criterion Optimization*. [S.l.]: Springer, 2011, (Lecture Notes in Computer Science, v. 6576). p. 535–549.

BEZERRA, L. C. T. et al. Food Regulated Pareto Multi-Species: a new ACO Approach for the Multi-objective Shortest Path Problem. Departamento de Informática e Matemática Aplicada, Universidade Federal do Rio Grande do Norte. Relatório Técnico no. UFRN-DIMAp-2011-104-RT. Natal, RN, Brasil, 2011.

Sumário

| | | |
|----------|---|-------|
| 1 | Introdução | p. 16 |
| 2 | O Problema do Caminho mais Curto Multiobjetivo | p. 19 |
| 2.1 | Otimização Multiobjetivo | p. 19 |
| 2.2 | O Problema do Caminho mais Curto Multiobjetivo | p. 24 |
| 3 | Otimização por Colônia de Formigas | p. 28 |
| 3.1 | Colônias de formigas multiobjetivo | p. 30 |
| 3.2 | Estudos comparativos | p. 33 |
| 4 | GRACE: o algoritmo proposto | p. 36 |
| 4.1 | O algoritmo GRACE | p. 36 |
| 4.2 | Metodologia | p. 38 |
| 4.3 | Experimentação | p. 40 |
| 5 | Análise experimental: ampliando o algoritmo GRACE | p. 47 |
| 6 | Múltiplas espécies reguladas por disponibilidade de comida | p. 63 |
| 7 | Considerações finais | p. 76 |
| | Referências | p. 78 |
| | Anexo 1 – Vioplots Cap. 4 | p. 83 |
| | Anexo 2 – Vioplots Cap. 6 | p. 93 |

1 Introdução

Problemas de caminhos mínimos são clássicos das áreas de teoria dos grafos e otimização combinatória. De maneira geral, encontrar o caminho mais curto em um grafo significa encontrar o caminho que une dois vértices e que minimiza uma determinada função objetivo. Dentre diversas aplicações reais, contextos de roteamento são particularmente relevantes. Na Internet, por exemplo, um dos protocolos que podem ser usados para o roteamento de pacotes é o OPSF (*Open Shortest Path First*), que procura o menor caminho disponível para o envio das informações (BURIOL et al., 2005). Em sistemas de navegação, que têm se tornado comuns em carros populares, caminhos mínimos ajudam tanto a planejar como a otimizar os recursos disponíveis. Recentemente, múltiplos objetivos (às vezes conflitantes) têm sido considerados na busca por caminhos mais curtos. Quando um caminhão viaja de um armazém até um local de entrega, por exemplo, não pensa somente no tempo, mas também no trânsito, na conservação das pistas e até no relevo do terreno, devido a seu efeito no consumo de combustível (DOERNER; HARTL; REIMANN, 2001). Ainda em contextos de transporte, uma operação de planejamento militar deve considerar o tempo, a distância e a capacidade de camuflagem do caminho escolhido (TARAPATA, 2003).

Para esses problemas, dificilmente haverá uma única solução que consiga minimizar (ou maximizar) todos os critérios, mas um conjunto de soluções que não podem ser consideradas melhores que as outras para todos os objetivos simultaneamente. Estas são chamadas soluções eficientes não-dominadas, e formam o conjunto Pareto-ótimo. A quantidade de soluções não-dominadas em um problema multiobjetivo cresce exponencialmente de acordo com o tamanho das instâncias e quantidade de objetivos, e muitos são considerados NP-Árduos (SERAFINI, 1986). Raith e Ehrgott (2009), por exemplo, afirmam que o Problema do Caminho mais Curto Bi-objetivo (BSP, do inglês, *Bi-objective Shortest Path Problem*) é considerado intratável (HANSEN, 1980). Assim, algoritmos exatos tendem a ter um fraco desempenho, o que, aliado à larga aplicabilidade dos modelos multiobjetivo, levou ao desenvolvimento de abordagens metaheurísticas criadas especificamente para

problemas com múltiplos critérios. Essas abordagens podem gerar conjuntos subótimos de soluções que tentam se aproximar do conjunto Pareto-ótimo, recuperar parte do próprio conjunto Pareto-ótimo, ou mesmo fazer as duas coisas simultaneamente.

Quando uma metaheurística é aplicada a um problema multiobjetivo, várias questões de projeto surgem. Neste trabalho, a Otimização por Colônia de Formigas (ACO, do inglês, *Ant Colony Optimization*) é investigada. Para esta metaheurística, deve-se decidir inicialmente se uma ou múltiplas colônias serão usadas. Caso seja feita a opção por múltiplas colônias, o algoritmo poderá direcionar suas colônias para investigar regiões específicas, ou manter colônias em paralelo investigando todo o espaço objetivo. Também é necessário definir se cada colônia terá uma matriz de feromônio ou se todas compartilharão uma matriz global. Caso múltiplas matrizes de feromônio e colônias especializadas em regiões sejam utilizadas, cada formiga poderá atualizar a matriz de sua colônia de origem ou a matriz da colônia responsável pela região à qual sua solução pertence, de acordo com a política adotada. Ademais, deve-se definir se haverá uma matriz de feromônio/heurística especializada em cada objetivo. Recentemente, alguns trabalhos foram publicados comparando ACOs multiobjetivo (LÓPEZ-IBÁÑEZ; PAQUETE; STÜTZLE, 2004; LÓPEZ-IBÁÑEZ; STÜTZLE, 2010a, 2010b, 2000), e apontam que o desempenho de cada abordagem varia de acordo com o problema e o tipo de instância analisada.

Neste trabalho, um algoritmo baseado em colônias de formigas multiobjetivo é criado e comparado com a literatura do Caminho mais Curto Multiobjetivo (MSP, do inglês *Multi-objective Shortest Path Problem*). Em seguida, uma análise experimental é realizada, levando em consideração possíveis variações utilizadas na literatura de ACOs multiobjetivo. Assim, diversas versões algorítmicas são produzidas. Por último, o conceito de regulação de múltiplas espécies por disponibilidade de comida é introduzido. Esta metáfora consiste em utilizar mais de um tipo de formiga no mesmo algoritmo, cada um com uma estratégia diferente que espera-se ter um bom desempenho em um conjunto de instâncias específico. Nenhuma interação é permitida entre as diferentes espécies. Contudo, como achar boas soluções significa achar comida, permite-se que as colônias cresçam ou diminuam de acordo com seu desempenho.

Para a avaliação de cada versão algorítmica, uma metodologia de experimentação específica para o contexto multiobjetivo é utilizada (KNOWLES; THIELE; ZITZLER, 2006), incluindo ranking de dominância, indicadores de qualidade e testes estatísticos. Um conjunto de 18 instâncias é utilizado (MSPP, 2010), oriundas de dois tipos de grafos distintos (cada grupo com 9 instâncias). Os resultados mostram que a primeira versão do algoritmo

proposto apresenta um desempenho significativamente melhor que um algoritmo baseado em colônias de formigas encontrado na literatura do MSP (HÄCKEL et al., 2008), e que o NSGA-II (DEB et al., 2002), um algoritmo evolucionário bem conhecido. A análise experimental feita em seguida revela que, enquanto para um tipo de instância o algoritmo proposto continua gerando os melhores conjuntos de aproximação, para o outro grupo diferentes abordagens conseguem superá-lo. No entanto, nenhuma das tentativas de integração das características de ambas as versões obtém sucesso. Por fim, mostra-se que o uso de duas espécies permite tal ambivalência, e que, quando a regulação por disponibilidade de comida é aplicada, o algoritmo consegue gerar conjuntos de aproximação tão bons quanto os gerados pelas versões com apenas uma espécie.

Neste trabalho, o Capítulo 2 destina-se a revisar conceitos teóricos referentes à otimização multiobjetivo e a apresentar o Problema do Caminho mais Curto Multiobjetivo, detalhando tanto sua formulação quanto seu estado da arte. O capítulo seguinte dedica-se a percorrer a literatura de Colônias de Formigas aplicadas ao contexto multiobjetivo, incluindo a revisão de artigos que comparam as diferentes propostas encontradas. No Capítulo 4, a primeira versão algorítmica é proposta, denominada *GRACE*. Ainda neste capítulo, compara-se o algoritmo GRACE a dois algoritmos da literatura. O Capítulo 5 tem por objetivo analisar diferentes abordagens propostas para Colônias de Formigas Multiobjetivo aplicadas ao algoritmo GRACE. Várias abordagens são descritas e comparadas, incluindo o GRACE original. Por fim, o conceito de regulação de múltiplas espécies por disponibilidade de comida é proposto no Capítulo 6, e duas comparações são realizadas: uma contra as melhores versões do experimento realizado no Capítulo 5, e outra contra os mesmos dois algoritmos da literatura utilizados no experimento do Capítulo 4. Considerações finais e trabalhos futuros são discutidos no Capítulo 7.

2 O Problema do Caminho mais Curto Multiobjetivo

O interesse crescente pela área de otimização multiobjetivo deve-se, principalmente, ao fato de que modelos deste tipo conseguem representar de forma mais coerente necessidades do mundo real. Nas palavras de Corne et al. (2003), a vida real “raramente é caracterizada como uma posição em uma linha de números reais”. A opção por modelos com mais de um critério a serem otimizados, no entanto, implica uma complexidade teórica maior. Por esse motivo, inicialmente apresentamos tais aspectos, para em seguida detalhar o Problema do Caminho mais Curto Multiobjetivo.

2.1 Otimização Multiobjetivo

Em contextos multiobjetivo, geralmente não existe uma solução que seja simultaneamente ótima para todos os critérios, mas soluções que apresentam vantagens e desvantagens entre si. Em um problema com k objetivos, uma solução não-dominada é uma solução que não pode ser considerada pior que outras para $k - 1$ objetivos, e que é melhor em pelo menos um. Em um problema de minimização, por exemplo, diz-se formalmente que:

$$s_1 \prec s_2 \text{ sse para } i = 1, \dots, k, z^i(s_1) \leq z^i(s_2), \text{ e } \exists i \text{ tal que } z^i(s_1) < z^i(s_2) \quad (2.1)$$

onde z^i refere-se ao valor da i -ésima função objetivo. A Tabela 1 lista alguns tipos de relação de dominância entre soluções utilizados na literatura.

Para cada problema de otimização multiobjetivo, um conjunto de soluções que são não-dominadas em relação a quaisquer outras soluções possíveis formam o chamado conjunto Pareto-ótimo, e as soluções contidas neste são chamadas soluções eficientes. Assim, solucionar um problema de otimização multiobjetivo significa recuperar seu conjunto Pareto-ótimo. Como o tamanho desses conjuntos tende a crescer exponencialmente com

Tabela 1: Relações de dominância entre soluções (KNOWLES; THIELE; ZITZLER, 2006).

| Relação | Interpretação no espaço objetivo | |
|--------------------|----------------------------------|--|
| Dominância estrita | $z^1 \prec\prec z^2$ | z^1 é melhor que z^2 em todos os objetivos |
| Dominância | $z^1 \prec z^2$ | z^1 não é pior que z^2 em todos os objetivos e é melhor em pelo menos um |
| Dominância fraca | $z^1 \preceq z^2$ | z^1 não é pior que z^2 em todos os objetivos |
| Incomparável | $z^1 \parallel z^2$ | Nem $z^1 \preceq z^2$ nem $z^2 \preceq z^1$ |
| Indiferente | $z^1 \sim z^2$ | z^1 tem o mesmo valor que z^2 em cada objetivo |

o número de objetivos, algoritmos heurísticos são usados para gerar conjuntos de aproximação em lugar de tentar recuperar todas as soluções eficientes de um problema.

Cada solução de um problema com mais de um critério possui não somente um valor de função objetivo, mas um vetor de valores, o chamado vetor objetivo. O conjunto dos vetores objetivos das soluções presentes no conjunto Pareto-ótimo é denominado fronteira de Pareto. Um otimizador multiobjetivo é considerado eficiente quando consegue aproximar a fronteira de Pareto de um problema. Isto pode significar recuperar parte das soluções eficientes de forma bem distribuída ao longo do espaço objetivo, ou gerar conjuntos cujo nível de optimalidade e diversidade atinjam um nível desejado. As relações de dominância apresentadas para a comparação de soluções podem ser extendidas para a comparação de conjuntos de aproximação gerados por otimizadores, conforme a Tabela 2. Esse tipo de relação é importante para a comparação de diferentes algoritmos que trabalhem em um mesmo problema de otimização multiobjetivo.

Tabela 2: Relações de dominância entre conjuntos de soluções em um problema multiobjetivo (KNOWLES; THIELE; ZITZLER, 2006).

| Relações | Interpretação no espaço objetivo | |
|--------------------|----------------------------------|--|
| Dominância estrita | $A \prec\prec B$ | Cada $z^2 \in B$ é estritamente dominado por pelo menos um $z^1 \in A$ |
| Dominância | $A \prec B$ | Cada $z^2 \in B$ é dominado por pelo menos um $z^1 \in A$ |
| Melhor | $A \triangleleft B$ | Cada $z^2 \in B$ é fracamente dominado por pelo menos um $z^1 \in A$ e $A \approx B$ |
| Dominância fraca | $A \preceq B$ | Cada $z^2 \in B$ é fracamente dominado por pelo menos um $z^1 \in A$ e $A \approx B$ |
| Incomparável | $A \parallel B$ | Nem $A \preceq B$ nem $B \preceq A$ |
| Indiferente | $A \sim B$ | $A \preceq B$ e $B \preceq A$ |

Uma abordagem comum em problemas multiobjetivo é o uso de escalarizações, que consiste em atribuir pesos para cada um dos objetivos a serem otimizados, transformando

assim o problema original em uma versão mono-objetivo. Tal abordagem pode ser utilizada quando, de antemão, sabe-se qual a ponderação correta que deve-se atribuir a cada objetivo de acordo com o propósito do tomador de decisões. No entanto, Mooney e Winstansley (2006) afirmam que uma correta determinação desses pesos pode ser uma tarefa muito difícil. Corroborando essa idéia, Pereira (2004) afirma que “ponderações devem ser feitas por analistas experientes”.

Um uso importante e benéfico das escalarizações aparece quando, em lugar de tratar todo o problema como se fosse mono-objetivo, os algoritmos fazem uso direcionado dessa técnica. O conjunto de soluções não-dominadas de um problema de otimização combinatoria multiobjetivo pode ser dividido em dois subconjuntos: (a) o conjunto das soluções suportadas, que podem ser determinadas com o auxílio de escalarizações, e; (b) o conjunto das soluções não-suportadas, que precisam ser encontradas de outras maneiras. Assim, muitos algoritmos utilizam-se de escalarizações para inicialmente recuperar as soluções suportadas de um problema (algumas ou todas), e em seguida buscar as demais soluções do conjunto Pareto-ótimo. Tal abordagem é denominada procedimento em duas fases (do inglês, *two-phases method*), e foi inicialmente proposta por Ulungu e Teghem (1995).

Um aspecto teórico adicional importante é que, para o contexto biobjetivo, soluções não-suportadas sempre serão encontradas em regiões triangulares delimitadas por duas soluções suportadas consecutivas (veja a Figura 1). Assim sendo, recuperar inicialmente soluções suportadas através de escalarizações torna-se um processo bastante vantajoso por restringir o espaço de busca a ser percorrido por um algoritmo ao procurar o restante do conjunto Pareto-ótimo. A generalização desse procedimento para qualquer quantidade de objetivos, no entanto, não é trivial.

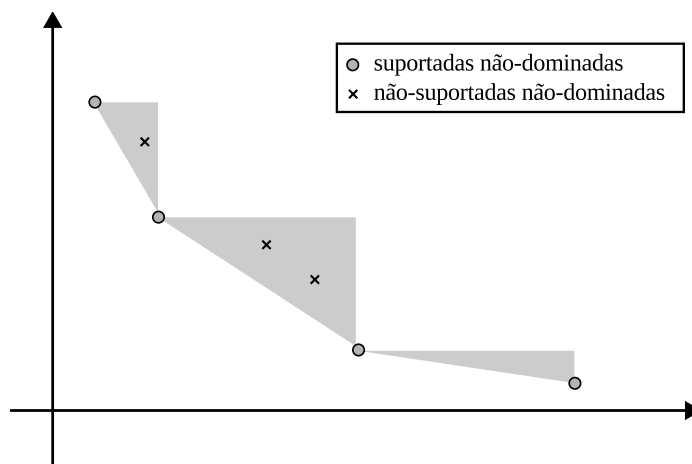


Figura 1: As soluções não-suportadas de um problema biobjetivo encontram-se nas regiões triangulares delimitadas por soluções suportadas consecutivas (RAITH; EHRGOTT, 2009).

A avaliação de otimizadores propostos para problemas com múltiplos objetivos é um assunto complexo, e diversas propostas são encontradas na literatura. Neste trabalho, descreveremos a abordagem apresentada por Knowles, Thiele e Zitzler (KNOWLES; THIELE; ZITZLER, 2006). Inicialmente, os conjuntos de aproximação gerados por diferentes otimizadores são comparados através de rankings de dominância. Dada uma lista de q otimizadores, r_i ($i = 1, \dots, q$) execuções independentes de cada e uma coleção \mathbf{C} contendo todos os conjuntos de aproximação C_j^i ($i = 1, \dots, q, j = 1, \dots, r_i$) gerados a cada execução do otimizador correspondente, cada conjunto C_j^i recebe um ranking, $rank(C_j^i)$, igual a um mais o número de conjuntos melhores que ele. Assim, cada otimizador é caracterizado por uma amostra $(rank(C_0^i), \dots, rank(C_{r_i}^i))$, e essas amostras podem ser comparadas utilizando testes estatísticos. Knowles, Thiele e Zitzler (2006) afirmam que, caso um otimizador Q^1 apresente rankings de dominância estatisticamente inferiores que outro otimizador Q^2 , então pode-se dizer que Q^1 gera conjuntos de aproximação melhores que Q^2 , e nenhuma investigação posterior é necessária.

Caso nenhuma diferença estatística seja encontrada com os dados produzidos pela metodologia do ranking de dominância, indicadores de qualidade podem ser utilizados. Indicadores de qualidade são funções que atribuem valores qualitativos a conjuntos de aproximação. Um indicador de qualidade unário I mapeia o conjunto de todos os possíveis conjuntos de aproximação no conjunto dos números reais. Cada indicador avalia uma característica específica dos conjuntos e, portanto, combinar múltiplos indicadores é recomendado. Knowles, Thiele e Zitzler (2006) descrevem alguns destes indicadores:

1. o indicador de hipervolume (I_H) (ZITZLER et al., 2002), que calcula o hipervolume do espaço objetivo que é fracamente dominado pelo conjunto de aproximação avaliado (delimitado por um ponto de referência);
2. o indicador ϵ -binário aditivo ($I_{\epsilon+}^1$) (ZITZLER et al., 2002), que calcula o menor ϵ que deve ser acrescido a cada solução de um conjunto A_2 para que este se torne fracamente dominado por um outro conjunto A_1 ;
3. o indicador R2 unário (I_{R2}^1) (HANSEN; JASZKIEWICZ, 1998), que analisa a qualidade de um conjunto de aproximação a partir de funções utilitárias que representam as preferências do tomador de decisões.

Alguns desses indicadores de qualidade usam conjuntos de referência para comparar diferentes conjuntos de aproximação. Para a geração de conjuntos de referência, inicialmente os conjuntos de aproximação gerados por todos os otimizadores comparados são

unidos. Em seguida, ou as soluções não dominadas são excluídas, ou um conjunto de soluções que represente um desempenho médio é gerado a partir deste (KNOWLES; THIELE; ZITZLER, 2006). No entanto, quando existem algoritmos exatos que permitem recuperar o conjunto Pareto-ótimo de um problema, este pode ser usado como conjunto de referência. Ademais, Veldhuizen (1999) e Zitzler (1999) listam algumas métricas que podem ser utilizadas nessa circunstância:

1. a taxa de erro (*error ratio*), definida como a proporção entre a quantidade de soluções do conjunto de aproximação que não pertencem de fato ao conjunto Pareto-ótimo e a cardinalidade do conjunto de aproximação. Assim, uma taxa de erro igual a 0 significa que todas as soluções do conjunto de aproximação são de fato Pareto-ótimas. Por outro lado, uma taxa igual a 1 significa que nenhuma solução do conjunto de aproximação é de fato Pareto-ótima;
2. a distância geracional (*generational distance*). Seja d_i a distância euclidiana entre uma solução i do conjunto de aproximação e a solução mais próxima a i no conjunto Pareto-ótimo. A distância geracional é definida como a média das distâncias d_i , para cada solução i do conjunto de aproximação avaliado;
3. a cobertura do conjunto (*set coverage*), também conhecida como métrica \mathcal{C} . Esta métrica foi definida para comparação entre quaisquer conjuntos A e B, e calcula a proporção entre o número de soluções de um conjunto B que são fracamente dominadas pelas soluções de A e a cardinalidade de B:

$$\mathcal{C}(A, B) = \frac{|\{b \in B \mid \exists a \in A : a \preceq b\}|}{|B|} \quad (2.2)$$

Quando o conjunto Pareto-ótimo (Y^*) é usado, a métrica \mathcal{C} calcula a proporção entre o número de soluções do conjunto de aproximação que são fracamente dominadas pelas soluções do conjunto Pareto-ótimo e a cardinalidade do conjunto de aproximação. Uma vez que a dominância fraca engloba a indiferença, $\mathcal{C}(Y^*, A)$ será sempre igual a 1, sejam quais forem as soluções contidas em A. Por outro lado, $\mathcal{C}(A, Y^*)$ contabilizará somente as soluções iguais nos dois conjuntos, se equivalendo ao complemento do indicador *error ratio*. Assim, é necessário alterar a relação de dominância a ser usada. Definimos, assim, a métrica \mathcal{C}_{\prec} :

$$\mathcal{C}_{\prec}(A, B) = \frac{|\{b \in B \mid \exists a \in A : a \prec b\}|}{|B|} \quad (2.3)$$

2.2 O Problema do Caminho mais Curto Multiobjetivo

O Problema do Caminho mais Curto estudado neste trabalho é uma generalização do clássico caminho mais curto ponto-a-ponto. Raith e Ehrgott (2009) apresentam sua versão com dois objetivos utilizando uma formulação de fluxo de redes. Neste trabalho, expandimos tal formulação para lidar com qualquer número de objetivos:

$$\min \quad z(x) = \begin{cases} z_1(x) = \sum_{(i,j) \in A} c_{ij}^1 x_{ij} \\ \dots \\ z_k(x) = \sum_{(i,j) \in A} c_{ij}^k x_{ij} \end{cases} \quad (2.4)$$

$$\text{s.t.} \quad \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 1 & \text{se } i = s, \\ 0 & \text{se } i \neq s, t \\ -1 & \text{se } i = t, \end{cases} \quad (2.5)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A. \quad (2.6)$$

onde s e t são, respectivamente, os vértices de origem e de destino, c é uma matriz de custo k -dimensional, z é o vetor objetivo composto de k funções objetivo, e x_{ij} representa a inclusão ou não de uma aresta na solução.

O estado da arte do MSP contém tanto algoritmos exatos como algoritmos aproximativos baseados em metaheurísticas. Skriver (2000) propôs uma classificação para os algoritmos exatos existentes para o BSP, conforme a Figura 2. Os algoritmos de rotulamento criam rótulos associados a cada vértice, que representam os custos de caminhos possíveis entre o(s) nó(s) de origem e o vértice ao qual o rótulo pertence. Ao final da execução de um algoritmo dessa classe, cada rótulo existente no vértice de destino representa uma solução eficiente para o BSP ponto-a-ponto. A principal diferença entre os algoritmos de atribuição de rótulo e os de correção de rótulo é que os primeiros fazem uso da ordenação lexicográfica para assegurar que rótulos possam ser definidos como permanentes antes do término da execução do algoritmo. Dentre a segunda classe de algoritmos apresentada por Skriver (2000), pode-se destacar a abordagem em duas fases, que mostrou-se eficiente na comparação entre algoritmos para o BSP feita por Raith e Ehrgott (2009): resolve instâncias com até 21.000 nós em menos de 30 segundos, e com até 30.000 nós em

menos de 40 segundos. Contudo, não são encontradas na literatura do MSP propostas de generalização deste algoritmo para mais de dois objetivos.

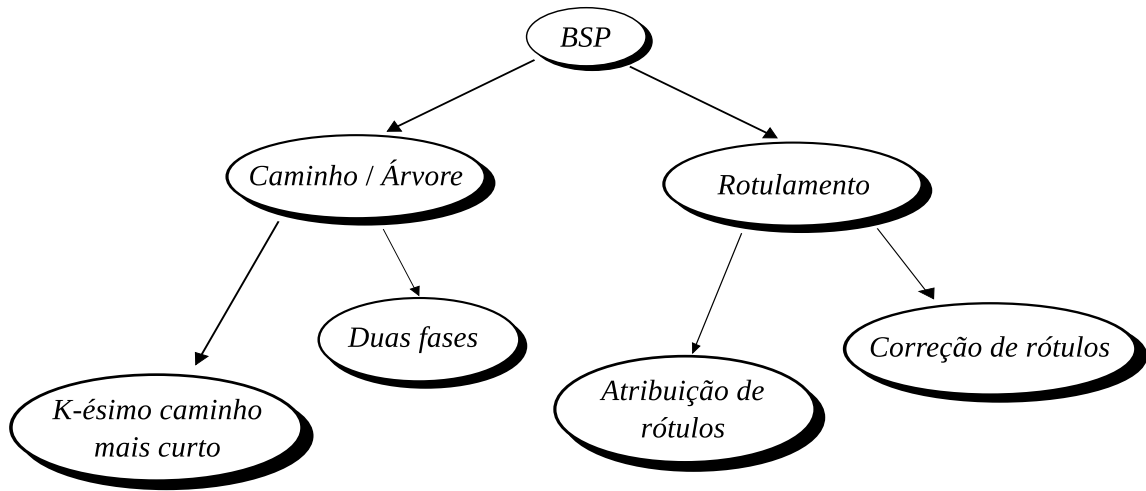


Figura 2: Classificação dos algoritmos exatos para o BSP (SKRIVER, 2000).

Dentre os algoritmos baseados em metaheurísticas, dois tipos podem ser encontrados: os algoritmos evolucionários e as colônias de formigas. Mooney e Winstansley (2006) usam duas populações em seu algoritmo evolucionário elitista. A população inicial é gerada através do mecanismo de *random walking* (COSTELLOE; MOONEY; WINSTANSLEY, 2001). Os autores usam a codificação por caminho (representação por vértices), e um operador de recombinação de um ponto com torneio binário. Um operador de mutação de caminhos também é utilizado: escolhe-se um *locus* aleatório, e substitui-se o gene encontrado naquela posição por um vértice sorteado aleatoriamente. Utilizando instâncias com três e quatro objetivos, variando de 100 a 3.500 nós, a experimentação apresentada mostra que o algoritmo proposto consegue aproximar um conjunto de referência gerado por três algoritmos: o algoritmo de Dijkstra (DIJKSTRA, 1959), um algoritmo *k*-ésimo caminho mais curto (não referenciado), e várias execuções do próprio algoritmo evolucionário. Os autores mostram também que seu algoritmo consegue encontrar as soluções extremas suportadas mais rapidamente que o algoritmo de Dijkstra em instâncias reais, mas não há informações sobre quantos objetivos estas contêm.

He, Qi e Fan (2007) também utilizam-se de duas populações em seu algoritmo evolucionário. A população inicial é gerada através de uma busca em profundidade. Os autores usam rankings de dominância e contagem de nicho para definir a adequação dos indivíduos, que são codificados com uma representação por caminhos (novamente utilizando vértices). A seleção para aplicação dos operadores genéticos é feita por meio de um torneio binário. O operador de recombinação de um ponto escolhe uma posição aleatória de um dos pais,

e caso o mesmo nó esteja presente no outro pai, os cromossomos são emparelhados nessa posição e a recombinação é efetuada (uma função reparadora elimina possíveis ciclos). O operador de mutação reconstrói o cromossomo a partir de uma posição aleatória através da busca em profundidade. Um exemplo ilustrativo mostrou a eficiência do algoritmo proposto em uma instância com 50 vértices e três objetivos.

Pangilinan e Janseens (2007) adaptaram o SPEA2 (ZITZLER et al., 2002) para o MSP usando uma implementação disponibilizada na plataforma PISA (BLEULER et al., 2003). Eles também utilizam codificação por caminhos, mas a população inicial é gerada aleatoriamente. Tanto o operador de recombinação como o de mutação são idênticos aos utilizados por (HE; QI; FAN, 2007), mas na mutação os indivíduos são reconstruídos aleatoriamente. Usando instâncias com três objetivos, os conjuntos de aproximação apresentados demonstram boa diversidade em dois dos três objetivos, mas os autores não puderam determinar a optimalidade das soluções uma vez que os conjuntos Pareto-ótimos não eram conhecidos.

Por último, Lin e Gen (2007) propuseram um algoritmo evolucionário especificamente para o BSP. Seu algoritmo com ponderações adaptativas utiliza-se de uma codificação proposta pelos próprios autores, a codificação baseada em prioridade, que permite a existência de cromossomos com tamanhos fixos. Nenhuma informação é dada sobre a geração da população inicial. A seleção de indivíduos é feita através de um sorteio por roleta. O operador de recombinação proposto pelos autores faz uso da codificação baseada em prioridade. Também são usados operadores de mutação (aleatório) e imigração (MICHAEL; STEWART; KELLY, 1991), que substitui parte da população por indivíduos gerados aleatoriamente. Por último, os autores fazem uso de dois controladores baseado em lógica fuzzy para ajustar os parâmetros de seu algoritmo automaticamente. Uma comparação foi feita reunindo implementações dos algoritmos NSGA-II (DEB et al., 2002), SPEA (ZITZLER; THIELE, 1999) e rwGA (ISHIBUCHI; MURATA, 1998), todos utilizando a mesma codificação e os mesmo operadores que o algoritmo proposto. Segundo os critérios de diversidade, cardinalidade, proporção de soluções não-dominadas e tempo computacional, os resultados mostram que a codificação baseada em prioridade e o auto-ajuste de parâmetros melhoram o desempenho do algoritmo proposto, e que o algoritmo apresentado consegue resultados melhores que os demais.

Dentre os algoritmos baseados na Otimização por Colônias de Formigas, Häckel et al. (2008) propuseram um algoritmo multi-colônia que usa escalarizações para direcionar a busca. Formigas de colônias diferentes não podem pesquisar a mesma região do espaço objetivo. A informação heurística é fornecida por um algoritmo de programação dinâmica

proposto, denominado LAH (*Look-Ahead Heuristic*). Quanto ao feromônio, equações são apresentadas como se mais de uma matriz fosse utilizada, mas os autores afirmam utilizar somente uma. Os experimentos realizados mostram que o uso da LAH melhora o desempenho do algoritmo, e que o algoritmo proposto encontra soluções bem distribuídas ao longo do espaço objetivo em comparação com um algoritmo de programação dinâmica não referenciado pelos autores.

O outro algoritmo baseado na mesma metaheurística foi proposto por Ghoseiri e Nadjari para o BSP (GHOSEIRI; NADJARI, 2010). Os autores utilizam somente uma colônia. Duas matrizes de feromônio são usadas, e cada uma se especializa em um objetivo. O algoritmo contém duas fontes de informação heurística: os pesos das arestas normalizados e o número de nós até o vértice de destino. Escalarizações são usadas para agregar tanto o feromônio como a informação heurística. Ademais, as matrizes de feromônio são atualizadas a cada passo do processo construtivo. A experimentação realizada pelos autores usou instâncias biobjetivo geradas pelo NetMaker (SKRIVER; ANDERSEN, 2000). Os resultados mostraram que o algoritmo proposto consegue gerar um conjunto de aproximação bem distribuído ao longo do espaço objetivo, mas as fronteiras de aproximação apresentadas graficamente não contêm nenhuma solução eficiente. O algoritmo foi comparado com um algoritmo de correção de rótulos (não identificado), e mostrou-se computacionalmente mais rápido.

Apesar da grande quantidade de algoritmos baseados em metaheurísticas encontrados, não é possível encontrar na literatura nenhuma comparação que permita avaliar o desempenho de cada proposta. No próximo capítulo, a Otimização por Colônia de Formigas é detalhada profundamente, com foco na aplicação ao contexto multiobjetivo.

3 Otimização por Colônia de Formigas

A Otimização por Colônia de Formigas pertence à classe das metaheurísticas bioinspiradas. Esta abordagem faz uso do conceito de inteligência de nuvem, a habilidade de grupos de se comunicarem mesmo na ausência de uma coordenação central, através de um estímulo que é ao mesmo tempo físico e local. Este fenômeno é chamado de estigmergia (tradução livre do inglês *stigmergy*) (GRASSÉ, 1959), e no caso das formigas, o estímulo é denominado feromônio (DORIGO; SOCHA, 2006). Ao procurar comida, formigas podem identificar e depositar feromônio sobre as trilhas que percorrem para que outras formigas as sigam. Uma vez que o feromônio evapora com o tempo, rotas mais curtas tem maior probabilidade de serem reforçadas que as mais longas. Assim, as formigas tendem a convergir para os caminhos mínimos. Na tentativa de mimetizar esse comportamento natural, o *Ant System* (AS) foi proposto (DORIGO, 1992), e posteriormente transformado no *Ant Colony System* (ACS) (DORIGO; GAMBARDELLA, 1997).

Em algoritmos baseados em ACO, soluções são construídas iterativamente por agentes chamados formigas. Cada formiga constrói uma solução e é capaz de avaliar seu estado, isto é, calcular o valor objetivo de sua solução atual (completa ou parcial). Enquanto constrói a solução, a formiga toma decisões baseada na informação disponível para as possíveis escolhas. Tal informação vem da experiência de outras formigas, através da quantidade de feromônio depositada em um determinado caminho, e da sua própria experiência, através de uma heurística. A probabilidade de uma formiga escolher a transição do estado i para o estado j , $p(e_{ij})$, é dada pela Equação 3.1 onde \mathcal{N}_i representa o conjunto de estados vizinhos de i , π_{ij} e n_{ij} representam, respectivamente, a quantidade de feromônio e a informação heurística associada à transição entre os estados i e j , e α e β são parâmetros que ponderam a importância da informação que vem do feromônio e da heurística, respectivamente. Cada vez que uma solução é construída, um depósito de feromônio pode ser efetuado, que evapora

com o tempo.

$$p(e_{ij}) = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{h \in \mathcal{N}_i} \tau_{ih}^\alpha \cdot \eta_{ih}^\beta} & \text{se } j \in \mathcal{N}_i \\ 0 & \text{caso contrário} \end{cases} \quad (3.1)$$

A quantidade de feromônio na transição e_{ij} em uma iteração $t + 1$ é calculada conforme a Equação 3.2, onde ρ é a taxa de evaporação. O depósito de feromônio $\Delta\tau_{ij}$ é a soma dos depósitos de feromônio de todas as formigas em uma dada iteração t . Cada um desses depósitos no *Ant System* é calculado usando a Equação 3.3, onde Q é um parâmetro do algoritmo e $L^a(t)$ é o comprimento do caminho π_a construído por uma formiga a em uma iteração t :

$$\tau_{ij}(t + 1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t, t + 1) \quad (3.2)$$

$$\Delta\tau_{ij}^a(t, t + 1) = \begin{cases} \frac{Q}{L^a(t)} & e_{ij} \in \pi_a \\ 0 & \text{caso contrário} \end{cases} \quad (3.3)$$

Duas mudanças principais foram propostas para transformar o AS no ACS. Primeiro, durante a avaliação das transições possíveis, as formigas podem escolher a melhor transição disponível em lugar de usar a estocasticidade. Essa alteração é chamada de regra proporcional pseudo-aleatória. A segunda mudança refere-se à atualização do feromônio. No AS, todas as formigas efetuam a atualização juntas ao terminar de construir suas soluções. Para o ACS, Dorigo e Gambardella (1997) propuseram uma atualização local, isto é, cada formiga pode efetuar seu depósito tão logo termine o seu processo construtivo. O alvo desta adaptação é permitir que formigas em uma mesma iteração possam usar o conhecido aprendido por outras, em lugar de efetuar buscas baseadas somente na informação das iterações anteriores.

Várias propostas de Colônias de Formigas para o contexto multiobjetivo, chamadas MOACO (Otimização por Colônias de Formigas Multiobjetivo, do inglês, *Multi-objective Ant Colony Optimization*), têm sido apresentadas recentemente. Algumas consideram os objetivos separadamente, ou estabelecem ordem de prioridade entre os mesmos. Neste trabalho, a revisão será limitada aos trabalhos que consideram os objetivos igualmente importantes (IREDI; MERKLE; MIDDENDORF, 2001; GUNTSCHE; MIDDENDORF, 2003; DOERNER et al., 2004; DOERNER; HARTL; REIMANN, 2001; ALAYA; SOLNON; GHÉDIRA, 2007).

3.1 Colônias de formigas multiobjetivo

Iredi, Merkle e Middendorf (2001) propuseram um algoritmo baseado em múltiplas colônias de formigas para um problema de alocação. Este algoritmo utiliza múltiplas colônias para buscar soluções em diferentes regiões do espaço objetivo. Cada formiga é associada a um vetor de escalarização que é usado para ponderar o feromônio e a informação heurística. Uma vez que lidam com um problema biobjetivo, duas matrizes de feromônio são utilizadas, uma por objetivo. Existem duas estratégias possíveis para a seleção de quais formigas poderão depositar feromônio: (i) formigas que encontram soluções não-dominadas em relação a arquivos locais (um por colônia), ou; (ii) formigas que encontram soluções não-dominadas em relação a um arquivo global. Os autores afirmam que a primeira opção leva a resultados iguais a abordagens com múltiplos reinícios, o que desejam evitar. Por esse motivo, a segunda estratégia é adotada com dois métodos de atualização: atualização por origem e atualização por região. Na atualização por origem as formigas só podem depositar feromônio na matriz de feromônio de sua própria colônia. Quando a atualização por região é utilizada, as soluções não-dominadas encontradas durante uma iteração são ordenadas lexicograficamente e divididas em r regiões (r é o número de colônias). Cada formiga atualiza, então, a matriz de feromônio da colônia i , $l \leq i \leq r$, onde i representa a região onde a formiga encontrou sua solução. Os autores também propuseram zonas de interseção entre as colônias, operacionalizadas através de vetores de escalarização comuns a colônias vizinhas. A experimentação realizada em seis instâncias biobjetivo mostra que a versão com 10 colônias combinada com a atualização por região produz resultados melhores que o algoritmo com apenas uma colônia. Os autores mostram ainda que, dependendo da quantidade de colônias usadas, permitir que todas as formigas busquem o espaço objetivo inteiro apresenta um desempenho pior que dividi-lo em intervalos disjuntos. Ademais, o uso das zonas de interseção melhora os resultados de maneira geral. Para duas colônias, essa diferença é percebida claramente. Quando cinco colônias são usadas, ela se torna menos significativa, e praticamente desaparece quando dez colônias são adotadas.

Guntsch e Middendorf (2003) propuseram um algoritmo de colônia de formigas baseado em população para o mesmo problema de alocação abordado por (IREDI; MERKLE; MIDDENDORF, 2001). Em seu trabalho, uma população P é obtida a partir de uma superpopulação Q , pela escolha das k soluções mais próximas de uma solução aleatoriamente escolhida π (k é um parâmetro do algoritmo). O feromônio utilizado pelo algoritmo é extraído de P : cada vez que uma solução é inserida neste conjunto, um depósito de

feromônio é efetuado para cada aresta presente nesta solução. Quando uma solução é removida deste conjunto, sua influência também é removida. Duas matrizes de feromônio são utilizadas, uma por objetivo. Os autores relatam testes para populações de tamanho 1, 3 e 5, concluindo que os melhores conjuntos de aproximação para uma determinada instância biobjetivo são gerados quando o tamanho 1 é usado, enquanto os resultados para os tamanhos 3 e 5 são praticamente iguais. Para uma segunda instância biobjetivo, os tamanhos 3 e 5 continuam a não apresentar diferença significativa. Para essa instância, porém, o uso da população de tamanho 1 obtém um desempenho pior que os demais.

Doerner, Hartl e Reimann (2001) propuseram um MOACO baseado na idéia da competição, e o testaram para um problema de transporte bi-objetivo. As COMPETants, como foram chamadas, consistem em um algoritmo com duas colônias, cada uma com foco em um dos objetivos, e no conceito de espiãs, formigas que podem analisar o feromônio da outra colônia e decidir se vão utilizá-lo. A cada iteração, as formigas escolhem a qual colônia pertencerão. Quanto mais baixa a qualidade média das soluções de cada colônia, maior a chance desta de receber formigas. Em seu trabalho, a função objetivo é calculada como uma função ponderada dos custos de rota e do tamanho da frota. Os autores afirmam que o uso de formigas espiãs e das colônias de tamanho adaptativo levam a resultados melhores do que quando estas estratégias não são adotadas.

Doerner et al. (2004) propuseram o P-ACO, um algoritmo de colônia de formigas baseado em população para o problema de seleção de portfólio multiobjetivo. Este algoritmo faz uso de escalarizações para o procedimento construtivo das formigas, que são geradas aleatoriamente no intervalo $[0,1)$. O algoritmo proposto utiliza a regra proporcional pseudo-aleatória. As soluções não-dominadas encontradas pelo procedimento construtivo são armazenadas em um arquivo global. Uma formiga só deposita feromônio caso encontre a melhor ou a segunda melhor soluções por objetivo em um iteração. Uma penalização também é realizada (atualização negativa de feromônio). Um vetor de feromônio é utilizado para cada objetivo. Uma comparação foi feita com implementações do *Pareto Simulated Annealing* (PSA) (CZYŻAK; JASZKIEWICZ, 1998) e do NSGA (SRINIVAS; DEB, 1994) usando 18 instâncias contendo de 5 a 10 objetivos, com 20 e 30 projetos. Todos os testes usaram o número de soluções e o número de soluções eficientes como métricas (os conjuntos Pareto-ótimo foram recuperados através de algoritmos exatos). Para instâncias com 20 projetos, o PSA apresenta desempenho melhor que os outros algoritmos para tempos de execução pequenos. À medida que o tempo passa, o P-ACO torna-se o algoritmo de melhor desempenho. Quando o número de projetos é elevado para 30, o P-ACO sempre produz resultados melhores que os demais otimizadores. Esses resultados também

são verdadeiros para uma instância real com seis objetivos. Após 40 minutos de execução, o PSA consegue gerar conjuntos de aproximação maiores, porém com menor número de soluções eficientes. Posteriormente, Doerner et al. (2006) melhoraram esse otimizador acrescentando um algoritmo de pré-processamento de Programação Linear Inteira. Esse pré-processamento permite ao algoritmo encontrar as soluções eficientes suportadas *a priori*, e utilizá-las para aquecer a matriz de feromônio. Os autores afirmam que, para esse aquecimento, o uso de todo o conjunto de soluções suportadas é melhor que o uso de apenas a melhor e a segunda melhor soluções por objetivo. A comparação foi expandida para incluir uma implementação da Busca Tabu Multiobjetivo (MOTS) (HANSEN, 2000), e mostrou que o P-ACO apresenta desempenho superior para a instância real testada anteriormente.

Alaya, Solnon e Ghédira (2007) apresentaram quatro variações de um MOACO genérico parametrizado pelo número de colônias e de trilhas de feromônio. Dado um problema com k objetivos, a primeira variação contém $k + 1$ colônias. Dentre estas, k colônias focalizam um objetivo cada, e possuem matriz de feromônio e fonte de informação heurística próprias. A colônia extra trabalha com todos os objetivos simultaneamente. Esta colônia escolhe qual matriz de feromônio utilizará, agrega a informação heurística de todas as colônias e atualiza todas as matrizes de feromônio. Para as colônias que focam em apenas um objetivo, somente a formiga que encontrou a melhor solução em relação ao objetivo correspondente efetua depósito de feromônio. A segunda variação é similar à primeira, com exceção da última colônia, que desta vez agrega o feromônio de todas as outras colônias. A terceira variante usa apenas uma colônia e uma matriz de feromônio, e a informação heurística é obtida através da soma das informações heurísticas de todos os objetivos. Nessa versão, somente formigas que encontrem soluções não-dominadas podem atualizar o feromônio. Por fim, a quarta variação também usa apenas uma colônia, mas contém uma matriz de feromônio para cada objetivo. A matriz a ser usada é sorteada aleatoriamente a cada iteração, e a informação heurística é calculada como na terceira variação. A formiga que encontra a melhor solução por objetivo efetua a atualização de feromônio da matriz correspondente. A comparação feita pelos autores utilizou o Problema da Mochila Multiobjetivo, e vários algoritmos evolucionários multiobjetivos. Utilizando a métrica C (ZITZLER; THIELE, 1999), os autores concluem que as versões com apenas uma colônia são melhores de forma geral que as versões com múltiplas colônias. O desempenho da quarta variação é semelhante ao da terceira, mas a primeira gera conjuntos de aproximação melhores para algumas instâncias. O mesmo acontece entre a primeira e a segunda variação: inicialmente, o desempenho de ambas se equipara, mas à medida que as instân-

cias crescem, a primeira versão se destaca. Na segunda parte do experimento, a quarta variação apresenta desempenho superior à maior parte dos algoritmos evolucionários utilizados, mas nenhuma diferença significativa é percebida em relação ao SPEA (ZITZLER; THIELE, 1999).

3.2 Estudos comparativos

López-Ibáñez, Paquete e Stützle (2004) apresentam uma discussão de diversas estratégias para MOACOs. Uma investigação experimental foi realizada utilizando o Problema de Alocação Quadrática Bi-objetivo (bQAP, do inglês, *Bi-objective Quadratic Assignment Problem*). Para tanto, o *MIN-MAX Ant System (MMAS)* (STÜTZLE; HOOS, 2000) foi utilizado como algoritmo base sobre o qual as diferentes alternativas foram aplicadas. Nenhuma informação heurística foi utilizada, devido à característica do problema. Somente a formiga que encontrava a melhor solução podia atualizar o feromônio. A estratégia de atualização por região foi adotada quando múltiplas colônias foram utilizadas. Também foi testada a hibridização com procedimentos de busca local. Usando instâncias geradas por (KNOWLES; CORNE, 2003) e os indicadores ϵ -binário e ϵ -unário (ZITZLER et al., 2002) os autores reportam que os resultados dependem fortemente da correlação dos objetivos, mas que o uso da busca local melhora o desempenho geral do algoritmo. Os diferentes MOACOs também foram comparados com a Busca Tabu Robusta (w-RoTS) (TAILLARD, 1991). Enquanto esta produziu conjuntos de aproximação melhores para instâncias não estruturadas, o contrário ocorreu nas instâncias estruturadas.

López-Ibáñez e Stützle (2010a) apresentam uma análise experimental de MOACOs para o Problema de Caixeiro Viajante Bi-Objetivo (bTSP, do inglês, *Bi-objective Traveling Salesman Problem*). Os autores incluem informação de duas heurísticas, cada uma especializada em um objetivo. Quando uma matriz de feromônio é usada, as heurísticas são ponderadas igualmente e somadas. Caso contrário, as heurísticas são ponderadas de acordo com escalarizações, e então somadas. Experimentos foram realizados com 3 instâncias Euclidianas contendo 500 cidades. Os autores concluem que a correlação dos objetivos não interfere tão decisivamente como no bQAP, e que mais uma vez o uso de busca local é determinante para a eficiência do algoritmo. Eles também observam que, na ausência da busca local, versões com apenas uma matriz de feromônio obtêm melhor desempenho na região central do espaço objetivo, enquanto as versões com múltiplas matrizes obtêm resultados melhores nas regiões extremas. O uso de múltiplas colônias melhora os resultados nas regiões onde as versões com apenas uma colônia apresentavam desempenho fraco.

García-Martínez, Cordón e Herrera (2007) revisam várias propostas de MOACO e propõem uma taxonomia de acordo com o número de soluções retornadas, o número de heurísticas utilizadas e o número de trilhas de feromônio que o algoritmo mantém. Os autores implementam adaptações de muitos desses algoritmos para o bTSP, e realizam uma comparação com outras abordagens, incluindo o NSGA-II (DEB et al., 2002) e o SPEA2 (ZITZLER; LAUMANN; THIELE, 2001). A experimentação realizada mostra que esses algoritmos evolucionários são mais rápidos, porém seu desempenho é fraco quando à dominância das soluções. Dentre os MOACOs, alguns geram conjuntos bem distribuídos, que às vezes são dominados por conjuntos concentrados na região central gerados por outros otimizadores. Os autores também relatam que a taxonomia proposta não é determinate para o desempenho do algoritmo, uma vez que algoritmos classificados em uma mesma família apresentam comportamentos distintos. Angus e Woodward (2009) apresentam um estudo teórico com o alvo de revisar a taxonomia apresentada por (GARCÍA-MARTÍNEZ; CORDÓN; HERRERA, 2007). Os autores classificam os algoritmos existentes de acordo com: (i) o número de matrizes de feromônio; (ii) como as ponderações são usadas no procedimento construtivo; (iii) as formas de avaliar as soluções; (iv) os métodos de atualização de feromônio, e; (v) como as soluções são armazenadas. Nenhuma experimentação foi realizada.

Por fim, López-Ibáñez e Stützle (2010b) também compararam os MOACOs existentes usando o bTSP e propuseram um *framework* para a criação deste tipo de algoritmo (LÓPEZ-IBÁÑEZ; STÜTZLE, 2000). Para cada algoritmo, os autores analisam o uso de diferentes abordagens. Para o P-ACO (DOERNER et al., 2004), por exemplo, o uso de múltiplas heurísticas leva a resultados melhores nas regiões extremas do espaço objetivo, enquanto a versão com apenas uma heurística apresenta melhor desempenho na região central. Para o BicriterionAnt (IREDI; MERKLE; MIDDENDORF, 2001), o uso de três colônias gera conjuntos de aproximação melhores que os gerados pela versão com uma colônia. Na comparação geral, os autores relatam que o BicriterionAnt (IREDI; MERKLE; MIDDENDORF, 2001), apresenta desempenho melhor que os outros tanto em dominância como em diversidade, mas que para regiões extremas alguns otimizadores o superam (em termos de dominância). O *framework* geral para MOACOs proposto mostra que o projeto deste tipo de algoritmo pode variar quanto a: (i) o número de colônias (N^{col}); (ii) a estratégia de atualização de feromônio (origem ou região), o máximo número de formigas que poderão efetuar depósito de feromônio (N^{upd}), e o critério de seleção (não dominância, melhor-por-objetivo ou melhor-por-objetivo-usando-escalarização); (iii) o número de ponderações por colônia ($|\Lambda|$), e se todas serão utilizadas a cada iteração ou se somente uma

será compartilhado por todas as formigas; (iv) o número de matrizes de feromônio ($[\pi]$) e a quantidade de heurísticas ($[\eta]$), e; (v) como as ponderações são utilizadas agregar as informações de diferentes fontes.

Neste capítulo, diversas propostas de algoritmos multiobjetivos baseados em colônias de formigas foram revisadas. Fica claro que a eficiência de cada abordagem varia de acordo com o problema e o tipo de instâncias usadas. No capítulo seguinte, o primeiro algoritmo proposto neste trabalho é apresentado, e uma experimentação é realizada para avaliá-lo.

4 GRACE: o algoritmo proposto

Na revisão realizada no capítulo anterior, percebe-se a grande quantidade de questões de projeto envolvidas no desenvolvimento de um algoritmo multiobjetivo baseado em colônias de formigas. Neste capítulo, apresentamos o GRACE (Colônia de Formigas Geracionais Aleatorizadas, do inglês *Generational Randomized Ant Colony Enhancement*), um MOACO em duas fases criado para o Problema do Caminho mais Curto Multiobjetivo. Inicialmente, a busca por soluções suportadas é apresentada. Em seguida, os ciclos geracionais das formigas são detalhados. Por fim, a experimentação realizada é relatada, destacando inicialmente detalhes da metodologia utilizada.

4.1 O algoritmo GRACE

Proposto como um algoritmo em duas fases, o algoritmo GRACE inicialmente faz uso de um procedimento de busca por soluções suportadas denominado *Logos* (Busca Logarítmica no Espaço Objetivo, do inglês, *Logarithmic Objective Space Search*), proposta em (BEZERRA et al., 2011). Essa estratégia foi desenvolvida para contextos com dois ou três objetivos (2 e 3-Logos, respectivamente). A 2-Logos divide iterativamente cada região do espaço de busca em duas sub-regiões, lembrando uma função logarítmica. O *framework* geral da 2-Logos é apresentado no Algoritmo 1. Em sua primeira execução, duas soluções s_i e s_f são obtidas considerando cada objetivo separadamente. Cada solução corresponde a solucionar uma versão escalarizada do MSP com vetores de escalarização $[1;0]$ e $[0;1]$. De forma geral, soluções s_i e s_f correspondem respectivamente aos pontos (x_i, y_i) e (x_f, y_f) do espaço objetivo. Um vetor de escalarização calculado como o ponto médio das coordenadas (x_i, y_i) e (x_f, y_f) é então gerado: $(x_{\text{pontoMedio}}, y_{\text{pontoMedio}})$. A versão escalarizada do MSP para essa ponderação é solucionada, retornando a solução s_{mid} (linha 1). Se essa solução for não-dominada em relação ao arquivo global, o procedimento é chamado recursivamente para (s_i, s_{mid}) e (s_{mid}, s_f) (linhas 2-4). Caso contrário, a recusaõ para. Essa estratégia é similar às buscas apresentadas por (STEINER; RADZIK, 2003) e (RAITH; EHRGOTT, 2009).

Algoritmo 1 Procedimento 2-Logos

Entrada: arquivo S , solução s_i , solução s_f

- 1: $s_{mid} = \text{Dijkstra}(\text{pontoMedio}(s_i.\text{pesos}, s_f.\text{pesos}))$;
- 2: **se** novaSolução(S, s_{mid}) **então**
- 3: 2-Logos(S, s_i, s_{mid});
- 4: 2-Logos(S, s_{mid}, s_f);
- 5: **fim se**

Analogamente, a 3-Logos inicialmente encontra as soluções eficientes suportadas referentes aos vetores canônicos utilizando o algoritmo de Dijkstra (DIJKSTRA, 1959). Essas três soluções são consideradas vértices de um triângulo onde as arestas são as linhas que conectam as soluções de entrada. Cada aresta é escaneada pelo procedimento 2-Logos (linha 1). Após a execução da 2-Logos em todas as três arestas, o centróide do triângulo é calculado (linha 2), e caso a solução obtida para essa ponderação seja nova no arquivo S de soluções não-dominadas, a 3-Logos é chamada recursivamente para os três subtriângulos (linhas 4-6). Ambos os procedimentos 2-Logos e 3-Logos são exemplificados na Figura 3.

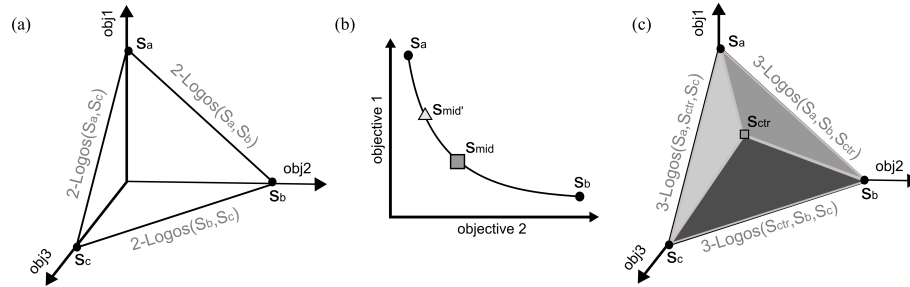


Figura 3: Exemplo de uma iteração da 3-Logos.

Algoritmo 2 Procedimento 3-Logos

Entrada: Arquivo S , solução s_a , solução s_b , solução s_c

- 1: $S = \text{adicionarSoluções}(S, 2\text{-Logos}(s_a, s_b), 2\text{-Logos}(s_a, s_c), 2\text{-Logos}(s_b, s_c))$;
- 2: $s_{ctr} = \text{Dijkstra}(\text{centroide}(s_a.\text{pesos}, s_b.\text{pesos}, s_c.\text{pesos}))$;
- 3: **se** novaSolução(S, s_{ctr}) **então**
- 4: 3-Logos(S, s_a, s_b, s_{ctr});
- 5: 3-Logos(S, s_a, s_{ctr}, s_c);
- 6: 3-Logos(S, s_{ctr}, s_b, s_c);
- 7: **fim se**

O pseudocódigo do algoritmo GRACE pode ser visto no Algoritmo 3. Apenas uma colônia é usada nesta proposta. As soluções encontradas na primeira fase (linhas 1-3) são usadas para aquecer a matriz de feromônio (linha 4): cada vez que uma aresta é encontrada, um depósito é efetuado. O algoritmo de Dijkstra fornece a informação heurística para as formigas a cada geração. Inicialmente, um vetor de escalarização é gerado aleatoriamente (linha 6). Em seguida, um grafo invertido é obtido pela inversão da direção de todas as arestas do grafo original. Por fim, o algoritmo de Dijkstra é executado no grafo

invertido (linha 7). Em outras palavras, a informação heurística disponível em cada vértice v é a distância de v até o nó de destino t , sob uma determinada escalarização. O procedimento construtivo realizado por todas as formigas dentro de uma mesma iteração usa a mesma ponderação. As formigas só podem depositar feromônio se encontrarem novas soluções não-dominadas (linhas 10-13). Cada formiga autorizada a atualizar a matriz de feromônio efetua depósitos de valor uniforme $\tau_{deposit}$. As novas soluções não-dominadas encontradas pelas formigas são adicionadas ao arquivo global ilimitado (linha 11). O laço principal (linha 5) é discretizado em grupos de R iterações, chamados *ciclos*. Se durante um ciclo uma nova solução não-dominada for encontrada, um novo ciclo de iterações é permitido e, assim, o algoritmo recebe mais tempo computacional. Essa verificação é feita nas linhas 15-17, onde o procedimento booleano *novaSolução(ciclo)* retorna verdadeiro caso uma nova solução não-dominada tenha sido gerada no ciclo correspondente.

Algoritmo 3 GRACE

Entrada: grafo G , vértice s , vértice t

```

1: soluções_extremas = Dijkstra(vetoresCanônicos());
2: suportadas = adicionarSoluções(suportadas, soluções_extremas);
3: 3-Logos(suportadas, soluções_extremas);
4: matrixFeromônio = aquecerFeromônio(suportadas);
5: para  $i = 0$ ;  $i < R$ ;  $i++$  faça
6:    $l = \text{vetorEscalarizaçãoAleatório}()$ ;
7:   heurística = informaçãoHeurística(Dijkstra(inverter( $G$ ),  $t$ ,  $l$ ));
8:   para  $j = 0$ ;  $j < n_{formigas}$ ;  $j++$  faça
9:      $s = \text{construirSolução}(\text{matrixFeromônio}, \text{heurística}, s, t, l)$ ;
10:    se novaSolução(não_suportadas,  $s$ ) então
11:      não_suportadas = adicionarSoluções(não_suportadas,  $s$ );
12:      atualizarFeromônio( $s$ );
13:    fim se
14:  fim para
15:  se ( $(i == R)$  e (novaSolução(ciclo))) então
16:     $i = 0$ ;
17:  fim se
18: fim para

```

4.2 Metodologia

Para a avaliação do algoritmo proposto, duas metodologias são usadas. Na primeira, inicialmente rankings de dominância são gerados a partir dos conjuntos de aproximação gerados em 30 execuções de cada algoritmo, e os resultados são comparados através de testes estatísticos. Para comparar mais de dois otimizadores, o teste de Kruskal-Wallis (KRUSKAL; WALLIS, 1952) é usado em uma primeira etapa, e se for encontrada diferença significativa ao nível de 0,05%, novos rankings são gerados para uma comparação dois-a-dois utilizando os testes bicaudal e unicaudal de Wilcoxon (WILCOXON, 1945). Se somente

dois otimizadores estiverem sendo comparados, não há necessidade do teste de Kruskal-Wallis, e os testes de Wilcoxon são executados diretamente. Os resultados retornados pelos testes de Wilcoxon são, por fim, comparados usando o teste de Taillard, Waelti e Zuber (2009) para comparação de proporções, novamente com nível de significância 0,05%. Neste trabalho, apenas duas casas decimais são consideradas para os p -valores retornados por este teste. Caso necessário, os indicadores de qualidade I_H e $I_{\epsilon+}^1$ são utilizados para gerar novas amostras (ver Seção 2.1), e todo o procedimento de testes estatísticos é repetido.

A segunda metodologia foi adotada durante a revisão deste trabalho, a partir da percepção que o teste de Kruskal-Wallis para múltiplas comparações se mostra mais adequado para o contexto deste experimento. Neste caso, os conjuntos Pareto-ótimo foram usados como conjuntos de referência, e os indicadores I_H , $I_{\epsilon+}^1$, *generational distance*, *error ratio* e a métrica \mathcal{C}_{\prec} são utilizados. Os valores retornados por cada indicador constituem amostras que representam cada algoritmo e são comparadas inicialmente pelo teste de Kruskal-Wallis, e em seguida pelo teste de Kruskal-Wallis para múltiplas comparações. Para ambos os testes, o nível de significância adotado é de 0.05%.

Os experimentos foram realizados usando 18 instâncias geradas por Santos (MSPP, 2010), oriundas de dois tipos de grafos distintos: *grid*, que representa uma grade, e *complete*, que contém grafos completos. Tempos de processamento fixos foram utilizados para cada instância como condição de parada. Os limites para os tempos de processamento foram estabelecidos de acordo com o tipo e o tamanho da instância. A lista de instâncias pode ser vista na Tabela 3, onde a coluna # representa o número da instância e a coluna *Tipo* mostra a identificação $\langle tipo \rangle N - \langle tamanho \rangle$, onde $\langle tipo \rangle$ representa o tipo da instância (*complete* ou *grid*) e $\langle tamanho \rangle$ representa o tamanho da instância: pequena (*small*), média (*medium*) ou grande (*large*). Ademais, $|N|$, $|A|$ e k informam, respectivamente, a quantidade de vértices, arestas e objetivos de cada instância, $t(s)$ é o tempo máximo de processamento em segundos, S é a semente usada por Santos (MSPP, 2010) para a geração da instância, e $|P_{set}|$ é a cardinalidade do conjunto Pareto ótimo, recuperado com o auxílio do algoritmo de Martins (MARTINS, 1984).

A primeira metodologia descrita nesta seção é utilizada ao longo de todo este trabalho. A segunda é aplicada somente quando o algoritmo GRACE é comparado com os outros algoritmos da literatura implementados aqui. Neste capítulo, a plataforma usada para as execuções dos algoritmos foi um processador Intel Core 2 Duo @ 2.2GHz, 1Gb de RAM, e distribuição Linux Ubuntu 9.04 32bits.

Tabela 3: Lista de instâncias. Somente os três primeiros objetivos foram usados em instâncias *large*.

| # | Tipo | $ N $ | $ A $ | k | $t(s)$ | S | $ P_{set} $ |
|----|------------------|-------|-------|-----|--------|-----|-------------|
| 1 | CompleteN-small | 25 | 600 | 3 | 5 | 45 | 31 |
| 2 | CompleteN-small | 50 | 2450 | 3 | 10 | 12 | 50 |
| 3 | CompleteN-small | 100 | 9900 | 3 | 12 | 13 | 79 |
| 4 | CompleteN-medium | 40 | 780 | 3 | 5 | 18 | 28 |
| 5 | CompleteN-medium | 120 | 14280 | 3 | 10 | 14 | 31 |
| 6 | CompleteN-medium | 200 | 39800 | 3 | 15 | 21 | 102 |
| 7 | CompleteN-large | 100 | 9900 | 6 | 8 | 1 | 34 |
| 8 | CompleteN-large | 150 | 22350 | 6 | 40 | 10 | 122 |
| 9 | CompleteN-large | 200 | 39800 | 6 | 40 | 1 | 108 |
| 10 | GridN-small | 64 | 224 | 3 | 7 | 14 | 80 |
| 11 | GridN-small | 144 | 528 | 3 | 36 | 1 | 276 |
| 12 | GridN-small | 256 | 960 | 3 | 81 | 26 | 465 |
| 13 | GridN-medium | 484 | 1848 | 3 | 100 | 1 | 1223 |
| 14 | GridN-medium | 961 | 3720 | 3 | 100 | 40 | 5674 |
| 15 | GridN-medium | 1225 | 4760 | 3 | 100 | 2 | 5562 |
| 16 | GridN-large | 121 | 440 | 6 | 60 | 41 | 175 |
| 17 | GridN-large | 484 | 1848 | 6 | 100 | 42 | 1981 |
| 18 | GridN-large | 900 | 3480 | 6 | 100 | 43 | 2726 |

4.3 Experimentação

Para avaliar o algoritmo GRACE, dois otimizadores multiobjetivo foram implementados, cada um representando uma classe de metaheurísticas existentes para o MSP. Dentre os algoritmos evolucionários, uma implementação disponível do NSGA-II (DEB et al., 2002) foi adaptada, utilizando codificação por caminhos (representação por vértices), e os operadores de mutação e recombinação usados por (PANGILINAN; JANSEENS, 2007). Diversos valores foram testados para o tamanho da população, número de gerações e taxas de recombinação e mutação. Uma vez que tempos limites fixos são usados, um número grande de gerações foi estabelecido para descobrir o tamanho máximo da população que o algoritmo poderia suportar: 200 gerações para instâncias completas e 100 para instâncias *grid*. As instâncias 1-9 foram divididas em dois grupos, uma vez que os tempos limites se mostraram críticos para a definição do tamanho da população. Para as instâncias *grid*, uma função linear $pop(x) = a.t(x) + b$ foi usada para determinar a população p de uma

instância x , de acordo com seu tempo de processamento t . A Tabela 4 mostra os valores utilizados para o teste dos tamanhos da população. Para a taxa de recombinação os valores 0,7, 0,8 e 0,9 foram testados. Já para a taxa de mutação, os valores usados foram 0,01, 0,005 e 0,001. O ranking de dominância mostrou que o uso da combinação 2500&3000 para instâncias completas gera conjuntos de aproximação significativamente melhores. Já para as instâncias do tipo *grid*, nem o ranking de dominância nem os indicadores unários apontaram diferença significativa entre os conjuntos gerados pelas diferentes configurações. Para as taxas de recombinação e mutação, nenhuma diferença significativa foi observada.

Tabela 4: Valores de parâmetros testados pro NSGA-II.

| #config | Complete 1-7 | Complete 8-9 | Grid a | Grid b |
|---------|--------------|--------------|----------|----------|
| 1 | 1000 | 1500 | 20 | 600 |
| 2 | 1500 | 2000 | 20 | 900 |
| 3 | 2500 | 3000 | 25 | 800 |

Dentre os algoritmos baseados em colônias de formigas, o algoritmo publicado por Häckel et al. (2008) foi implementado de acordo com a descrição original, e neste trabalho será referenciado como HACO (ver Seção 2.2). O único parâmetro não definido pelos autores foi a penalização ξ . Diferentes valores foram testados ($\xi \in \{0, 2; 0, 4; 0, 5; 0, 6; 0, 8\}$), e a configuração $\xi = 0, 8$ é utilizada. A configuração de parâmetros escolhida para os três algoritmos foi: (i) HACO: $\chi_{colonies} = 3$, $\chi_{ants} = 4$, $\alpha = 0, 5$, $\beta = 3$, $\tau_0 = 0, 05$, $\rho = 0, 1$, $q_0 = 0, 5$, e $\xi = 0, 8$; (ii) NSGA-II: $pop=2500/3000$, $a = 25$, $b = 800$, $cross = 0, 9$, $mut = 0, 005$, e; (iii) GRACE: $n_{ants} = 300$, $\alpha = 0, 6$, $\beta = 0, 6$, $\tau_0 = 1$, $\tau_{deposit} = 10$ e $R = 5$.

Na etapa inicial da experimentação realizada, os resultados do teste de Kruskal-Wallis sobre os rankings de dominância gerados mostram diferença significativa para todas as instâncias (Tabela 5). Comparações dois-a-dois com os testes de Wilcoxon foram realizados em todas as instâncias. Em seguida, o número de sucessos de cada otimizador foi contado e submetido ao teste para comparação de proporções disponibilizado por (TAILLARD; WAELTI; ZUBER, 2009). Um sucesso é computado quando um otimizador apresenta amostras significativamente melhores que outro considerando somente os rankings de dominância. Para um nível de significância 0,05, uma entrada inferior a este valor na linha i e coluna j significa que o otimizador representado pela linha i é significativamente melhor que o otimizador representado pela coluna j , para o tipo de instância em questão.

Os resultados dessa comparação podem ser vistos nas Tabelas 6. Para o conjunto de instâncias *complete* percebe-se que, tanto quando somente os rankings de dominân-

Tabela 5: p -valores do teste de Kruskal-Wallis usando rankings de dominância de todos os otimizadores.

| # | p -valor | # | p -valor |
|---|-----------------|----|-----------------|
| 1 | 1.27e-14 | 10 | 8.96e-16 |
| 2 | 8.78e-13 | 11 | 5.87e-12 |
| 3 | 2.80e-15 | 12 | 4.50e-15 |
| 4 | 1.29e-10 | 13 | 1.35e-14 |
| 5 | 1.90e-09 | 14 | 1.81e-15 |
| 6 | 2.14e-13 | 15 | 4.86e-16 |
| 7 | 1.66e-10 | 16 | 2.24e-11 |
| 8 | 0.00508 | 17 | 1.22e-14 |
| 9 | 3.09e-13 | 18 | 3.97e-16 |

cia são levados em consideração (coluna DR), como quando os rankings e os indicadores de qualidade são analisados conjuntamente (coluna DR+QI), a implementação feita do HACO gera conjuntos de aproximação melhores que a adaptação do NSGA-II, e que o algoritmo GRACE obtém rankings melhores que ambos. Para as instâncias *grid*, o NSGA-II apresenta desempenho melhor que o HACO. No entanto, mais uma vez os conjuntos de aproximação gerados pelo GRACE são considerados estatisticamente melhores que os produzidos pelos outros otimizadores. Conforme Knowles, Thiele e Zitzler (2009), pode-se afirmar que o algoritmo GRACE gera conjuntos de aproximação melhores que as implementações do NSGA-II e do HACO para ambos os conjuntos de instâncias, e que entre os dois últimos o desempenho depende do tipo de instância utilizado.

Tabela 6: p -valores da comparação de proporções: GRACE, NSGA-II (DEB et al., 2002) e HACO (HÄCKEL et al., 2008).

| | DR | | | DR + QI | | |
|----------|-------|----------|-------------|---------|----------|----------|
| Complete | GRACE | NSGA-II | HACO | GRACE | NSGA-II | HACO |
| GRACE | - | 0 | 0.01 | - | 0 | 0 |
| NSGA-II | 1 | - | 1 | 1 | - | 1 |
| HACO | 1 | 0 | - | 1 | 0 | - |
| Grid | GRACE | NSGA-II | HACO | GRACE | NSGA-II | HACO |
| GRACE | - | 0 | 0 | - | 0 | 0 |
| NSGA-II | 1 | - | 0 | 1 | - | 0 |
| HACO | 1 | 1 | - | 1 | 1 | - |

Na análise em relação aos conjuntos Pareto-ótimos, todos os indicadores são favoráveis ao algoritmo GRACE, para ambos os conjuntos de instâncias. O algoritmo HACO apresenta melhor desempenho que o NSGA-II para as instâncias 1–9, porém o oposto ocorre nas instâncias 10–18. As Tabelas 7–10 mostra as medianas, mínimos, máximos, desvios-padrão e coeficientes de variação para cada indicador/algoritmo/instância.

Tabela 7: Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias completas.

| # | $I_{\epsilon+}$ | | | I_H | | | <i>gen. dist.</i> | | | <i>error ratio</i> | | | \mathcal{C}_{\prec} | | |
|---|-----------------|--------|--------|-------|-------|------|-------------------|--------|--------|--------------------|-------|------|-----------------------|-------|------|
| | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO |
| 1 | 68.00 | 248.00 | 248.00 | 0.00 | 0.04 | 0.03 | 20.30 | 30.75 | 29.21 | 0.03 | 0.15 | 0.17 | 0.03 | 0.15 | 0.17 |
| | 0.00 | 131.00 | 248.00 | 0.00 | 0.03 | 0.02 | 0.00 | 0.00 | 25.35 | 0.00 | 0.00 | 0.13 | 0.00 | 0.00 | 0.13 |
| | 192.00 | 412.00 | 290.00 | 0.00 | 0.06 | 0.03 | 31.49 | 80.53 | 39.02 | 0.10 | 0.35 | 0.25 | 0.10 | 0.35 | 0.25 |
| | 60.43 | 77.60 | 19.25 | 0.00 | 0.01 | 0.00 | 14.34 | 20.89 | 3.61 | 0.03 | 0.10 | 0.03 | 0.03 | 0.10 | 0.03 |
| | 0.97 | 0.30 | 0.07 | 0.98 | 0.21 | 0.10 | 0.98 | 0.69 | 0.12 | 1.11 | 0.68 | 0.17 | 1.11 | 0.68 | 0.17 |
| 2 | 120.00 | 339.00 | 120.00 | 0.00 | 0.09 | 0.01 | 5.92 | 54.02 | 5.37 | 0.03 | 0.42 | 0.04 | 0.03 | 0.42 | 0.04 |
| | 89.00 | 314.00 | 120.00 | 0.00 | 0.05 | 0.01 | 0.00 | 23.88 | 0.00 | 0.00 | 0.14 | 0.00 | 0.00 | 0.14 | 0.00 |
| | 145.00 | 430.00 | 194.00 | 0.01 | 0.13 | 0.02 | 12.45 | 91.49 | 27.66 | 0.10 | 0.59 | 0.19 | 0.10 | 0.59 | 0.19 |
| | 13.64 | 27.93 | 16.39 | 0.00 | 0.02 | 0.00 | 4.05 | 13.20 | 7.21 | 0.03 | 0.11 | 0.05 | 0.03 | 0.11 | 0.05 |
| | 0.11 | 0.08 | 0.13 | 0.29 | 0.18 | 0.30 | 0.86 | 0.25 | 1.15 | 0.87 | 0.28 | 1.21 | 0.87 | 0.28 | 1.21 |
| 3 | 206.00 | 436.00 | 324.00 | 0.01 | 0.14 | 0.03 | 16.86 | 81.33 | 24.03 | 0.20 | 0.64 | 0.33 | 0.20 | 0.64 | 0.33 |
| | 126.00 | 363.00 | 294.00 | 0.01 | 0.08 | 0.02 | 7.29 | 45.08 | 15.65 | 0.08 | 0.33 | 0.08 | 0.08 | 0.33 | 0.08 |
| | 304.00 | 555.00 | 363.00 | 0.02 | 0.18 | 0.04 | 22.29 | 123.28 | 36.52 | 0.30 | 0.86 | 0.50 | 0.30 | 0.86 | 0.50 |
| | 58.93 | 50.15 | 25.51 | 0.00 | 0.03 | 0.00 | 3.70 | 16.34 | 4.78 | 0.06 | 0.16 | 0.09 | 0.06 | 0.16 | 0.09 |
| | 0.27 | 0.11 | 0.08 | 0.17 | 0.23 | 0.15 | 0.22 | 0.21 | 0.19 | 0.34 | 0.25 | 0.28 | 0.34 | 0.25 | 0.28 |
| 4 | 103.00 | 245.00 | 230.00 | 0.00 | 0.03 | 0.02 | 0.00 | 38.54 | 24.02 | 0.00 | 0.20 | 0.17 | 0.00 | 0.20 | 0.17 |
| | 50.00 | 147.00 | 132.00 | 0.00 | 0.02 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 132.00 | 305.00 | 234.00 | 0.01 | 0.07 | 0.03 | 14.61 | 48.17 | 27.02 | 0.05 | 0.31 | 0.19 | 0.05 | 0.31 | 0.19 |
| | 15.24 | 45.40 | 36.19 | 0.00 | 0.01 | 0.00 | 4.70 | 14.88 | 6.97 | 0.02 | 0.09 | 0.05 | 0.02 | 0.09 | 0.05 |
| | 0.15 | 0.21 | 0.17 | 0.26 | 0.33 | 0.17 | 2.77 | 0.48 | 0.32 | 2.77 | 0.53 | 0.32 | 2.77 | 0.53 | 0.32 |
| 5 | 72.00 | 517.00 | 196.00 | 0.01 | 0.16 | 0.04 | 0.00 | 126.19 | 71.29 | 0.00 | 0.92 | 0.29 | 0.00 | 0.92 | 0.29 |
| | 62.00 | 196.00 | 121.00 | 0.01 | 0.06 | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 72.00 | 627.00 | 196.00 | 0.01 | 0.23 | 0.06 | 51.86 | 219.17 | 235.31 | 0.25 | 1.00 | 0.43 | 0.25 | 1.00 | 0.43 |
| | 3.31 | 139.02 | 23.37 | 0.00 | 0.05 | 0.01 | 18.71 | 37.10 | 50.05 | 0.07 | 0.27 | 0.13 | 0.07 | 0.27 | 0.13 |
| | 0.05 | 0.31 | 0.13 | 0.07 | 0.35 | 0.24 | 1.42 | 0.29 | 0.65 | 1.57 | 0.34 | 0.51 | 1.57 | 0.34 | 0.51 |

Tabela 8: Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias completas (continuação).

| # | $I_{\epsilon+}$ | | | I_H | | | <i>gen. dist.</i> | | | <i>error ratio</i> | | | \mathcal{C}_{\prec} | | |
|---|-----------------|--------|--------|-------|-------|------|-------------------|--------|-------|--------------------|-------|------|-----------------------|-------|------|
| | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO |
| 6 | 135.00 | 550.00 | 219.00 | 0.01 | 0.21 | 0.07 | 8.89 | 103.79 | 36.22 | 0.06 | 0.89 | 0.55 | 0.06 | 0.89 | 0.55 |
| | 123.00 | 426.00 | 210.00 | 0.01 | 0.14 | 0.04 | 0.00 | 68.03 | 27.77 | 0.00 | 0.57 | 0.29 | 0.00 | 0.57 | 0.29 |
| | 182.00 | 666.00 | 377.00 | 0.02 | 0.34 | 0.09 | 20.43 | 204.47 | 45.43 | 0.14 | 1.00 | 0.71 | 0.14 | 1.00 | 0.71 |
| | 18.99 | 44.19 | 53.10 | 0.00 | 0.05 | 0.01 | 6.11 | 31.06 | 4.53 | 0.05 | 0.12 | 0.09 | 0.05 | 0.12 | 0.09 |
| | 0.14 | 0.08 | 0.21 | 0.16 | 0.21 | 0.14 | 0.66 | 0.28 | 0.12 | 0.68 | 0.13 | 0.16 | 0.68 | 0.13 | 0.16 |
| 7 | 87.00 | 386.00 | 157.00 | 0.00 | 0.12 | 0.03 | 0.00 | 51.48 | 17.71 | 0.00 | 0.25 | 0.08 | 0.00 | 0.25 | 0.08 |
| | 87.00 | 157.00 | 157.00 | 0.00 | 0.05 | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 87.00 | 535.00 | 157.00 | 0.01 | 0.23 | 0.03 | 15.43 | 112.89 | 37.90 | 0.06 | 0.71 | 0.24 | 0.06 | 0.71 | 0.24 |
| | 0.00 | 104.51 | 0.00 | 0.00 | 0.05 | 0.00 | 6.70 | 41.34 | 11.81 | 0.02 | 0.23 | 0.06 | 0.02 | 0.23 | 0.06 |
| | 0.00 | 0.28 | 0.00 | 0.45 | 0.41 | 0.16 | 1.64 | 0.73 | 0.55 | 1.64 | 0.79 | 0.58 | 1.64 | 0.79 | 0.58 |
| 8 | 283.00 | 440.00 | 331.00 | 0.02 | 0.09 | 0.05 | 18.65 | 85.87 | 26.06 | 0.43 | 1.00 | 0.53 | 0.43 | 0.40 | 0.53 |
| | 170.00 | 244.00 | 276.00 | 0.01 | 0.05 | 0.03 | 14.21 | 53.33 | 18.80 | 0.22 | 1.00 | 0.34 | 0.22 | 0.12 | 0.34 |
| | 354.00 | 608.00 | 358.00 | 0.02 | 0.21 | 0.06 | 24.96 | 141.03 | 33.01 | 0.55 | 1.00 | 0.65 | 0.55 | 0.88 | 0.65 |
| | 60.87 | 94.74 | 22.73 | 0.00 | 0.04 | 0.01 | 2.18 | 23.10 | 3.53 | 0.07 | 0.00 | 0.08 | 0.07 | 0.17 | 0.08 |
| | 0.24 | 0.20 | 0.07 | 0.22 | 0.38 | 0.16 | 0.12 | 0.25 | 0.13 | 0.18 | 0.00 | 0.15 | 0.18 | 0.41 | 0.15 |
| 9 | 158.00 | 541.00 | 312.00 | 0.01 | 0.20 | 0.06 | 9.42 | 106.27 | 30.68 | 0.13 | 0.92 | 0.60 | 0.13 | 0.92 | 0.60 |
| | 137.00 | 426.00 | 232.00 | 0.00 | 0.13 | 0.03 | 3.52 | 76.61 | 22.27 | 0.06 | 0.62 | 0.43 | 0.06 | 0.62 | 0.43 |
| | 167.00 | 671.00 | 364.00 | 0.01 | 0.32 | 0.09 | 12.16 | 221.65 | 39.09 | 0.20 | 1.00 | 0.71 | 0.20 | 1.00 | 0.71 |
| | 7.60 | 64.26 | 25.02 | 0.00 | 0.04 | 0.01 | 1.97 | 33.55 | 4.04 | 0.04 | 0.12 | 0.08 | 0.04 | 0.12 | 0.08 |
| | 0.05 | 0.12 | 0.08 | 0.13 | 0.20 | 0.20 | 0.22 | 0.30 | 0.13 | 0.33 | 0.13 | 0.13 | 0.33 | 0.13 | 0.13 |

Tabela 9: Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias *grid*.

| # | $I_{\epsilon+}$ | | | I_H | | | <i>gen. dist.</i> | | | <i>error ratio</i> | | | \mathcal{C}_{\prec} | | |
|----|-----------------|---------|----------|-------|--------|------|-------------------|--------|---------|--------------------|--------|------|-----------------------|--------|------|
| | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO |
| 10 | 0.00 | 141.00 | 835.00 | 0.00 | 0.00 | 0.05 | 0.00 | 0.00 | 80.42 | 0.00 | 0.00 | 0.68 | 0.00 | 0.00 | 0.68 |
| | 0.00 | 38.00 | 791.00 | 0.00 | 0.00 | 0.04 | 0.00 | 0.00 | 65.42 | 0.00 | 0.00 | 0.62 | 0.00 | 0.00 | 0.62 |
| | 0.00 | 550.00 | 1051.00 | 0.00 | 0.01 | 0.06 | 0.00 | 17.04 | 91.10 | 0.00 | 0.10 | 0.75 | 0.00 | 0.10 | 0.75 |
| | 0.00 | 133.47 | 91.24 | 0.00 | 0.00 | 0.00 | 0.00 | 5.22 | 7.15 | 0.00 | 0.03 | 0.03 | 0.00 | 0.03 | 0.03 |
| | nan | 0.72 | 0.10 | nan | 2.00 | 0.08 | nan | 1.59 | 0.09 | nan | 1.94 | 0.04 | nan | 1.94 | 0.04 |
| 11 | 68.00 | 995.00 | 1243.00 | 0.00 | 0.01 | 0.05 | 0.00 | 19.40 | 99.99 | 0.00 | 0.42 | 0.88 | 0.00 | 0.42 | 0.88 |
| | 58.00 | 463.00 | 1166.00 | 0.00 | 0.00 | 0.04 | 0.00 | 8.73 | 92.03 | 0.00 | 0.13 | 0.81 | 0.00 | 0.13 | 0.81 |
| | 222.00 | 1572.00 | 1661.00 | 0.00 | 0.02 | 0.05 | 2.39 | 32.26 | 116.09 | 0.01 | 0.76 | 0.93 | 0.01 | 0.76 | 0.93 |
| | 42.30 | 309.80 | 106.20 | 0.00 | 0.01 | 0.00 | 0.53 | 5.97 | 6.85 | 0.00 | 0.17 | 0.03 | 0.00 | 0.17 | 0.03 |
| | 0.49 | 0.33 | 0.08 | 1.62 | 0.44 | 0.06 | 2.26 | 0.30 | 0.07 | 1.91 | 0.39 | 0.04 | 1.91 | 0.39 | 0.04 |
| 12 | 281.00 | 795.00 | 3971.00 | 0.00 | 0.01 | 0.15 | 2.75 | 16.73 | 767.65 | 0.01 | 0.45 | 1.00 | 0.01 | 0.45 | 1.00 |
| | 102.00 | 571.00 | 3456.00 | 0.00 | 0.00 | 0.14 | 0.56 | 11.20 | 669.08 | 0.00 | 0.22 | 1.00 | 0.00 | 0.22 | 1.00 |
| | 295.00 | 1298.00 | 4571.00 | 0.00 | 0.03 | 0.16 | 3.93 | 32.60 | 909.09 | 0.03 | 0.86 | 1.00 | 0.03 | 0.86 | 1.00 |
| | 58.08 | 180.40 | 276.28 | 0.00 | 0.01 | 0.00 | 0.85 | 7.19 | 54.62 | 0.01 | 0.20 | 0.00 | 0.01 | 0.20 | 0.00 |
| | 0.24 | 0.22 | 0.07 | 0.31 | 0.52 | 0.03 | 0.33 | 0.36 | 0.07 | 0.44 | 0.40 | 0.00 | 0.44 | 0.40 | 0.00 |
| 13 | 442.00 | 2341.00 | 7378.00 | 0.00 | 0.05 | 0.20 | 3.82 | 37.43 | 1629.72 | 0.13 | 0.99 | 1.00 | 0.13 | 0.99 | 1.00 |
| | 395.00 | 1179.00 | 7062.00 | 0.00 | 0.02 | 0.19 | 2.90 | 12.99 | 1241.14 | 0.11 | 0.58 | 1.00 | 0.11 | 0.58 | 1.00 |
| | 542.00 | 4146.00 | 8086.00 | 0.00 | 0.10 | 0.21 | 4.41 | 154.22 | 2193.62 | 0.16 | 1.00 | 1.00 | 0.16 | 1.00 | 1.00 |
| | 47.37 | 668.36 | 283.88 | 0.00 | 0.02 | 0.00 | 0.37 | 36.56 | 215.28 | 0.01 | 0.10 | 0.00 | 0.01 | 0.10 | 0.00 |
| | 0.10 | 0.28 | 0.04 | 0.09 | 0.36 | 0.02 | 0.10 | 0.68 | 0.13 | 0.11 | 0.11 | 0.00 | 0.11 | 0.11 | 0.00 |
| 14 | 1147.00 | 5731.00 | 14928.00 | 0.01 | 0.10 | 0.22 | 25.59 | 102.38 | 2919.84 | 0.49 | 1.00 | 1.00 | 0.49 | 1.00 | 1.00 |
| | 1057.00 | 3686.00 | 13204.00 | 0.01 | 0.06 | 0.21 | 20.95 | 41.35 | 2431.43 | 0.42 | 1.00 | 1.00 | 0.42 | 1.00 | 1.00 |
| | 1233.00 | 9349.00 | 15908.00 | 0.01 | 0.15 | 0.23 | 30.42 | 660.11 | 4250.58 | 0.55 | 1.00 | 1.00 | 0.55 | 1.00 | 1.00 |
| | 56.07 | 1114.47 | 700.81 | 0.00 | 0.02 | 0.00 | 2.14 | 122.73 | 427.42 | 0.03 | 0.00 | 0.00 | 0.03 | 0.00 | 0.00 |
| | 0.05 | 0.19 | 0.05 | 0.02 | 0.20 | 0.02 | 0.08 | 0.86 | 0.14 | 0.05 | 0.00 | 0.00 | 0.05 | 0.00 | 0.00 |

Tabela 10: Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias *grid* (continuação).

| # | $I_{\epsilon+}$ | | | I_H | | | <i>gen. dist.</i> | | | <i>error ratio</i> | | | \mathcal{C}_{\prec} | | |
|----|-----------------|---------|----------|-------|--------|------|-------------------|--------|---------|--------------------|--------|------|-----------------------|--------|------|
| | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO |
| 15 | 1175.00 | 6795.00 | 17278.00 | 0.01 | 0.11 | 0.25 | 22.05 | 141.42 | 4677.95 | 0.45 | 1.00 | 1.00 | 0.45 | 1.00 | 1.00 |
| | 1103.00 | 4574.00 | 16083.00 | 0.01 | 0.07 | 0.24 | 18.69 | 75.02 | 3679.02 | 0.42 | 1.00 | 1.00 | 0.42 | 1.00 | 1.00 |
| | 1346.00 | 9593.00 | 17965.00 | 0.01 | 0.15 | 0.25 | 23.94 | 340.55 | 7498.04 | 0.52 | 1.00 | 1.00 | 0.52 | 1.00 | 1.00 |
| | 72.39 | 1146.65 | 529.68 | 0.00 | 0.02 | 0.00 | 1.44 | 63.12 | 916.50 | 0.02 | 0.00 | 0.00 | 0.02 | 0.00 | 0.00 |
| | 0.06 | 0.17 | 0.03 | 0.01 | 0.15 | 0.01 | 0.07 | 0.41 | 0.19 | 0.05 | 0.00 | 0.00 | 0.05 | 0.00 | 0.00 |
| 16 | 0.00 | 490.00 | 1337.00 | 0.00 | 0.00 | 0.06 | 0.00 | 10.68 | 165.36 | 0.00 | 0.09 | 0.74 | 0.00 | 0.09 | 0.74 |
| | 0.00 | 263.00 | 1307.00 | 0.00 | 0.00 | 0.05 | 0.00 | 4.67 | 118.49 | 0.00 | 0.04 | 0.50 | 0.00 | 0.04 | 0.50 |
| | 12.00 | 863.00 | 1615.00 | 0.00 | 0.01 | 0.06 | 0.00 | 31.05 | 223.45 | 0.00 | 0.44 | 0.93 | 0.00 | 0.44 | 0.93 |
| | 6.08 | 130.14 | 103.99 | 0.00 | 0.00 | 0.00 | 0.00 | 6.46 | 25.32 | 0.00 | 0.10 | 0.10 | 0.00 | 0.10 | 0.10 |
| | 1.15 | 0.28 | 0.07 | 1.15 | 0.67 | 0.04 | nan | 0.55 | 0.16 | nan | 0.74 | 0.14 | nan | 0.74 | 0.14 |
| 17 | 486.00 | 2234.00 | 7175.00 | 0.00 | 0.05 | 0.15 | 3.50 | 48.54 | 1013.37 | 0.20 | 0.98 | 1.00 | 0.20 | 0.98 | 1.00 |
| | 408.00 | 1608.00 | 6489.00 | 0.00 | 0.03 | 0.15 | 2.94 | 24.95 | 828.94 | 0.17 | 0.81 | 1.00 | 0.17 | 0.81 | 1.00 |
| | 567.00 | 4122.00 | 7846.00 | 0.00 | 0.08 | 0.16 | 4.17 | 148.26 | 1207.47 | 0.24 | 1.00 | 1.00 | 0.24 | 1.00 | 1.00 |
| | 44.52 | 705.40 | 404.85 | 0.00 | 0.01 | 0.00 | 0.27 | 28.96 | 87.39 | 0.02 | 0.04 | 0.00 | 0.02 | 0.04 | 0.00 |
| | 0.09 | 0.28 | 0.06 | 0.06 | 0.21 | 0.02 | 0.08 | 0.50 | 0.09 | 0.09 | 0.04 | 0.00 | 0.09 | 0.04 | 0.00 |
| 18 | 1026.00 | 4653.00 | 10690.00 | 0.00 | 0.10 | 0.20 | 9.32 | 172.88 | 2564.73 | 0.31 | 1.00 | 1.00 | 0.31 | 1.00 | 1.00 |
| | 800.00 | 3712.00 | 10107.00 | 0.00 | 0.08 | 0.19 | 7.61 | 104.51 | 2092.44 | 0.26 | 1.00 | 1.00 | 0.26 | 1.00 | 1.00 |
| | 1175.00 | 7810.00 | 11454.00 | 0.00 | 0.14 | 0.21 | 11.09 | 483.98 | 3269.54 | 0.38 | 1.00 | 1.00 | 0.38 | 1.00 | 1.00 |
| | 93.70 | 1077.71 | 371.68 | 0.00 | 0.02 | 0.00 | 0.95 | 93.68 | 276.61 | 0.03 | 0.00 | 0.00 | 0.03 | 0.00 | 0.00 |
| | 0.09 | 0.21 | 0.03 | 0.04 | 0.16 | 0.01 | 0.10 | 0.49 | 0.11 | 0.09 | 0.00 | 0.00 | 0.09 | 0.00 | 0.00 |

5 Análise experimental: ampliando o algoritmo GRACE

No capítulo anterior, um algoritmo multiobjetivo em duas fases baseado na otimização por colônia de formigas foi proposto, e sua comparação com dois otimizadores encontrados na literatura revelou-se promissora. No entanto, mais que o desenvolvimento de um otimizador eficiente, esse trabalho tem por alvo identificar as potencialidades de diversas alternativas apresentadas na literatura dos MOACO, para compreender melhor a contribuição de cada uma no contexto do MSP. Neste capítulo, várias dessas alternativas são implementadas sobre o GRACE, gerando variantes que são comparadas utilizando a mesma metodologia apresentada na Sessão 4.2. Para o experimento deste capítulo, no entanto, todos os testes foram executados em um Intel Xeon QuadCore W3520 2.9GHz, com 8G de RAM, usando uma distribuição Scientific Linux 5.5 64bits.

As extensões adaptadas para o GRACE foram implementadas tanto com múltiplas como com apenas uma colônia. Quando o conceito por trás da versão não permite o uso de apenas uma colônia, isto é dito explicitamente. Nas abordagens multi-colônia, um único tipo de heurística e de feromônio foram usados para todas as colônias. Ademais, as colônias não interagem diretamente umas com as outras: cada uma possui sua própria matriz de feromônio e sua heurística. Contudo, para a seleção das formigas que atualizarão a matriz de feromônio, o teste de dominância é feito em relação a um arquivo global externo que contém soluções encontradas por todas as colônias ao longo da execução do algoritmo. Para o teste da estratégia de atualização de feromônio por região, surge uma questão importante: como ordenar as soluções para determinar a qual região a solução encontrada pertence. No contexto biobjetivo tal questão é solucionada através de uma simples ordenação lexicográfica, que divide o espaço objetivo de maneira igualitária. Para problemas com mais de dois objetivos, no entanto, não há solução trivial. A estratégia usada para contornar esse problema é apresentada posteriormente neste capítulo.

Quatro versões do algoritmo base foram criadas a fim de avaliar diferentes métodos

de criação de vetores de escalarização a serem utilizados para o cálculo da informação heurística. Essas versões foram nomeadas: *Global-Aleatório*, *Fixo-Único*, *Fixo-Múltiplos* e *Global-Pseudo-Aleatório*. Na versão *Global-Aleatório* (*GRan*, do inglês *Global-Random*), os vetores de escalarização são gerados aleatoriamente. Não há divisão do espaço objetivo, independentemente de quantas colônias sejam usadas. Dessa forma, só é possível o uso da estratégia de atualização das matrizes de feromônio por origem. Não há relação direta entre o número de vetores de escalarização e o número de colônias do algoritmo.

Na versão *Fixo-Único* (*FUn*), a segunda variante criada, os vetores de escalarização são gerados sistematicamente. Inicialmente, um parâmetro d é escolhido para determinar o número de divisões no intervalo $[0,1]$. Este número define os possíveis pesos que cada componente dos vetores de escalarização pode receber. Para $d = 2$, por exemplo, existem três valores possíveis: 0, 0,5 e 1. Em seguida, todos os possíveis vetores de escalarização são gerados, distribuídos uniformemente na região delimitada pelos pontos extremos do espaço objetivo. Para um problema com três objetivos, o número de combinações é dado por $((d+1)*(d+2))/2$, e esta região é delimitada pelos pontos extremos (1,0,0), (0,1,0) e (0,0,1). Assim, para $k = 3$ e $d = 2$, os vetores de escalarização gerados são [1; 0; 0], [0,5; 0,5; 0], [0,5; 0; 0,5], [0; 0,5; 0,5], [0; 1; 0] e [0; 0; 1]. Nesta versão algorítmica, o número de colônias é igual ao número de ponderações, uma vez que cada um desses vetores pertencerá a uma única colônia.

A versão *Fixo-Múltiplos* (*FMu*) consiste em uma variação da versão *FUn* onde os vetores de escalarização gerados são distribuídos igualmente entre as colônias. Assim, para m colônias e d divisões, $((d+1)*(d+2))/2m$ vetores de escalarização são distribuídos para cada colônia. Quando esta divisão não é exata, a colônia m pode receber vetores extra correspondentes ao resto da divisão inteira. Espera-se que esta variante tenha as mesmas características que a versão *FUn*, mas que seja computacionalmente mais eficiente.

A variante *Global-Pseudo-Aleatório* (*PGRan*, do inglês *Pseudo-Global-Random*) une as características das três versões anteriores. Para problemas com três objetivos a serem otimizados, os vetores de escalarização são gerados aleatoriamente dentro dos limites estabelecidos por um procedimento de triangularização. Dado um número d de divisões, o mesmo procedimento usado na versão *FUn* para geração dos vetores de escalarização é usado na região delimitada pelos pontos (1,0,0), (0,1,0) e (0,0,1), como mostra a Figura 4. Esses pontos são considerados vértices de regiões triangulares. Os vetores de escalarização de cada colônia são gerados aleatoriamente dentro de cada uma dessas regiões. Assim, nessa versão, o número de colônias é igual a d^2 . Uma vez que o esforço computacional

cresce rapidamente para valores crescente de d , valores pequenos são escolhidos.

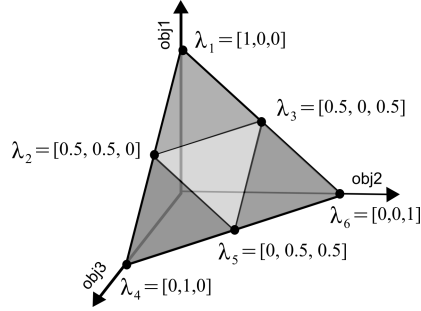


Figura 4: Exemplo de ponderações e colônias para a versão PGRan. Com $d = 2$, seis vetores são gerados, resultando em quatro colônias.

Para a avaliação dessas versões, os parâmetros de cada variante foram ajustados primeiramente. A versão GRan foi testada com 1, 3 e 5 colônias. A versão com apenas uma colônia corresponde ao algoritmo base GRACE. Cada formiga só pode atualizar a matriz de feromônio de sua própria colônia. Os p -valores obtidos com o teste de Kruskal-Wallis são mostrados na Tabela 11, onde a coluna $\#$ mostra a identificação da instância, e as colunas DR, $I_{\epsilon+}^1$ e I_H mostram, respectivamente, os p -valores correspondentes ao ranking de dominância, o indicador ϵ -unário e o hipervolume. Considerando um nível de significância 0,05 e apenas os ranking de dominância, os resultados apresentados na Tabela 11 mostram diferença significativa para 3 instâncias de cada classe. Para as instâncias em que o ranking de dominância não apontou diferenças, o I_H revela diferença significativa em uma instância *complete* e em 4 *grids*. Nestas quatro instâncias *grid*, o $I_{\epsilon+}^1$ também acusa diferença significativa.

Tabela 11: Resultados do teste de Kruskal-Wallis nos rankings de dominância e nos indicadores de qualidade gerados para a GRan usando 1, 3 e 5 colônias. NaN significa que as amostras comparadas são iguais.

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|----------------|-------------------|----------------|----|-----------------|-------------------|-----------------|
| 1 | 0.00195 | - | - | 10 | 0.02245 | - | - |
| 2 | 0.00013 | - | - | 11 | 1.38e-06 | - | - |
| 3 | 0.15987 | 0.21236 | 0.58046 | 12 | NaN | 0.00014 | 4.66e-15 |
| 4 | 0.35899 | 0.50174 | 0.16563 | 13 | NaN | 3.51e-05 | 1.49e-13 |
| 5 | 0.46921 | 0.30966 | 0.59756 | 14 | NaN | 0.54740 | 0.11592 |
| 6 | 0.16459 | 0.26382 | 0.90371 | 15 | NaN | 0.25459 | 0.50540 |
| 7 | 0.04576 | - | - | 16 | 4.75e-13 | - | - |
| 8 | 0.60301 | 0.00376 | 0.00821 | 17 | NaN | 1.55e-11 | 9.07e-17 |
| 9 | 0.84017 | 0.79358 | 0.24028 | 18 | NaN | 0.03452 | 0.00796 |

A comparação dois-a-dois foi efetuada considerando todas as instâncias, e seu resultado pode ser visto na Tabela 12. Quando somente os rankings de dominância são considerados

(coluna DR), o uso de apenas uma colônia apresenta melhor desempenho que o uso de três colônias para conjunto *complete*, e que o uso de cinco colônias para as instâncias *grid*. Os resultados em geral mostram que o uso de múltiplas colônias (da forma definida nesse trabalho) não melhora a performance do GRACE original. Os resultados também mostram que a versão com 3 colônias mostra-se melhor que a versão com 5 colônias para instâncias *grid*. Uma explicação possível para estes resultados é que, uma vez que essa versão multi-colônia equivale a uma estratégia de múltiplos reinícios, o uso de múltiplas colônias só aumenta o esforço computacional.

Tabela 12: p -valores da comparação de proporções para a versão GRan.

| | | DR | | | DR + QI | | |
|----------|----------|------|-------------|-------------|---------|-------------|-------------|
| | Colonies | 1 | 3 | 5 | 1 | 3 | 5 |
| Complete | 1 | - | 0.04 | 0.09 | - | 0.01 | 0.01 |
| | 3 | 0.96 | - | 0.75 | 0.99 | - | 0.75 |
| | 5 | 0.91 | 0.25 | - | 0.99 | 0.25 | - |
| Grid | 1 | - | 0.09 | 0.04 | - | 0 | 0 |
| | 3 | 0.91 | - | 0.25 | 1 | - | 0.01 |
| | 5 | 0.96 | 0.75 | - | 1 | 0.99 | - |

Para a versão FUN, os valores 1, 2 e 3 foram testados para o parâmetro d , que correspondem a 3, 6 e 10 vetores de escalarização e colônias. Novamente, apenas a atualização por origem é possível. Os resultados para a comparação dessas três versões contra a GRan com somente uma colônia são apresentados na Tabela 13. Esses resultados dizem respeito aos rankings de dominância e aos indicadores de qualidade. Considerando o ranking de dominância, percebe-se diferença significativa entre os otimizadores em quatro instâncias *complete* e em três instâncias *grid*. O indicador ϵ -unário aditivo detectou diferença significativa em mais uma instância *complete* e em três outras instâncias *grid*. O indicador de hipervolume aponta diferença significativa em cinco instâncias *grid*.

Tabela 13: p -valores do teste de Kruskal-Wallis: GRan com uma colônia contra FUN.

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|-----------------|-------------------|---------|----|-----------------|-------------------|-----------------|
| 1 | 3.85e-05 | - | - | 10 | 7.48e-06 | - | - |
| 2 | 0.00251 | - | - | 11 | 4.92e-09 | - | - |
| 3 | 0.00541 | - | - | 12 | NaN | 2.30e-08 | 6.27e-21 |
| 4 | 0.06157 | 0.59784 | 0.11722 | 13 | NaN | 0.00063 | 7.85e-16 |
| 5 | 0.07878 | 0.08633 | 0.01941 | 14 | NaN | 0.46923 | 0.06072 |
| 6 | 0.61764 | 0.76049 | 0.093 | 15 | NaN | 0.21608 | 0.00357 |
| 7 | 0.1909 | 0.28956 | 0.05886 | 16 | 6.94e-19 | - | - |
| 8 | 0.10916 | 0.00476 | 0.33655 | 17 | NaN | 7.19e-14 | 4.14e-09 |
| 9 | 0.00974 | - | - | 18 | NaN | 0.12736 | 0.00020 |

Os valores finais do teste de comparação de proporções efetuado na comparação dois-a-dois são mostrados na Tabela 14. Os resultados mostram que a versão GRan com apenas uma colônia apresenta desempenho melhor que todas as versões FUn para todas as instâncias testadas. Entre as diferentes configurações usadas para a FUn, nenhuma diferença estatisticamente significativa foi encontrada para as instâncias *complete*. Por outro lado, os resultados mostram que, quando menor o número de divisões, melhor a performance desta versão do algoritmo para o conjunto *grid*.

Tabela 14: p -valores para a comparação de proporções da GRan contra a FUn.

| | | DR | | | | DR + QI | | | |
|----------|----|------|-------------|-------------|-------------|---------|----------|----------|-------------|
| Versions | | G | 3 | 6 | 10 | G | 3 | 6 | 10 |
| Complete | G | - | 0.04 | 0.04 | 0.25 | - | 0 | 0 | 0 |
| | 3 | 0.96 | - | 0.5 | 0.5 | 1 | - | 0.25 | 0.25 |
| | 6 | 0.96 | 0.5 | - | 0.5 | 1 | 0.75 | - | 0.5 |
| | 10 | 0.75 | 0.5 | 0.5 | - | 0.91 | 0.75 | 0.5 | - |
| Grid | G | - | 0.04 | 0.04 | 0.04 | - | 0 | 0 | 0 |
| | 3 | 0.96 | - | 0.09 | 0.09 | 1 | - | 0 | 0 |
| | 6 | 0.96 | 0.91 | - | 0.25 | 1 | 1 | - | 0.04 |
| | 10 | 0.96 | 0.91 | 0.75 | - | 1 | 1 | 0.96 | - |

Para a versão FMu, testes foram realizados usando $m \in \{1, 3, 5\}$ e $d \in \{3, 6, 10\}$, que correspondem a uma total de 10, 28 e 66 vetores de escalarização, respectivamente. Inicialmente, as versões com $m = 1$ foram comparadas para os diferentes valores do parâmetro d . Nenhuma diferença significativa foi percebida quer entre os rankings de dominância quer entre os indicadores de qualidade. O mesmo foi observado para $m = 3$ e $m = 5$. Isso significa que o parâmetro m pode ser avaliado para qualquer valor de d , e os resultados esperados deve ser semelhantes. Desta forma, fixando $d = 6$, a Tabela 15 mostra os resultados do teste de Kruskal-Wallis para a compração dos diferentes valores de m .

Aparentemente, os mesmos conjuntos de aproximação foram gerados para as instâncias *complete*, independentemente da configuração usada. Para as instâncias *grid*, no entanto, diferenças significativas foram encontradas para três instâncias segundo os rankings de dominância, e para outras seis instâncias segundo os indicadores de qualidade. A Tabela 16 mostra os resultados do teste de Taillard, Waelti e Zuber (2009) aplicado na comparação dois-a-dois com o teste de Wilcoxon (1945). Para qualquer número de divisões, quanto menor a quantidade de colônias usadas, melhor a performance geral do algoritmo para as instâncias *grid*.

Por fim, para a PGRan, testes foram executados para $d \in \{1, 2, 3, 4, 5\}$, que corres-

Tabela 15: Análise do parâmetro m para a FMu com o teste de Kruskal-Wallis: $d = 6$, $m \in \{1, 3, 5\}$. NaN significa que as amostras comparadas são iguais.

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|-----|-------------------|---------|----|-----------------|-------------------|-----------------|
| 1 | NaN | NaN | NaN | 10 | 0.19424 | 0.03246 | 0.03246 |
| 2 | NaN | NaN | NaN | 11 | 0.01064 | - | - |
| 3 | NaN | NaN | NaN | 12 | NaN | 0.00023 | 9.61e-16 |
| 4 | NaN | NaN | NaN | 13 | NaN | 0.00386 | 1.96e-15 |
| 5 | NaN | NaN | NaN | 14 | NaN | 0.58568 | 0.03837 |
| 6 | NaN | NaN | NaN | 15 | NaN | 0.14769 | 0.50766 |
| 7 | NaN | NaN | NaN | 16 | 2.27e-09 | - | - |
| 8 | NaN | NaN | NaN | 17 | NaN | 4.15e-14 | 3.39e-16 |
| 9 | NaN | NaN | NaN | 18 | NaN | 0.28622 | 0.00019 |

Tabela 16: Resultados da comparação de proporções para as instâncias *grid*.

| | | DR | | | DR + QI | | |
|------|----------|------|------|------|---------|----------|-------------|
| | Colonies | 1 | 3 | 5 | 1 | 3 | 5 |
| Grid | 1 | - | 0.09 | 0.09 | - | 0 | 0 |
| | 3 | 0.91 | - | 0.25 | 1 | - | 0.01 |
| | 5 | 0.91 | 0.75 | - | 1 | 0.99 | - |

pondem a 1, 4, 9, 16 e 25 colônias, respectivamente. A Tabela 17 lista os resultados para a comparação entre todas as versões com o teste de Kruskal-Wallis. Os p -valores apresentados nessa tabela mostram que existe diferença significativa para a maior parte das instâncias. Os resultados da comparação dois-a-dois apresentados na Tabela 18 mostram que a versão com uma colônia, que se equivale à GRan com apenas uma colônia, apresenta um desempenho melhor que todas as outras versões.

Tabela 17: Resultados para a comparação da PGRan usando $d \in \{1, 2, 3, 4, 5\}$.

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|-----------------|-------------------|----------------|----|-----------------|-------------------|-----------------|
| 1 | 6.30e-06 | - | - | 10 | 1.17e-12 | - | - |
| 2 | 0.07911 | 0.00568 | 0.00031 | 11 | 1.34e-08 | - | - |
| 3 | 0.07548 | 0.00406 | 0.05932 | 12 | NaN | 4.48e-14 | 6.24e-24 |
| 4 | 0.02135 | - | - | 13 | NaN | 4.00e-06 | 1.38e-20 |
| 5 | 0.32284 | 0.41566 | 0.37870 | 14 | NaN | 0.88783 | 0.00022 |
| 6 | 0.20164 | 0.74161 | 0.58289 | 15 | NaN | 0.09336 | 0.00021 |
| 7 | 0.02269 | - | - | 16 | 4.02e-21 | - | - |
| 8 | 0.66612 | 0.04934 | 0.00142 | 17 | NaN | 2.35e-15 | 2.89e-21 |
| 9 | 0.09430 | 0.35075 | 0.00032 | 18 | NaN | 0.03802 | 6.12e-09 |

Duas comparações globais foram realizadas. Enquanto a primeira compara apenas versões com uma colônia, a segunda avalia as versões onde múltiplas colônias foram usadas. Para configurações com apenas uma colônia, as versões algorítmicas testadas foram:

Tabela 18: p -valores para a comparação de proporções da PGRan com $d \in \{1, 2, 3, 4, 5\}$.

| | DR | | | | | DR + QI | | | | |
|----------|------|------|-------------|-------------|-------------|---------|----------|----------|-------------|-------------|
| Complete | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
| 1 | - | 0.09 | 0.09 | 0.09 | 0.04 | - | 0 | 0 | 0.01 | 0 |
| 2 | 0.91 | - | 0.09 | 0.25 | 0.25 | 1 | - | 0.25 | 0.25 | 0.04 |
| 3 | 0.91 | 0.91 | - | 0.5 | 0.25 | 1 | 0.75 | - | 0.75 | 0.09 |
| 4 | 0.91 | 0.75 | 0.5 | - | 0.25 | 0.99 | 0.75 | 0.25 | - | 0 |
| 5 | 0.96 | 0.75 | 0.75 | 0.75 | - | 1 | 0.96 | 0.91 | 1 | - |
| Grid | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
| 1 | - | 0.09 | 0.04 | 0.04 | 0.04 | - | 0 | 0 | 0 | 0 |
| 2 | 0.91 | - | 0.09 | 0.09 | 0.09 | 1 | - | 0 | 0 | 0 |
| 3 | 0.96 | 0.91 | - | 0.09 | 0.09 | 1 | 1 | - | 0 | 0 |
| 4 | 0.96 | 0.91 | 0.91 | - | 0.5 | 1 | 1 | 1 | - | 0 |
| 5 | 0.96 | 0.91 | 0.91 | 0.5 | - | 1 | 1 | 1 | 1 | - |

GRan, FMu com $d = 3$ e PGRan com $d = 1$. Além destas, a FUn foi substituída pela FMu com $d = 1$, e chamada *Fix*. A Tabela 19 apresenta os resultados obtidos através do teste de Kruskal-Wallis. Os resultados mostram diferenças significativas para todas as instâncias completas e para duas instâncias *grid* segundo os rankings de dominância. Dentre o conjunto *grid*, o indicador de hipervolume indica diferenças significativas em seis instâncias e o indicador ϵ -unário aditivo mostra diferença em três instâncias a uma nível de significância de 0,05. Os resultados da comparação dois-a-dois exibidos na Tabela 20 mostram que, para diferentes tipos de instâncias, desempenhos diferentes são observados entre os algoritmos. Dentre as instâncias *complete*, a Tabela 20 mostra que as performances das abordagens GRan e PGRan são melhores que as outras versões. Para as instâncias *grid*, o contrário acontece. As abordagens sistemáticas FUn e FMu apresentam desempenhos melhores que as outras para praticamente todas as instâncias.

Tabela 19: Teste de Kruskal-Wallis para versões com uma colônia: GRan, Fix, FMu e PGRan.

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|-----------------|-------------------|---------|----|-----------------|-------------------|-----------------|
| 1 | 5.33e-23 | - | - | 10 | 0.28296 | 0.28296 | 0.28296 |
| 2 | 1.07e-23 | - | - | 11 | 0.00337 | - | - |
| 3 | 7.51e-25 | - | - | 12 | NaN | 0.66977 | 1.46e-11 |
| 4 | 4.17e-22 | - | - | 13 | NaN | 0.27017 | 7.01e-12 |
| 5 | 4.39e-22 | - | - | 14 | NaN | 0.04226 | 2.20e-07 |
| 6 | 1.62e-24 | - | - | 15 | NaN | 0.71302 | 0.00170 |
| 7 | 2.14e-22 | - | - | 16 | 2.33e-02 | - | - |
| 8 | 1.27e-25 | - | - | 17 | NaN | 6.77e-06 | 1.79e-18 |
| 9 | 3.23e-25 | - | - | 18 | NaN | 5.82e-05 | 6.00e-11 |

Tabela 20: p -valores para comparação de proporções entre versões com uma colônia: GRan, Fix, FMu e PGRan.

| | DR | | | | DR + QI | | | |
|----------|------|----------|----------|-------|----------|----------|----------|----------|
| Complete | GRan | Fix | FMu | PGRan | GRan | Fix | FMu | PGRan |
| GRan | - | 0 | 0 | 0.5 | - | 0 | 0 | 0.5 |
| Fix | 1 | - | 0.5 | 1 | 1 | - | 0.5 | 1 |
| FMu | 1 | 0.5 | - | 1 | 1 | 0.5 | - | 1 |
| PGRan | 0.5 | 0 | 0 | - | 0.5 | 0 | 0 | - |
| GRan | - | 0.91 | 0.75 | 0.5 | - | 1 | 1 | 0.5 |
| Fix | 0.09 | - | 0.5 | 0.09 | 0 | - | 0.5 | 0 |
| FMu | 0.25 | 0.5 | - | 0.25 | 0 | 0.5 | - | 0 |
| PGRan | 0.5 | 0.91 | 0.75 | - | 0.5 | 1 | 1 | - |

Na segunda comparação geral, todas as versões analisadas usam $m = 3$: GRan, FMu com $d = 3$ e Fix (FMu com $d = 1$). A PGRan com $d = 2$ (a configuração que mais se aproxima de $m = 3$) também foi incluída neste experimento. A Tabela 21 mostra que, como na comparação geral entre versões com uma colônia, os rankings de dominância indicam diferenças significativas para o conjunto de instâncias *complete*. Para as instâncias *grid*, o desempenho dos otimizadores pode ser considerado diferente para 8 das 9 instâncias. A Tabela 22 mostra que todos os otimizadores conseguem gerar conjuntos de aproximação melhores que a FMu para todas as instâncias *complete*, enquanto o oposto acontece para as instâncias *grid*: a versão FMu apresenta desempenho melhor que todas as outras.

Tabela 21: Comparação geral das versões com três colônias usando o teste de Kruskal-Wallis: GRan, Fix, FMu e PGRan.

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|-----------------|-------------------|---------|----|-----------------|-------------------|-----------------|
| 1 | 1.66e-15 | - | - | 10 | 0.08541 | 0.08154 | 0.08101 |
| 2 | 1.39e-15 | - | - | 11 | 0.10916 | 1.09e-08 | 7.89e-13 |
| 3 | 7.94e-20 | - | - | 12 | NaN | 0.03913 | 7.03e-14 |
| 4 | 8.24e-15 | - | - | 13 | NaN | 1.55e-05 | 9.75e-12 |
| 5 | 2.33e-13 | - | - | 14 | NaN | 0.09508 | 5.87e-07 |
| 6 | 8.58e-19 | - | - | 15 | NaN | 0.02027 | 0.02720 |
| 7 | 3.42e-15 | - | - | 16 | 1.38e-08 | - | - |
| 8 | 1.92e-24 | - | - | 17 | NaN | 6.44e-05 | 1.82e-15 |
| 9 | 4.16e-22 | - | - | 18 | NaN | 0.00074 | 5.74e-09 |

Resumindo os resultados deste grupo de versões, as abordagens que utilizam apenas uma colônia apresentam desempenho melhor nas instâncias testadas que as versões com múltiplas colônias. Dentre as variantes com apenas uma colônia, a GRan gera os melhores conjuntos de aproximação para as instâncias *complete*, mas apresenta o pior desempenho

Tabela 22: Comparação de proporções entre as versões GRan, Fix, FMu e PGRan ($m = 3$).

| | DR | | | | DR + QI | | | |
|----------|------|------|----------|-------|----------|-------------|----------|-------------|
| Complete | GRan | Fix | FMu | PGRan | GRan | Fix | FMu | PGRan |
| GRan | - | 0.25 | 0 | 0.5 | - | 0.04 | 0 | 0.5 |
| Fix | 0.75 | - | 0 | 0.5 | 0.96 | - | 0 | 0.91 |
| FMu | 1 | 1 | - | 1 | 1 | 1 | - | 1 |
| PGRan | 0.5 | 0.5 | 0 | - | 0.5 | 0.09 | 0 | - |
| Grid | GRan | Fix | FMu | PGRan | GRan | Fix | FMu | PGRan |
| GRan | - | 0.5 | 0.75 | 0.5 | - | 0.75 | 1 | 0.04 |
| Fix | 0.5 | - | 0.75 | 0.25 | 0.25 | - | 1 | 0.01 |
| FMu | 0.25 | 0.25 | - | 0.25 | 0 | 0 | - | 0 |
| PGRan | 0.5 | 0.75 | 0.75 | - | 0.96 | 0.99 | 1 | - |

nas instâncias *grid*. O oposto ocorre com a versão FMu: enquanto esta gera os melhores conjuntos para as instâncias *grid*, seu desempenho é superado por todas as outras versões no conjunto *complete*. Dentre as versões com múltiplas colônias, a GRan também gera conjuntos melhores que a FMu e que a FUn para as instâncias *complete*. Quando às *grid*, a FMu apresenta o melhor desempenho.

Uma quinta versão, denominada *Mixed* (*Mix*), foi desenvolvida como uma nova tentativa de agregação das características estocásticas e sistemáticas das versões anteriores. Essa variante assemelha-se a FMu, exceto pela inclusão de um vetor de escalarização aleatório por geração em cada colônia. Essa alteração permite às colônias realizar buscas em regiões diferentes do espaço objetivo, o que pode resultar em uma interferência positiva ou negativa no desempenho geral do algoritmo.

Além desta versão, um segundo grupo de versões algorítmicas foram desenvolvidas para permitir que diferentes estratégias de atualização de feromônio pudessem ser usadas por abordagens multi-colônia. Uma vez que não existe maneira trivial para ordenar as soluções no espaço objetivo de maneira igualitária, colônias que priorizam objetivos específicos foram usadas. Uma primeira versão proposta é denominada *Obj*. Os vetores de escalarização são gerados da mesma forma que na versão FMu, mas a cada colônia são atribuídas somente as ponderações que priorizam o critério específico no qual a colônia concentra seu esforços. Por exemplo, um vetor $[0,5; 0,5; 0]$ seria rejeitado pela colônia especializada no primeiro objetivo por não refletir a prioridade da mesma. O número de vetores de escalarização por colônia será menor que $(d+1)(d+2)/2$, onde d é o número de divisões (como na FMu). A quantidade de colônias é diretamente relacionada ao número de objetivos. Uma vez que deve haver pelo menos uma colônia por objetivo, uma ver-

são com apenas uma colônia não pode ser desenvolvida. Contudo, não há limite para a quantidade de colônias que podem focar um mesmo objetivo. Nos experimentos relatados aqui, apenas uma colônia por objetivo é usada. Cada colônia possui sua própria matriz de feromônio, que se especializa em seu objetivo primário.

Para a atualização do feromônio, três estratégias foram testadas: *atualização por origem*, *atualização por região*, e o uso de ambas as estratégias simultaneamente. Caso a atualização por região seja usada, é necessário determinar a qual região a solução gerada pertence. Dada uma nova solução s_{nova} e um arquivo externo A , a regra adotada diz que:

1. se uma solução $s \in A$ é dominada pela nova solução s_{nova} , então as matrizes de feromônio das colônias que priorizam os objetivos que foram otimizados são atualizadas;
2. se s_{nova} não domina nenhuma solução de A , ela é comparada com a última solução encontrada. Para cada objetivo. $i \in \{1, \dots, k\}$, se $z^i(s_{nova}) < z^i(s_{ltima})$, um depósito de feromônio será realizado na(s) matriz(es) da(s) colônia(s) que prioriza(m) tal objetivo.

Quando as atualizações por origem e por região são utilizadas simultaneamente, a estratégia aplicada para determinar a quais regiões a solução pertence é utilizada, mas a matriz da colônia a qual a formiga pertence sempre é atualizada. Uma segunda versão, denominada *ObjMixed* (*OMix*), é uma variante de *Obj*, criada a partir da inclusão em cada colônia de um vetor de escalarização aleatório a cada geração. Esta versão é uma tentativa de permitir que as colônias busquem regiões do espaço objetivo que normalmente não poderiam. Novamente, tenta-se agregar características estocásticas e sistemáticas em um mesmo algoritmo.

Para a avaliação das versões *Mix*, *Obj* e *OMix*, o ajuste de parâmetro foi inicialmente realizado, e em seguida uma comparação geral que inclui as melhores abordagens encontradas no primeiro grupo de versões foi efetuada. Para a versão *Mix*, os resultados da experimentação anterior foi usado para reduzir o número de possíveis valores a serem testados para o ajuste dos parâmetros m e d . Os valores testados para m são 1 e 3, e para d são 3, 6 e 10. O teste de Kruskal-Wallis não revelou diferença significativa entre as versões com diferentes valores para d , qualquer que seja o valor de m . No entanto, o mesmo efeito não foi observado na análise do parâmetro m . Diferenças significativas foram percebidas no teste de Wilcoxon entre as versões com uma ou três colônias dependendo do valor escolhido para o parâmetro d no conjunto de instâncias *complete*. Para $d = 6$, a

versão com apenas uma colônia produz conjuntos de aproximação melhores que a versão com $m = 3$, enquanto para $d = 1$ e $d = 3$ não é observada diferença significativa. Para as instâncias *grid*, todos os valores de d apresentam comportamento semelhante. Um exemplo desses resultados pode ser visto na Tabela 23 para $d = 6$. O teste de comparação de proporções confirmou que a versão com apenas uma colônia gera melhores conjuntos de aproximação que a versão com múltiplas colônias para ambos os conjuntos de instâncias testados (p -valor < 0.01).

Tabela 23: Resultados do teste bicaudal de Wilcoxon para a versão Mix: $m = 1$ e $m = 3$ ($d = 6$).

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|----------------|-------------------|----------------|----|-----------------|-------------------|-----------------|
| 1 | 0.69334 | 0.53947 | 0.37614 | 10 | 0.08599 | 0.06904 | 0.06904 |
| 2 | 0.03959 | - | - | 11 | 3.11e-05 | - | - |
| 3 | 0.16491 | 0.09387 | 0.04962 | 12 | NaN | 0.00020 | 7.40e-17 |
| 4 | 0.36820 | 0.96358 | 0.81861 | 13 | NaN | 0.00015 | 3.38e-17 |
| 5 | 0.27834 | 0.23760 | 0.02753 | 14 | NaN | 0.71329 | 0.18226 |
| 6 | 0.20222 | 0.44755 | 0.93568 | 15 | NaN | 0.65548 | 0.70819 |
| 7 | 0.37640 | 0.33371 | 0.45529 | 16 | 9.63e-11 | - | - |
| 8 | 0.16074 | 0.00279 | 0.41467 | 17 | NaN | 0.00010 | 1.69e-17 |
| 9 | 0.33371 | 0.73703 | 0.063228 | 18 | NaN | 0.15839 | 1.40e-05 |

Para o ajuste de parâmetros da versão Obj, testes foram realizados inicialmente para analisar o parâmetro d . Os valores testados foram 3, 4 e 5. Considerando qualquer das estratégias de atualização de feromônio utilizadas neste trabalho (atualização por origem, por região ou ambas simultaneamente), nenhuma diferença significativa foi apontada pelo teste de Kruskal-Wallis a um nível de significância 0,05.

Na segunda parte do experimento, as três estratégias de atualização foram comparadas diretamente para entender sua contribuição individual ao desempenho do algoritmo. A Tabela 24 mostra os p -valores para os rankings de dominância e os indicadores de qualidade quando $d = 3$ é usado. O teste de Kruskal-Wallis não indica diferença significativa para as instâncias do grupo *complete*, mas o contrário ocorre para sete instâncias do tipo *grid*. Os resultados obtidos com a comparação dois-a-dois são apresentados na Tabela 25, e mostram que a estratégia de atualização por origem apresenta pior desempenho em relação às outras para as instâncias *grid* testadas. Nenhuma diferença foi encontrada entre os desempenhos do algoritmo quando compara-se a atualização por região e o uso de ambas as estratégias simultaneamente. Este resultado indica que a eficiência do algoritmo está diretamente relacionada à estratégia de atualização por região.

Os mesmos valores usados pra o ajuste do parâmetro d na versão Obj foram usados

Tabela 24: Resultados do teste de Kruskal-Wallis para a versão Obj com diferentes métodos de atualização de feromônio: por origem, por região ou o uso de ambas simultaneamente.

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|---------|-------------------|---------|----|-----------------|-------------------|-----------------|
| 1 | 0.74524 | NaN | NaN | 10 | 0.03702 | - | - |
| 2 | 0.87877 | NaN | NaN | 11 | 0.00277 | - | - |
| 3 | 0.96529 | NaN | NaN | 12 | NaN | 0.61858 | 6.84e-10 |
| 4 | 0.60516 | NaN | NaN | 13 | NaN | 0.09566 | 1.28e-10 |
| 5 | 0.09474 | NaN | NaN | 14 | NaN | 0.95142 | 0.61685 |
| 6 | 0.14312 | NaN | NaN | 15 | NaN | 0.68661 | 0.45992 |
| 7 | 0.88791 | NaN | NaN | 16 | 1.87e-13 | - | - |
| 8 | 0.35550 | NaN | NaN | 17 | NaN | 2.80e-05 | 2.51e-13 |
| 9 | 0.14968 | NaN | NaN | 18 | NaN | 0.08899 | 0.00690 |

Tabela 25: Comparação de proporções da versão Obj com diferentes estratégias de atualização de feromônio.

| Grid | DR | | | DR + QI | | |
|--------|-------------|--------|-------|----------|--------|-------|
| | Origem | Região | Ambas | Origem | Região | Ambas |
| Origem | - | 0.96 | 0.91 | - | 1 | 1 |
| Região | 0.04 | - | 0.5 | 0 | - | 0.5 |
| Ambas | 0.09 | 0.5 | - | 0 | 0.5 | - |

para a versão OMix. Nenhuma diferença significativa foi encontrada para as instâncias *grid*, independentemente da estratégia de atualização de feromônio usada. Para o conjunto *complete*, no entanto, dois resultados diferentes foram observados. Para todos os métodos, $d = 3$ produziu conjuntos de aproximação melhores que quando $d = 5$ foi usado. Nenhuma diferença significativa foi percebida entre as versões com $d = 3$ e $d = 4$.

As diferentes estratégias de atualização de feromônio foram testadas para $d = 3$. A Tabela 26 mostra que, para a classe de instâncias *complete*, nenhuma diferença significativa foi percebida, exceto para as instâncias 6 e 8 onde os indicadores de qualidade hipervolume e ϵ -unário aditivo, respectivamente, detectaram diferença significativa a um nível de 0,05. Por outro lado, diferenças significativas foram encontradas para sete dentre nove instâncias *grid*. O teste para comparação de proporções (Tabela 27) mostrou que não há diferença entre as versões com as diferentes estratégias de atualização para o conjunto *complete*, mas para as instâncias *grid*, o mesmo resultado observado para a versão Obj se repete: a estratégia de atualização por região é crítica para a eficiência do algoritmo.

Para concluir este grupo de experimentos, três comparações gerais foram realizadas, contendo as versões: GRan, FMu, Mix, Obj e OMix. A primeira considera as versões

Tabela 26: Resultados do teste de Kruskal-Wallis para a versão OMix com $d = 3$: atualização por origem, atualização por região e o uso de ambas simultaneamente.

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|---------|-------------------|----------------|----|-----------------|-------------------|-----------------|
| 1 | 0.74524 | 0.26767 | 0.51013 | 10 | 0.00293 | - | - |
| 2 | 0.47284 | 0.79410 | 0.21648 | 11 | 0.07043 | 0.00017 | 4.22e-06 |
| 3 | 0.96529 | 0.56443 | 0.44202 | 12 | NaN | 0.15397 | 6.33e-12 |
| 4 | 0.60516 | 0.48452 | 0.84530 | 13 | NaN | 0.00848 | 1.00e-08 |
| 5 | 0.09474 | 0.35550 | 0.09422 | 14 | NaN | 0.90868 | 0.97435 |
| 6 | 0.14312 | 0.06349 | 0.03547 | 15 | NaN | 0.85855 | 0.49478 |
| 7 | 0.88791 | 0.34751 | 0.07134 | 16 | 7.40e-12 | - | - |
| 8 | 0.35550 | 0.04803 | 0.66676 | 17 | NaN | 7.87e-10 | 4.75e-16 |
| 9 | 0.14968 | 0.75931 | 0.63876 | 18 | NaN | 0.00592 | 0.00157 |

Tabela 27: p -valores para o teste de comparação de proporções da versão OMix com as diferentes estratégias de atualização de feromônio.

| | DR | | | DR + QI | | |
|----------|--------|--------|-------|----------|--------|-------|
| Complete | Origem | Região | Ambas | Origem | Região | Ambas |
| Origem | - | 0.5 | 0.5 | - | 0.25 | 0.5 |
| Região | 0.5 | - | 0.5 | 0.75 | - | 0.25 |
| Ambas | 0.5 | 0.5 | - | 0.5 | 0.75 | - |
| Grid | Origem | Região | Ambas | Origem | Região | Ambas |
| Origem | - | 0.91 | 0.91 | - | 1 | 1 |
| Região | 0.09 | - | 0.5 | 0 | - | 0.75 |
| Ambas | 0.09 | 0.5 | - | 0 | 0.25 | - |

com os parâmetros m e d iguais a 3, e as versões Obj e OMix com a atualização por origem. A segunda também utiliza $m = d = 3$, mas as versões Obj e OMix usam ambas as estratégias de atualização de feromônio simultaneamente. A motivação por trás dessa separação é determinar o impacto da estratégia de atualização por região sobre essas duas versões quando somente abordagens com múltiplas colônias são utilizadas. Por fim, na terceira comparação, $m = 1$ e $d = 3$ são usados. Novamente, ambas as estratégias de atualização de feromônio são usadas simultaneamente para as versões Obj e OMix.

As Tabelas 28 e 29 mostram os resultados quando m e d são iguais a 3, e somente a atualização por origem é usada pelas versões Obj e OMix. A Tabela 28 mostra que diferenças significativas podem ser vistas em 12 das 18 instâncias segundo os rankings de dominância. Diferenças significativas também são percebidas para as demais instâncias por, pelo menos, um dos indicadores de qualidade. Os resultados da comparação dois-a-dois apresentados na Tabela 29 mostram que a versão Mix se comporta exatamente como a GRan, significando que ela perde as características da FMu quando um vetor de escalarização aleatório é incluído. Um desempenho similar é observado entre as versões

Obj e FMu em ambos os conjuntos de instâncias. A mesma tabela também mostra que as versões GRan e Mix são ao mesmo tempo as melhores e as piores para as classes *complete* e *grid*, respectivamente. Já a versão OMix apresenta um desempenho melhor que as versões FMu e Obj para o conjunto *complete*, mas não tão bom quando as versões GRan e Mix.

Tabela 28: Resultados do teste de Kruskal-Wallis ($m = d = 3$): GRan, FMu, Mix, Obj e OMix (as duas últimas versões usando a atualização por origem).

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|-----------------|-------------------|---------|----|-----------------|-------------------|-----------------|
| 1 | 5.73e-29 | - | - | 10 | 0.04356 | - | - |
| 2 | 9.76e-29 | - | - | 11 | 0.00261 | - | - |
| 3 | 1.50e-29 | - | - | 12 | NaN | 0.01786 | 7.37e-17 |
| 4 | 1.70e-27 | - | - | 13 | NaN | 3.67e-05 | 1.21e-16 |
| 5 | 1.53e-26 | - | - | 14 | NaN | 0.94244 | 6.50e-06 |
| 6 | 1.38e-27 | - | - | 15 | NaN | 0.28765 | 4.53e-05 |
| 7 | 1.14e-26 | - | - | 16 | 5.06e-07 | - | - |
| 8 | 4.29e-30 | - | - | 17 | NaN | 0.11306 | 4.77e-18 |
| 9 | 1.60e-29 | - | - | 18 | NaN | 0.00019 | 6.30e-11 |

Tabela 29: Comparação de proporções com $m = d = 3$: GRan, FMu, Mix, Obj e OMix (as duas últimas com atualização por origem).

| | DR | | | | | DR + QI | | | | |
|----------|------|----------|------|----------|----------|----------|----------|----------|----------|-------------|
| Complete | GRan | FMu | Mix | Obj | OMix | GRan | FMu | Mix | Obj | OMix |
| GRan | - | 0 | 0.25 | 0 | 0 | - | 0 | 0.25 | 0 | 0 |
| FMu | 1 | - | 1 | 0.5 | 1 | 1 | - | 1 | 0.5 | 1 |
| Mix | 0.75 | 0 | - | 0 | 0 | 0.75 | 0 | - | 0 | 0 |
| Obj | 1 | 0.5 | 1 | - | 1 | 1 | 0.5 | 1 | - | 1 |
| OMix | 1 | 0 | 1 | 0 | - | 1 | 0 | 1 | 0 | - |
| Grid | GRan | FMu | Mix | Obj | OMix | GRan | FMu | Mix | Obj | OMix |
| GRan | - | 0.91 | 0.5 | 0.91 | 0.75 | - | 1 | 0.5 | 1 | 1 |
| FMu | 0.09 | - | 0.09 | 0.5 | 0.5 | 0 | - | 0 | 0.5 | 0.04 |
| Mix | 0.5 | 0.91 | - | 0.91 | 0.75 | 0.5 | 1 | - | 1 | 1 |
| Obj | 0.09 | 0.5 | 0.09 | - | 0.25 | 0 | 0.5 | 0 | - | 0.25 |
| OMix | 0.25 | 0.5 | 0.25 | 0.75 | - | 0 | 0.96 | 0 | 0.75 | - |

Quando a atualização por região é combinada com a atualização por origem, novamente há diferença significativa entre os conjuntos de aproximação gerados em todas as instâncias. A Tabela 30 mostra que os desempenhos das versões Obj e OMix são melhorados significativamente nas instâncias *grid*, e ambas conseguem superar a versão FMu. Nenhuma mudança é observada para as instâncias *complete* no que concerne a essas duas versões.

Por fim, a Tabela 31 mostra os resultados da terceira comparação geral. Diferenças

Tabela 30: Comparação de proporções com $m = d = 3$: GRan, FMu, Mix, Obj e OMix (as duas últimas com atualização por origem e por região simultaneamente).

| | DR | | | | | DR + QI | | | | |
|----------|-------------|-------------|-------------|----------|----------|----------|----------|----------|----------|----------|
| Complete | GRan | FMu | Mix | Obj | OMix | GRan | FMu | Mix | Obj | OMix |
| GRan | - | 0 | 0.25 | 0 | 0 | - | 0 | 0.25 | 0 | 0 |
| FMu | 1 | - | 1 | 0.5 | 1 | 1 | - | 1 | 0.5 | 1 |
| Mix | 0.75 | 0 | - | 0 | 0 | 0.75 | 0 | - | 0 | 0 |
| Obj | 1 | 0.5 | 1 | - | 1 | 1 | 0.5 | 1 | - | 1 |
| OMix | 1 | 0 | 1 | 0 | - | 1 | 0 | 1 | 0 | - |
| Grid | GRan | FMu | Mix | Obj | OMix | GRan | FMu | Mix | Obj | OMix |
| GRan | - | 0.91 | 0.5 | 0.96 | 0.96 | - | 1 | 0.5 | 1 | 1 |
| FMu | 0.09 | - | 0.09 | 0.91 | 0.96 | 0 | - | 0 | 1 | 1 |
| Mix | 0.5 | 0.91 | - | 0.96 | 0.96 | 0.5 | 1 | - | 1 | 1 |
| Obj | 0.04 | 0.09 | 0.04 | - | 0.5 | 0 | 0 | 0 | - | 1 |
| OMix | 0.04 | 0.04 | 0.04 | 0.5 | - | 0 | 0 | 0 | 0.09 | - |

significativas são encontradas para 16 das 18 instâncias pelo teste de Kruskal-Wallis com um nível de significância 0,05. A Tabela 32 apresenta os resultados dos testes dois-a-dois, onde os mesmos resultados observados anteriormente para o conjunto de instâncias *complete* se mantém. As versões Obj e OMix, contudo, não conseguem gerar conjuntos de aproximação tão bons quanto a versão FMu para as instâncias *grid*.

Tabela 31: p -valores dos teste de Kruskal-Wallis: comparando as versões GRan, FMu e Mix com apenas uma colônia, e as versões Obj e OMix com 3 colônias e atualização por origem de região simultaneamente.

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|-----------------|-------------------|---------|----|---------|-------------------|-----------------|
| 1 | 3.66e-29 | - | - | 10 | 0.07402 | 0.07402 | 0.07402 |
| 2 | 3.63e-29 | - | - | 11 | 0.07596 | 1.21e-06 | 1.54e-10 |
| 3 | 4.36e-29 | - | - | 12 | NaN | 0.57686 | 8.97e-10 |
| 4 | 3.17e-28 | - | - | 13 | NaN | 0.02665 | 1.15e-10 |
| 5 | 1.19e-24 | - | - | 14 | NaN | 0.19158 | 3.81e-07 |
| 6 | 3.95e-27 | - | - | 15 | NaN | 0.75345 | 0.01110 |
| 7 | 4.57e-28 | - | - | 16 | 0.23917 | 0.33997 | 0.11440 |
| 8 | 1.81e-30 | - | - | 17 | NaN | 1.04e-06 | 2.41e-18 |
| 9 | 8.47e-30 | - | - | 18 | NaN | 1.81e-05 | 3.09e-09 |

Pelos resultados apresentados nos experimentos realizados, percebe-se que, de forma geral, as versões com apenas uma colônia apresentam desempenho superior às abordagens multi-colônias testadas nesse trabalho para esse problema, considerando o conjunto de instâncias utilizado. Observa-se também que o uso de estocasticidade leva a bons resultados em um conjunto de instâncias com as características do *complete*, mas que para instâncias com uma estruturação semelhante às *grid* abordagens sistemáticas apresentam

Tabela 32: p -valores da comparação dois-a-dois: versões GRan, FMu e Mix com apenas uma colônia, e versões Obj e OMix com 3 colônias e atualização por origem e região simultaneamente.

| | DR | | | | | DR + QI | | | | |
|----------|------|----------|------|----------|----------|----------|----------|----------|-------------|----------|
| Complete | GRan | FMu | Mix | Obj | OMix | GRan | FMu | Mix | Obj | OMix |
| GRan | - | 0 | 0.09 | 0 | 0 | - | 0 | 0.09 | 0 | 0 |
| FMu | 1 | - | 1 | 0.5 | 1 | 1 | - | 1 | 0.5 | 1 |
| Mix | 0.91 | 0 | - | 0 | 0 | 0.91 | 0 | - | 0 | 0 |
| Obj | 1 | 0.5 | 1 | - | 1 | 1 | 0.5 | 1 | - | 1 |
| OMix | 1 | 0 | 1 | 0 | - | 1 | 0 | 1 | 0 | - |
| Grid | GRan | FMu | Mix | Obj | OMix | GRan | FMu | Mix | Obj | OMix |
| GRan | - | 0.5 | 0.5 | 0.5 | 0.5 | - | 1 | 0.5 | 1 | 1 |
| FMu | 0.5 | - | 0.5 | 0.5 | 0.5 | 0 | - | 0 | 0.04 | 0 |
| Mix | 0.5 | 0.5 | - | 0.5 | 0.5 | 0.6 | 1 | - | 1 | 1 |
| Obj | 0.5 | 0.5 | 0.5 | - | 0.5 | 0 | 0.96 | 0 | - | 0.09 |
| OMix | 0.5 | 0.5 | 0.5 | 0.5 | - | 0 | 1 | 0 | 0.91 | - |

um desempenho superior. Ademais, as tentativas de integração entre as diferentes versões não obtiveram sucesso uma vez que, invariavelmente, uma das características prevalece sobre as demais. No próximo capítulo, é proposta uma abordagem para a integração de múltiplas características em um único algoritmo.

6 Múltiplas espécies reguladas por disponibilidade de comida

Os resultados dos experimentos relatados no capítulo anterior mostram que diferentes configurações algorítmicas produzem os melhores conjuntos de aproximação para cada tipo de instâncias. Algumas tentativas de combinar as características estocásticas e sistemáticas das diferentes versões em um único algoritmo foram feitas, mas nenhuma versão conseguiu se sobressair sobre as demais em todas as classes de instâncias. Na tentativa de obter um *framework* que identifique potencialidades entre diferentes estratégias algorítmicas, este capítulo destina-se ao desenvolvimento de um algoritmo que possa adaptar-se dependendo das características da instância. Para conseguir isso, a idéia da regulação por disponibilidade de comida entre diferentes espécies de formigas é apresentada e comparada com as outras versões que apresentaram melhor desempenho na experimentação anterior. Em seguida, uma comparação final é realizada para avaliar esta versão em relação aos dois algoritmos utilizados na experimentação do Capítulo 4.

Neste capítulo, um algoritmo multiobjetivo baseado na Otimização por Colônia de Formigas é proposto, desenvolvido a partir do conceito de múltiplas espécies que podem percorrer as mesmas trilhas ao procurar comida, mas sem causar interferência na atualização de feromônio umas das outras. Uma vez que as formigas pertencem a diferentes espécies, elas só podem reconhecer seu próprio feromônio. Cada espécie se especializa em uma estratégia de busca. Não há um número pré-determinado de espécies a ser usado, e cada uma pode seguir uma estratégia de busca distinta. Também não há limite para o número de colônias por espécie e como estas podem interagir. Neste trabalho, duas espécies são utilizadas: uma com a busca estocástica da versão GRan e outra com o aspecto sistemático da versão FMu. Cada espécie prioriza todos os critérios simultaneamente, em vez de trabalhar com um objetivo específico.

A primeira versão algorítmica apresentada neste capítulo é denominada RubroNegra (R&B, do inglês *Red&Black*). Nesta versão as formigas são particionadas em dois conjun-

tos de tamanhos iguais. Cada conjunto representa uma espécie e usa sua própria estratégia de busca. As espécies não interagem umas com as outras. Um arquivo externo armazena as soluções não-dominadas encontradas por todas as formigas. Um possível revés nessa estratégia é que o uso de múltiplas espécies pode prejudicar a eficiência geral do algoritmo, uma vez que um esforço computacional maior é exigido. Com cada estratégia recebendo menos tempo computacional para trabalhar, isto pode resultar em conjuntos de aproximação piores que os gerados pelas versões algorítmicas apresentadas anteriormente. Para contornar essa limitação, um segundo conceito é proposto: a regulação por disponibilidade de comida.

A mimetização originalmente proposta por Dorigo (1992) surgiu a partir da observação de formigas procurando comida. No entanto, a habilidade destas de encontrar fontes de comida só é usada para a criação de um processo de reforço de aprendizado onde a informação surge a partir de depósitos de feromônio. Na natureza, se uma colônia de formigas consegue encontrar comida, esta cresce. Caso contrário, ela pode diminuir ao ponto de desaparecer. Neste trabalho, propõe-se que o mesmo fenômeno ocorra dentro das espécies de formigas artificiais. O algoritmo R&B básico é, então, modificado para permitir a regulação das espécies. Se a estratégia de busca sendo utilizada por uma espécie em particular mostra-se eficiente, isto significa que comida está sendo encontrada, o que permite que sua colônia cresça. Uma vez que as espécies estão procurando comida em um mesmo ambiente (espaço de busca) e que as fontes de comida são limitadas (o número de soluções eficientes não-dominadas de um problema de otimização combinatória multiobjetivo é limitado), o crescimento de uma espécie causa a diminuição da outra.

As versões reguladas por disponibilidade de comida também podem ser implementadas usando uma ou múltiplas colônias para cada espécie. Neste trabalho, o mesmo número de colônias e de formigas em cada colônia é usado para ambas as espécies. Um contador de novas soluções não-dominadas é associado a cada colônia. A cada iteração, todas as formigas de todas as colônias de todas as espécies constroem suas soluções. Quando uma formiga de uma dada colônia encontra uma nova solução não-dominada em relação ao arquivo global, o contador correspondente à colônia à qual pertence é incrementado. Para operacionalizar o processo de regulação por disponibilidade de comida, as colônias são agrupadas em pares. Cada par contém uma colônia de cada espécie. Ademais, estes pares permanecem os mesmos ao longo de toda a execução do algoritmo. A cada f_{gen} gerações, os contadores das colônias em cada par são comparados. Permite-se o crescimento das colônias que encontraram mais soluções não-dominadas, enquanto as demais encolhem. Esse processo é controlado por uma taxa de variação f_{rate} , inicialmente igual a 1. A

quantidade de formigas em cada colônia é dada por $f_{rate} \cdot n_{ij}$, onde n_{ij} representa o número de formigas na i -ésima colônia da espécie j , $i = 1, \dots, m$, $j = \{1, 2\}$. Quando uma colônia cresce, a variável f_{rate} é aumentada por um parâmetro Δf_{rate} . O oposto acontece quando o número de formigas é decrementado em uma colônia. O crescimento de uma colônia pode ser limitado ou ilimitado. Versões algorítmicas considerando cada uma dessas possibilidades são propostas.

Quatro variações que aplicam a regulação por disponibilidade de comida à versão R&B são investigadas. Duas destas variantes permitem crescimento ilimitados do tamanho das colônias. Nas outras variantes, um tamanho máximo é estabelecido: quando a variável f_{rate} de uma das colônias de um par chega a zero, a outra não pode mais crescer. Versões denominadas R&B-FRd e R&B-FRld correspondem à versão R&B com crescimento ilimitado e limitado das colônias, respectivamente (diferenciadas pelo sufixo l). A quantidade inicial de formigas em cada espécie também leva a diferentes variantes, particularmente no caso de experimentações com tempos de processamento limitados (como neste trabalho). Cada espécie pode receber a mesma quantidade de formigas que eram usadas nas versões com apenas uma espécie, ou pode haver uma divisão proporcional entre as espécies. Enquanto as versões R&B-FRd e R&B-FRld usam tal divisão proporcional (representada pelo sufixo d), nas versões R&B-FR e R&B-FRl a quantidade inicial de formigas por espécie é igual à quantidade total de formigas na versão original R&B (representada pela ausência do sufixo d). Novamente, o crescimento limitado ou ilimitado das colônias nestas variantes é diferenciado pelo sufixo l .

Uma vez que a versão R&B herda as características das versões GRan e FMu, ela também herda seus parâmetros. Experimentos foram realizados para $m \in \{1, 3\}$ e $d \in \{3, 6, 10\}$. A variação de valores para o parâmetro d com valores fixos do parâmetro m produziu comportamentos semelhantes. A Tabela 33 mostra os resultados obtidos para $m = 1$. Diferenças significativas são percebidas em seis instâncias completas pelo ranking de dominância. Já para as instâncias *grid*, observa-se diferença somente em uma instância, apontada pelo indicador de hipervolume. Comparações dois-a-dois no conjunto de instâncias completas mostram que os melhores resultados são obtidos para $d = 3$. Também são percebidas diferenças significativas entre $d = 6$ e $d = 10$, com a primeira configuração apresentando os melhores resultados.

Para o ajuste do parâmetro m , comparações foram realizadas usando para o parâmetro d os valores 3, 6 e 10. Os resultados são semelhantes, independente do valor deste parâmetro. Os p -valores obtidos com o teste bicaudal de Wilcoxon são apresentados na Tabela 34

Tabela 33: Comparação efetuada na experimentação do parâmetro d da versão R&B com $m = 1$.

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|-----------------|-------------------|---------|----|---------|-------------------|----------------|
| 1 | 6.47e-05 | - | - | 10 | 0.36788 | 0.36788 | 0.36788 |
| 2 | 8.32e-08 | - | - | 11 | 0.72339 | 0.21347 | 0.44775 |
| 3 | 0.00139 | - | - | 12 | NaN | 0.18566 | 0.32502 |
| 4 | 0.08576 | 0.75264 | 0.17745 | 13 | NaN | 0.17040 | 0.44394 |
| 5 | 0.60309 | 0.36788 | 0.60302 | 14 | NaN | 0.46112 | 0.31178 |
| 6 | 0.74014 | 0.52739 | 0.77601 | 15 | NaN | 0.68690 | 0.12025 |
| 7 | 0.00063 | - | - | 16 | 0.12535 | 0.29756 | 0.30742 |
| 8 | 2.17e-07 | - | - | 17 | NaN | 0.10729 | 0.00190 |
| 9 | 2.77e-07 | - | - | 18 | NaN | 0.61626 | 0.25463 |

($d = 3$). Resultados significativos foram encontrados em oito das nove instâncias *complete* e em três instâncias *grid* segundo os rankings de dominância, e para outras quatro instâncias *grid* com relação a pelo menos um indicador de qualidade. O teste para comparação de proporções mostra que a versão com apenas uma colônia apresenta desempenho superior à versão com 3 colônias para as instâncias *grid* (p -valor < 0.01), enquanto o oposto ocorre com as *complete* (p -valor > 0.99).

Tabela 34: p -valores obtidos com o teste bicaudal de Wilcoxon para a versão R&B com $m \in \{1, 3\}$ e $d = 3$.

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|-----------------|-------------------|---------|----|-----------------|-------------------|-----------------|
| 1 | 3.39e-05 | - | - | 10 | 0.00270 | - | - |
| 2 | 6.68e-05 | - | - | 11 | 4.29e-05 | - | - |
| 3 | 2.02e-06 | - | - | 12 | NaN | 0.00155 | 1.69e-17 |
| 4 | 4.47e-05 | - | - | 13 | NaN | 0.00677 | 1.64e-15 |
| 5 | 0.16040 | 0.16080 | 0.16042 | 14 | NaN | 0.37928 | 0.06112 |
| 6 | 0.00183 | - | - | 15 | NaN | 0.80926 | 0.53250 |
| 7 | 0.00012 | - | - | 16 | 2.12e-10 | - | - |
| 8 | 4.38e-05 | - | - | 17 | NaN | 4.32e-09 | 1.69e-17 |
| 9 | 0.00239 | - | - | 18 | NaN | 0.55379 | 0.00087 |

Na comparação das versões com três colônias R&B, GRan, FMu, Obj e OMix (as duas últimas usando atualização por origem e região simultaneamente), o teste de Kruskal-Wallis mostra diferenças significativas para todas as instâncias, como apresentado na Tabela 35. Os resultados listados na Tabela 36 mostram que, para o conjuntos *complete*, a versão R&B consegue gerar conjuntos de aproximação melhores que as versões FMu e Obj, mas não que a versão GRan. Diferenças significativas não são percebidas entre as versão R&B e OMix. Já para o conjunto *grid*, os conjuntos de aproximação só são considerados estatisticamente melhores que os conjuntos produzidos pela versão GRan.

Tabela 35: p -valores obtidos com o teste de Kruskal-Wallis para as versões com três colônias GRan, FMu, R&B, Obj e OMix, com $d = 3$.

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|-----------------|-------------------|---------|----|-----------------|-------------------|-----------------|
| 1 | 1.52e-28 | - | - | 10 | 0.00061 | - | - |
| 2 | 5.65e-28 | - | - | 11 | 1.01e-15 | - | - |
| 3 | 5.61e-28 | - | - | 12 | NaN | 3.18e-06 | 6.66e-25 |
| 4 | 1.06e-26 | - | - | 13 | NaN | 1.29e-08 | 1.85e-22 |
| 5 | 3.28e-22 | - | - | 14 | NaN | 0.75412 | 0.00014 |
| 6 | 1.28e-21 | - | - | 15 | NaN | 0.02611 | 0.05925 |
| 7 | 9.00e-27 | - | - | 16 | 4.29e-15 | - | - |
| 8 | 2.10e-28 | - | - | 17 | NaN | 3.80e-16 | 7.04e-26 |
| 9 | 7.79e-28 | - | - | 18 | NaN | 0.00011 | 1.27e-08 |

Tabela 36: Resultados da comparação de proporções dois-a-dois entre as versões com três colônias R&B, GRan, FMu, Obj e OMix (com $d = 3$).

| Complete | DR | | | | | DR + QI | | | | |
|----------|-------------|----------|-------------|----------|----------|----------|-------------|-------------|----------|----------|
| | GRan | FMu | R&B | Obj | OMix | GRan | FMu | R&B | Obj | OMix |
| GRan | - | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 | 0 |
| FMu | 1 | - | 1 | 0.5 | 1 | 1 | - | 1 | 0.5 | 1 |
| R&B | 1 | 0 | - | 0 | 1 | 1 | 0 | - | 0 | 0.25 |
| Obj | 1 | 0.5 | 1 | - | 1 | 1 | 0.5 | 1 | - | 1 |
| OMix | 1 | 0 | 1 | 0 | - | 1 | 0 | 0.75 | 0 | - |
| Grid | GRan | FMu | R&B | Obj | OMix | GRan | FMu | R&B | Obj | OMix |
| GRan | - | 0.91 | 0.75 | 0.96 | 0.96 | - | 1 | 1 | 1 | 1 |
| FMu | 0.09 | - | 0.5 | 0.5 | 0.09 | 0 | - | 0.02 | 0.98 | 0 |
| R&B | 0.25 | 0.5 | - | 0.96 | 0.91 | 0 | 0.98 | - | 1 | 1 |
| Obj | 0.04 | 0.5 | 0.04 | - | 0.75 | 0 | 0.02 | 0 | - | 0.75 |
| OMix | 0.04 | 0.91 | 0.09 | 0.25 | - | 0 | 0 | 0 | 0.25 | - |

Na comparação entre as versões R&B, GRan e FMu com apenas uma colônia, e as versões Obj e OMix com três colônias e atualização por origem e região simultaneamente, o teste de Kruskal-Wallis novamente aponta diferenças significativas para todas as instâncias. Neste caso, os conjuntos de aproximação gerados para as instâncias *grid* pela versão R&B são melhores que os conjuntos gerados pelas versões Obj e OMix, e não há diferença entre os conjuntos gerados pelas versões R&B e FMu. A Tabela 37 apresenta estes resultados. Para o conjunto de instâncias *complete*, no entanto, os conjuntos gerados pela versão R&B são melhores que os gerados pelas versões FMu e Obj, contudo piores que os produzidos pelas versões GRan e OMix.

Os resultados apresentados mostram que o uso de múltiplas espécies permite que o algoritmo gere conjuntos de aproximação tão bons quanto os produzidos pela versão FMu para as instâncias *grid*, mas não tão bons quanto os gerados pela versão GRan

Tabela 37: Comparação de proporções: versões R&B, GRan e FMu com apenas uma colônia, e versões Obj e OMix com 3 colônias e atualização por origem e região simultaneamente.

| | DR | | | | | DR + QI | | | | |
|----------|------|----------|----------|----------|----------|----------|----------|----------|-------------|-------------|
| Complete | GRan | FMu | R&B | Obj | OMix | GRan | FMu | R&B | Obj | OMix |
| GRan | - | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 | 0 |
| FMu | 1 | - | 1 | 0.5 | 1 | 1 | - | 1 | 0.5 | 1 |
| R&B | 1 | 0 | - | 0 | 1 | 1 | 0 | - | 0 | 1 |
| Obj | 1 | 0.5 | 1 | - | 1 | 1 | 0.5 | 1 | - | 1 |
| OMix | 1 | 0 | 0 | 0 | - | 1 | 0 | 0 | 0 | - |
| Grid | GRan | FMu | R&B | Obj | OMix | GRan | FMu | R&B | Obj | OMix |
| GRan | - | 0.75 | 0.91 | 0.5 | 1 | - | 1 | 1 | 1 | 1 |
| FMu | 0.25 | - | 0.5 | 0.5 | 0.5 | 0 | - | 0.09 | 0.04 | 0 |
| R&B | 0.09 | 0.5 | - | 0.5 | 0.5 | 0 | 0.91 | - | 0.04 | 0.01 |
| Obj | 0.5 | 0.5 | 0.5 | - | 0.5 | 0 | 0.96 | 0.96 | - | 0.09 |
| OMix | 0.25 | 0.5 | 0.5 | 0.5 | - | 0 | 1 | 0.99 | 0.91 | - |

para o conjunto *complete*. Estes resultados devem-se provavelmente ao que foi discutido anteriormente: a divisão dos recursos computacionais entre as diferentes espécies prejudica o desempenho geral do algoritmo. Para contornar esse problema, as versões R&B com regulação por disponibilidade de comida verificam automaticamente onde o esforço de busca está sendo recompensado e enviam mais recursos para essas colônias.

Além dos parâmetros m e d , no caso das versões com regulação por disponibilidade de comida é necessário regular também os parâmetros f_{gen} e Δf_{rate} . Os valores testados para o ajuste dessas versões foram: $m \in \{1, 3\}$ por espécie, $d \in \{1, 3, 6\}$, $\Delta f_{rate} \in \{0, 1; 0, 3; 0, 5\}$ e $f_{gen} \in \{1, 5\}$. Para o parâmetro d , os experimentos realizados mostraram que, quanto menor a quantidade de vetores de escalarização, melhores os conjuntos de aproximação gerados por todas as variantes para o conjunto de instâncias *complete*. Já para o conjuntos *grid*, o melhor desempenho observado ocorreu para a configuração $d = 3$. Dessa forma, o valor $d = 1$ foi adotado no restante do teste para as *complete* e o valor $d = 3$ para as *grid*. O segundo teste objetivou analisar os parâmetros Δf_{rate} e f_{gen} . Uma vez que estes são altamente correlacionados, todas as seis possíveis combinações foram comparadas. De forma geral, nenhuma configuração apresentou desempenho superior às demais, mas algumas se mostraram piores que todas as outras: $\Delta f_{rate} = 0, 1$ com $f_{gen} = 1$ e $\Delta f_{rate} = 0, 1$ com $f_{gen} = 5$. A configuração $\Delta f_{rate} = 0, 3$ com $f_{gen} = 1$ foi escolhida. Por fim, para a análise do número de colônias por espécies, os testes de Wilcoxon não mostraram diferença estatística entre os dois valores para o conjunto de instâncias *complete*, mas para as *grid* a versão com apenas uma colônia apresentou desempenho significativamente superior à versão com três colônias (p -valor $< 0, 01$).

Duas comparações gerais foram efetuadas entre a versão R&B e suas variantes. Inicialmente as versões com três colônias foram consideradas. A Tabela 38 mostra que o ranking de dominância identifica diferenças significativas para todas as instâncias *complete*. Ademais, em cinco das instâncias *grid* pelo menos um indicador de qualidade revela diferença significativa. Comparações dois-a-dois mostraram que todas as versões com regulação por disponibilidade de comida melhoram os resultados obtidos pela versão R&B original (p -valor $< 0,01$). O mesmo teste, no entanto, não identificou diferenças significativas entre as quatro variantes. Para as versões com apenas uma colônia, os mesmos resultados se mantêm para as instâncias *complete*, mas para *grid* não é apontada diferença significativa entre os conjuntos produzidos pela versão R&B e suas variantes.

Tabela 38: p -valores obtidos com o teste de Kruskal-Wallis na comparação das variantes com regulação por disponibilidade de comida em relação à versão R&B.

| # | DR | $I_{\epsilon+}^1$ | I_H^1 | # | DR | $I_{\epsilon+}^1$ | I_H^1 |
|---|-----------------|-------------------|---------|----|----------------|-------------------|-----------------|
| 1 | 1.17e-13 | - | - | 10 | 0.27478 | 0.23883 | 0.27224 |
| 2 | 5.82e-15 | - | - | 11 | NaN | 0.54268 | 0.00088 |
| 3 | 1.30e-30 | - | - | 12 | NaN | 0.75195 | 0.00031 |
| 4 | 7.22e-13 | - | - | 13 | NaN | 0.91722 | 0.23137 |
| 5 | 0.00165 | - | - | 14 | NaN | 0.01836 | 0.04177 |
| 6 | 1.37e-11 | - | - | 15 | NaN | 0.96750 | 0.41583 |
| 7 | 8.69e-10 | - | - | 16 | 0.03203 | - | - |
| 8 | 6.36e-29 | - | - | 17 | NaN | 0.16866 | 7.18e-05 |
| 9 | 1.07e-15 | - | - | 18 | NaN | 0.80297 | 0.51552 |

Uma vez que as variantes da versão R&B que usam regulação por disponibilidade de comida apresentam resultados semelhantes em relação à qualidade dos conjuntos de aproximação gerados, a variante R&B-FRd com uma colônia foi escolhida para ser comparada com as versões GRan e FMu (também com uma colônia), e com as versões com 3 colônias Obj e OMix que usam atualização por região e origem simultaneamente. O teste estatístico de comparação de proporções para a análise dois-a-dois mostrou que a versão R&B-FRd apresenta desempenho similar à versão GRan nas instâncias do conjunto *complete*, assim como em relação à versão FMu para o conjunto de instâncias *grid*. Estes resultados são apresentados na Tabela 39. Uma vez que esta variante consegue apresentar desempenhos tão bons quanto as melhores versões com uma espécie para ambos os conjuntos de instâncias, estes resultados confirmam que a abordagem com múltiplas espécies reguladas por disponibilidade de comida consegue identificar a estratégia de busca que mais se adequa a cada tipo de instância e direcionar os recursos computacionais. Dessa forma, essa abordagem consegue apresentar desempenho satisfatório independentemente da características das instâncias.

Tabela 39: Teste de comparação de proporções usando as versões com uma colônia R&B-FRd, GRan e FMu, e as versões com 3 colônias Obj e OMix (estas usando atualizações por origem e região simultaneamente).

| | DR | | | | | DR + QI | | | | |
|----------|------|----------|------|----------|----------|----------|----------|------|-------------|-------------|
| Complete | GRan | FMu | FRd | Obj | OMix | GRan | FMu | FRd | Obj | OMix |
| GRan | - | 0 | 0.09 | 0 | 0 | - | 0 | 0.09 | 0 | 0 |
| FMu | 1 | - | 1 | 0.5 | 1 | 1 | - | 1 | 0.5 | 1 |
| FRd | 0.91 | 0 | - | 0 | 0 | 0.91 | 0 | - | 0 | 0 |
| Obj | 1 | 0.5 | 1 | - | 1 | 1 | 0.5 | 1 | - | 1 |
| OMix | 1 | 0 | 1 | 0 | - | 1 | 0 | 1 | 0 | - |
| Grid | GRan | FMu | FRd | Obj | OMix | GRan | FMu | FRd | Obj | OMix |
| GRan | - | 0.75 | 0.75 | 0.5 | 0.75 | - | 1 | 1 | 1 | 1 |
| FMu | 0.25 | - | 0.5 | 0.5 | 0.5 | 0 | - | 0.5 | 0.09 | 0 |
| FRd | 0.25 | 0.5 | - | 0.5 | 0.5 | 0 | 0.5 | - | 0.04 | 0.01 |
| Obj | 0.5 | 0.5 | 0.5 | - | 0.5 | 0 | 0.91 | 0.96 | - | 0.09 |
| OMix | 0.25 | 0.5 | 0.5 | 0.5 | - | 0 | 1 | 0.99 | 0.91 | - |

Para concluir a experimentação das versões com regulação por disponibilidade de comida, a versão R&B-FRd com uma colônia é comparada aos dois otimizadores multiobjetivo utilizados na comparação do Capítulo 4. As mesmas implementações e configurações de parâmetro para ambos os algoritmos usados na experimentação daquele capítulo foram adotadas para esta comparação. Quanto aos rankings de dominância os resultados dos testes de Kruskal-Wallis apontam diferenças significativas para todas as instâncias. Os resultados dos testes dois-a-dois de comparação de proporções são mostradas na Tabela 40. Para o conjunto de instâncias *complete*, a variante R&B-FRd apresentam melhor desempenho dentre os três otimizadores. Ademais, o algoritmo HACO também é considerado estatisticamente melhor que o NSGA-II. Para o conjunto de instância *grid*, o algoritmo NSGA-II apresenta desempenho superior a implementação feita do HACO. Contudo, mais uma vez os conjuntos de aproximação produzidos pela variante R&B-FRd são considerados significativamente melhores que os conjuntos produzidos pelos demais otimizadores. Quando os conjuntos Pareto-ótimos são usados, todos os indicadores continuam favorecendo o algoritmo GRACE. Os resultados desta comparação podem ser vistos nas Tabelas 41–44, onde são listadas as medianas, mínimos, máximos, desvios-padrão e coeficientes de variação para cada indicador/algoritmo/instância.

Tabela 40: p -valores da comparação de proporções: R&B-FRd com uma colônia, NSGA-II (DEB et al., 2002) e HACO (HÄCKEL et al., 2008).

| | DR | | | DR + QI | | |
|----------|---------|----------|----------|---------|----------|----------|
| Complete | R&B-FRd | NSGA-II | HACO | R&B-FRd | NSGA-II | HACO |
| R&B-FRd | - | 0 | 0.25 | - | 0 | 0 |
| NSGA-II | 1 | - | 1 | 1 | - | 1 |
| HACO | 0.75 | 0 | - | 1 | 0 | - |
| Grid | R&B-FRd | NSGA-II | HACO | R&B-FRd | NSGA-II | HACO |
| R&B-FRd | - | 0 | 0 | - | 0 | 0 |
| NSGA-II | 1 | - | 0 | 1 | - | 0 |
| HACO | 1 | 1 | - | 1 | 1 | - |

Tabela 41: Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias completas.

| # | $I_{\epsilon+}$ | | | I_H | | | <i>gen. dist.</i> | | | <i>error ratio</i> | | | \mathcal{C}_{\prec} | | |
|---|-----------------|--------|--------|-------|-------|------|-------------------|--------|--------|--------------------|-------|------|-----------------------|-------|------|
| | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO |
| 1 | 43.00 | 248.00 | 248.00 | 0.00 | 0.04 | 0.03 | 0.00 | 30.68 | 27.47 | 0.00 | 0.13 | 0.17 | 0.00 | 0.13 | 0.17 |
| | 0.00 | 118.00 | 248.00 | 0.00 | 0.03 | 0.02 | 0.00 | 0.00 | 25.36 | 0.00 | 0.00 | 0.14 | 0.00 | 0.00 | 0.14 |
| | 192.00 | 412.00 | 290.00 | 0.00 | 0.06 | 0.03 | 37.06 | 68.14 | 35.90 | 0.07 | 0.36 | 0.22 | 0.07 | 0.36 | 0.22 |
| | 54.84 | 81.97 | 17.15 | 0.00 | 0.01 | 0.00 | 14.29 | 19.60 | 2.15 | 0.02 | 0.10 | 0.01 | 0.02 | 0.10 | 0.01 |
| | 1.05 | 0.34 | 0.07 | 1.34 | 0.23 | 0.08 | 1.21 | 0.63 | 0.08 | 1.16 | 0.71 | 0.09 | 1.16 | 0.71 | 0.09 |
| 2 | 120.00 | 339.00 | 120.00 | 0.00 | 0.10 | 0.01 | 5.92 | 50.34 | 5.17 | 0.03 | 0.36 | 0.03 | 0.03 | 0.36 | 0.03 |
| | 89.00 | 314.00 | 120.00 | 0.00 | 0.06 | 0.01 | 0.00 | 35.26 | 0.00 | 0.00 | 0.29 | 0.00 | 0.00 | 0.29 | 0.00 |
| | 194.00 | 386.00 | 120.00 | 0.01 | 0.14 | 0.01 | 27.58 | 93.37 | 14.25 | 0.09 | 0.62 | 0.07 | 0.09 | 0.62 | 0.07 |
| | 19.60 | 26.10 | 0.00 | 0.00 | 0.02 | 0.00 | 6.24 | 12.25 | 5.58 | 0.03 | 0.09 | 0.03 | 0.03 | 0.09 | 0.03 |
| | 0.16 | 0.08 | 0.00 | 0.27 | 0.19 | 0.16 | 1.12 | 0.23 | 0.91 | 0.98 | 0.22 | 0.82 | 0.98 | 0.22 | 0.82 |
| 3 | 152.00 | 425.00 | 324.00 | 0.01 | 0.13 | 0.03 | 15.42 | 78.46 | 21.87 | 0.15 | 0.62 | 0.30 | 0.15 | 0.62 | 0.30 |
| | 141.00 | 381.00 | 206.00 | 0.01 | 0.08 | 0.02 | 8.31 | 43.24 | 16.80 | 0.03 | 0.33 | 0.17 | 0.03 | 0.33 | 0.17 |
| | 233.00 | 565.00 | 363.00 | 0.01 | 0.20 | 0.04 | 21.00 | 102.14 | 32.02 | 0.28 | 0.92 | 0.43 | 0.28 | 0.92 | 0.43 |
| | 31.19 | 50.85 | 32.02 | 0.00 | 0.03 | 0.00 | 3.46 | 14.99 | 3.90 | 0.07 | 0.15 | 0.07 | 0.07 | 0.15 | 0.07 |
| | 0.18 | 0.12 | 0.10 | 0.14 | 0.27 | 0.15 | 0.23 | 0.19 | 0.17 | 0.44 | 0.23 | 0.23 | 0.44 | 0.23 | 0.23 |
| 4 | 132.00 | 245.00 | 230.00 | 0.00 | 0.03 | 0.02 | 0.00 | 35.03 | 24.02 | 0.00 | 0.18 | 0.17 | 0.00 | 0.18 | 0.17 |
| | 36.00 | 151.00 | 132.00 | 0.00 | 0.02 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 196.00 | 305.00 | 230.00 | 0.01 | 0.07 | 0.02 | 17.35 | 120.41 | 25.43 | 0.06 | 0.33 | 0.18 | 0.06 | 0.33 | 0.18 |
| | 47.35 | 38.16 | 42.07 | 0.00 | 0.01 | 0.00 | 6.13 | 20.93 | 8.06 | 0.02 | 0.08 | 0.06 | 0.02 | 0.08 | 0.06 |
| | 0.34 | 0.17 | 0.20 | 0.79 | 0.33 | 0.22 | 2.34 | 0.58 | 0.39 | 2.34 | 0.43 | 0.39 | 2.34 | 0.43 | 0.39 |
| 5 | 72.00 | 507.00 | 184.00 | 0.01 | 0.16 | 0.03 | 0.00 | 118.28 | 67.76 | 0.00 | 0.88 | 0.29 | 0.00 | 0.88 | 0.29 |
| | 72.00 | 196.00 | 121.00 | 0.00 | 0.07 | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 72.00 | 590.00 | 196.00 | 0.01 | 0.23 | 0.05 | 0.00 | 174.90 | 132.94 | 0.00 | 1.00 | 0.44 | 0.00 | 1.00 | 0.44 |
| | 0.00 | 127.55 | 27.08 | 0.00 | 0.04 | 0.01 | 0.00 | 30.39 | 35.00 | 0.00 | 0.25 | 0.14 | 0.00 | 0.25 | 0.14 |
| | 0.00 | 0.28 | 0.15 | 0.06 | 0.28 | 0.26 | nan | 0.26 | 0.53 | nan | 0.33 | 0.52 | nan | 0.33 | 0.52 |

Tabela 42: Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias completas (continuação).

| # | $I_{\epsilon+}$ | | | I_H | | | <i>gen. dist.</i> | | | <i>error ratio</i> | | | \mathcal{C}_{\prec} | | |
|---|-----------------|--------|--------|-------|-------|------|-------------------|--------|-------|--------------------|-------|------|-----------------------|-------|------|
| | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO | GRACE | NSGAI | HACO |
| 6 | 182.00 | 550.00 | 213.00 | 0.02 | 0.23 | 0.07 | 0.00 | 99.49 | 32.85 | 0.00 | 0.91 | 0.52 | 0.00 | 0.91 | 0.52 |
| | 135.00 | 386.00 | 188.00 | 0.02 | 0.16 | 0.06 | 0.00 | 52.12 | 18.11 | 0.00 | 0.33 | 0.33 | 0.00 | 0.33 | 0.33 |
| | 182.00 | 667.00 | 290.00 | 0.02 | 0.32 | 0.09 | 24.20 | 146.31 | 41.80 | 0.11 | 1.00 | 0.80 | 0.11 | 1.00 | 0.80 |
| | 13.03 | 63.08 | 32.95 | 0.00 | 0.04 | 0.01 | 6.30 | 23.41 | 5.25 | 0.03 | 0.15 | 0.11 | 0.03 | 0.15 | 0.11 |
| | 0.07 | 0.12 | 0.14 | 0.06 | 0.18 | 0.09 | 2.41 | 0.22 | 0.17 | 2.13 | 0.17 | 0.22 | 2.13 | 0.17 | 0.22 |
| 7 | 87.00 | 374.00 | 157.00 | 0.00 | 0.12 | 0.03 | 0.00 | 75.39 | 22.55 | 0.00 | 0.33 | 0.12 | 0.00 | 0.33 | 0.12 |
| | 35.00 | 157.00 | 157.00 | 0.00 | 0.05 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 87.00 | 535.00 | 157.00 | 0.02 | 0.26 | 0.03 | 30.46 | 149.49 | 39.45 | 0.06 | 0.83 | 0.24 | 0.06 | 0.83 | 0.24 |
| | 10.40 | 104.54 | 0.00 | 0.00 | 0.06 | 0.01 | 8.68 | 39.55 | 10.68 | 0.03 | 0.23 | 0.05 | 0.03 | 0.23 | 0.05 |
| | 0.12 | 0.28 | 0.00 | 0.65 | 0.46 | 0.20 | 1.34 | 0.59 | 0.47 | 1.26 | 0.68 | 0.49 | 1.26 | 0.68 | 0.49 |
| 8 | 188.00 | 498.00 | 331.00 | 0.01 | 0.13 | 0.04 | 18.28 | 63.82 | 25.39 | 0.41 | 0.67 | 0.50 | 0.41 | 0.67 | 0.50 |
| | 170.00 | 362.00 | 235.00 | 0.01 | 0.10 | 0.03 | 14.73 | 32.89 | 16.56 | 0.27 | 0.33 | 0.31 | 0.27 | 0.33 | 0.31 |
| | 406.00 | 638.00 | 341.00 | 0.03 | 0.16 | 0.06 | 29.22 | 117.63 | 31.69 | 0.50 | 0.90 | 0.65 | 0.50 | 0.90 | 0.65 |
| | 69.16 | 87.51 | 35.11 | 0.01 | 0.02 | 0.01 | 3.46 | 20.87 | 3.94 | 0.06 | 0.16 | 0.08 | 0.06 | 0.16 | 0.08 |
| | 0.30 | 0.18 | 0.11 | 0.36 | 0.14 | 0.18 | 0.18 | 0.31 | 0.16 | 0.14 | 0.25 | 0.16 | 0.14 | 0.25 | 0.16 |
| 9 | 167.00 | 480.00 | 312.00 | 0.01 | 0.19 | 0.06 | 8.02 | 98.43 | 30.72 | 0.10 | 1.00 | 0.58 | 0.10 | 1.00 | 0.58 |
| | 155.00 | 413.00 | 239.00 | 0.01 | 0.15 | 0.04 | 0.00 | 68.08 | 24.12 | 0.00 | 0.57 | 0.42 | 0.00 | 0.57 | 0.42 |
| | 232.00 | 687.00 | 312.00 | 0.02 | 0.29 | 0.08 | 13.12 | 152.54 | 58.98 | 0.21 | 1.00 | 0.68 | 0.21 | 1.00 | 0.68 |
| | 25.83 | 70.79 | 16.39 | 0.00 | 0.04 | 0.01 | 3.55 | 24.84 | 7.46 | 0.05 | 0.12 | 0.07 | 0.05 | 0.12 | 0.07 |
| | 0.15 | 0.14 | 0.05 | 0.26 | 0.19 | 0.18 | 0.45 | 0.24 | 0.23 | 0.56 | 0.13 | 0.13 | 0.56 | 0.13 | 0.13 |

Tabela 43: Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias *grid*.

| # | $I_{\epsilon+}$ | | | I_H | | | <i>gen. dist.</i> | | | <i>error ratio</i> | | | \mathcal{C}_{\prec} | | |
|----|-----------------|---------|----------|-------|--------|------|-------------------|--------|---------|--------------------|--------|------|-----------------------|--------|------|
| | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO |
| 10 | 0.00 | 145.00 | 835.00 | 0.00 | 0.00 | 0.05 | 0.00 | 4.83 | 78.66 | 0.00 | 0.01 | 0.70 | 0.00 | 0.01 | 0.70 |
| | 0.00 | 60.00 | 791.00 | 0.00 | 0.00 | 0.04 | 0.00 | 0.00 | 65.04 | 0.00 | 0.00 | 0.65 | 0.00 | 0.00 | 0.65 |
| | 0.00 | 550.00 | 996.00 | 0.00 | 0.01 | 0.06 | 0.00 | 42.04 | 86.07 | 0.00 | 0.14 | 0.76 | 0.00 | 0.14 | 0.76 |
| | 0.00 | 180.30 | 81.76 | 0.00 | 0.00 | 0.00 | 0.00 | 9.43 | 5.20 | 0.00 | 0.03 | 0.02 | 0.00 | 0.03 | 0.02 |
| | nan | 0.68 | 0.10 | nan | 1.33 | 0.08 | nan | 1.62 | 0.07 | nan | 1.91 | 0.03 | nan | 1.91 | 0.03 |
| 11 | 158.00 | 995.00 | 1230.00 | 0.00 | 0.01 | 0.04 | 1.93 | 16.25 | 93.40 | 0.02 | 0.33 | 0.85 | 0.02 | 0.33 | 0.85 |
| | 99.00 | 514.00 | 1095.00 | 0.00 | 0.00 | 0.04 | 0.70 | 9.57 | 77.44 | 0.01 | 0.13 | 0.77 | 0.01 | 0.13 | 0.77 |
| | 293.00 | 1417.00 | 1469.00 | 0.00 | 0.04 | 0.05 | 4.86 | 51.67 | 107.96 | 0.07 | 0.85 | 0.90 | 0.07 | 0.85 | 0.90 |
| | 58.32 | 284.49 | 83.38 | 0.00 | 0.01 | 0.00 | 1.20 | 9.83 | 9.30 | 0.02 | 0.21 | 0.04 | 0.02 | 0.21 | 0.04 |
| | 0.32 | 0.29 | 0.07 | 0.46 | 0.41 | 0.08 | 0.56 | 0.50 | 0.10 | 0.66 | 0.50 | 0.04 | 0.66 | 0.50 | 0.04 |
| 12 | 295.00 | 777.00 | 3895.00 | 0.00 | 0.01 | 0.15 | 2.64 | 21.26 | 710.23 | 0.01 | 0.55 | 1.00 | 0.01 | 0.55 | 1.00 |
| | 205.00 | 571.00 | 3541.00 | 0.00 | 0.00 | 0.14 | 1.32 | 10.96 | 605.74 | 0.00 | 0.25 | 1.00 | 0.00 | 0.25 | 1.00 |
| | 295.00 | 1061.00 | 4439.00 | 0.00 | 0.03 | 0.16 | 3.99 | 37.63 | 845.28 | 0.04 | 0.86 | 1.00 | 0.04 | 0.86 | 1.00 |
| | 23.95 | 145.84 | 236.84 | 0.00 | 0.01 | 0.00 | 0.68 | 7.10 | 51.76 | 0.01 | 0.20 | 0.00 | 0.01 | 0.20 | 0.00 |
| | 0.09 | 0.18 | 0.06 | 0.19 | 0.52 | 0.02 | 0.24 | 0.33 | 0.07 | 0.52 | 0.36 | 0.00 | 0.52 | 0.36 | 0.00 |
| 13 | 560.00 | 2107.00 | 7356.00 | 0.00 | 0.05 | 0.20 | 5.17 | 35.49 | 1518.03 | 0.19 | 0.99 | 1.00 | 0.19 | 0.99 | 1.00 |
| | 442.00 | 1261.00 | 6579.00 | 0.00 | 0.03 | 0.19 | 4.60 | 14.67 | 1166.41 | 0.15 | 0.33 | 1.00 | 0.15 | 0.33 | 1.00 |
| | 738.00 | 3740.00 | 7806.00 | 0.00 | 0.08 | 0.21 | 5.82 | 91.76 | 2284.55 | 0.22 | 1.00 | 1.00 | 0.22 | 1.00 | 1.00 |
| | 77.95 | 642.24 | 302.54 | 0.00 | 0.01 | 0.01 | 0.33 | 18.31 | 309.94 | 0.02 | 0.14 | 0.00 | 0.02 | 0.14 | 0.00 |
| | 0.14 | 0.29 | 0.04 | 0.09 | 0.26 | 0.03 | 0.07 | 0.41 | 0.19 | 0.10 | 0.15 | 0.00 | 0.10 | 0.15 | 0.00 |
| 14 | 1260.00 | 5614.00 | 14692.00 | 0.01 | 0.10 | 0.22 | 31.11 | 107.39 | 2948.92 | 0.28 | 1.00 | 1.00 | 0.28 | 1.00 | 1.00 |
| | 1129.00 | 3975.00 | 12563.00 | 0.01 | 0.06 | 0.21 | 23.52 | 42.48 | 2412.18 | 0.21 | 1.00 | 1.00 | 0.21 | 1.00 | 1.00 |
| | 1525.00 | 7509.00 | 16028.00 | 0.01 | 0.13 | 0.22 | 35.92 | 293.52 | 4411.80 | 0.34 | 1.00 | 1.00 | 0.34 | 1.00 | 1.00 |
| | 104.48 | 1019.18 | 785.92 | 0.00 | 0.02 | 0.00 | 3.33 | 60.92 | 490.75 | 0.03 | 0.00 | 0.00 | 0.03 | 0.00 | 0.00 |
| | 0.08 | 0.18 | 0.05 | 0.01 | 0.19 | 0.02 | 0.11 | 0.54 | 0.16 | 0.12 | 0.00 | 0.00 | 0.12 | 0.00 | 0.00 |

Tabela 44: Em ordem, mediana, mínimo, máximo, desvio padrão, coeficiente de variação. Instâncias *grid* (continuação).

| # | $I_{\epsilon+}$ | | | I_H | | | <i>gen. dist.</i> | | | <i>error ratio</i> | | | \mathcal{C}_{\prec} | | |
|----|-----------------|---------|----------|-------|--------|------|-------------------|--------|---------|--------------------|--------|------|-----------------------|--------|------|
| | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO | GRACE | NSGAII | HACO |
| 15 | 1338.00 | 7262.00 | 16878.00 | 0.01 | 0.11 | 0.25 | 24.54 | 140.37 | 4846.35 | 0.32 | 1.00 | 1.00 | 0.32 | 1.00 | 1.00 |
| | 1228.00 | 4690.00 | 15489.00 | 0.01 | 0.08 | 0.25 | 21.34 | 67.54 | 3784.51 | 0.27 | 1.00 | 1.00 | 0.27 | 1.00 | 1.00 |
| | 1431.00 | 9562.00 | 17868.00 | 0.01 | 0.14 | 0.26 | 30.58 | 280.21 | 9520.06 | 0.39 | 1.00 | 1.00 | 0.39 | 1.00 | 1.00 |
| | 42.30 | 1254.47 | 624.46 | 0.00 | 0.02 | 0.00 | 2.10 | 58.54 | 1343.29 | 0.03 | 0.00 | 0.00 | 0.03 | 0.00 | 0.00 |
| | 0.03 | 0.18 | 0.04 | 0.01 | 0.16 | 0.01 | 0.08 | 0.38 | 0.26 | 0.09 | 0.00 | 0.00 | 0.09 | 0.00 | 0.00 |
| 16 | 12.00 | 392.00 | 1307.00 | 0.00 | 0.00 | 0.05 | 0.00 | 7.25 | 145.19 | 0.00 | 0.06 | 0.68 | 0.00 | 0.06 | 0.68 |
| | 0.00 | 252.00 | 1176.00 | 0.00 | 0.00 | 0.05 | 0.00 | 3.40 | 112.45 | 0.00 | 0.02 | 0.50 | 0.00 | 0.02 | 0.50 |
| | 80.00 | 571.00 | 1646.00 | 0.00 | 0.01 | 0.06 | 0.00 | 20.15 | 206.34 | 0.00 | 0.45 | 0.86 | 0.00 | 0.45 | 0.86 |
| | 23.15 | 87.57 | 95.50 | 0.00 | 0.00 | 0.00 | 0.00 | 5.07 | 25.55 | 0.00 | 0.10 | 0.09 | 0.00 | 0.10 | 0.09 |
| | 1.21 | 0.21 | 0.07 | 1.96 | 0.56 | 0.05 | nan | 0.56 | 0.17 | nan | 1.01 | 0.14 | nan | 1.01 | 0.14 |
| 17 | 590.00 | 2736.00 | 6928.00 | 0.00 | 0.05 | 0.16 | 6.39 | 52.46 | 966.56 | 0.34 | 1.00 | 1.00 | 0.34 | 1.00 | 1.00 |
| | 530.00 | 1774.00 | 6266.00 | 0.00 | 0.04 | 0.15 | 5.75 | 19.78 | 862.92 | 0.30 | 0.78 | 1.00 | 0.30 | 0.78 | 1.00 |
| | 735.00 | 4649.00 | 7590.00 | 0.00 | 0.12 | 0.16 | 7.06 | 444.68 | 1203.21 | 0.40 | 1.00 | 1.00 | 0.40 | 1.00 | 1.00 |
| | 54.74 | 643.74 | 374.40 | 0.00 | 0.02 | 0.00 | 0.38 | 84.67 | 76.92 | 0.02 | 0.05 | 0.00 | 0.02 | 0.05 | 0.00 |
| | 0.09 | 0.23 | 0.05 | 0.05 | 0.31 | 0.02 | 0.06 | 1.17 | 0.08 | 0.07 | 0.05 | 0.00 | 0.07 | 0.05 | 0.00 |
| 18 | 1341.00 | 5185.00 | 10459.00 | 0.00 | 0.10 | 0.20 | 17.80 | 172.01 | 2679.09 | 0.31 | 1.00 | 1.00 | 0.31 | 1.00 | 1.00 |
| | 1128.00 | 3707.00 | 9890.00 | 0.00 | 0.08 | 0.18 | 12.34 | 77.44 | 2150.68 | 0.26 | 1.00 | 1.00 | 0.26 | 1.00 | 1.00 |
| | 1496.00 | 6875.00 | 11411.00 | 0.00 | 0.13 | 0.20 | 25.21 | 361.40 | 5355.85 | 0.39 | 1.00 | 1.00 | 0.39 | 1.00 | 1.00 |
| | 109.82 | 839.56 | 318.68 | 0.00 | 0.01 | 0.00 | 2.89 | 64.08 | 628.51 | 0.03 | 0.00 | 0.00 | 0.03 | 0.00 | 0.00 |
| | 0.08 | 0.16 | 0.03 | 0.04 | 0.14 | 0.02 | 0.16 | 0.37 | 0.23 | 0.10 | 0.00 | 0.00 | 0.10 | 0.00 | 0.00 |

7 Considerações finais

O interesse crescente no desenvolvimento de metaheurísticas aplicadas a problemas com múltiplos objetivos tem levantado questões de projeto que levam a muitas abordagens distintas. Neste trabalho, o Problema do Caminho mais Curto Multiobjetivo (MSP) foi investigado, e um algoritmo baseado na Otimização por Colônias de Formigas foi proposto, denominado GRACE. Este algoritmo foi comparado com dois otimizadores multiobjetivo encontrados na literatura, usando dois conjuntos de instâncias triobjetivo com características estruturais distintas, e os resultados apresentados atestam que o algoritmo proposto gera conjuntos de aproximação melhores que os demais otimizadores para todas as instâncias testadas. Ademais, uma extensa revisão foi realizada com o objetivo de identificar as diferentes abordagens encontradas na literatura das Colônias de Formigas Multiobjetivo (MOACO), e várias dessas propostas foram testadas como extensão do algoritmo GRACE. Duas versões algorítmicas se destacaram: uma variante com características estocásticas (equivalente ao algoritmo GRACE original) gerou os melhores conjuntos de aproximação para o primeiro grupo de instâncias, enquanto uma abordagem sistemática prevaleceu no segundo grupo. Não obstante, notou-se também que a versão estocástica apresenta pior desempenho para o segundo conjunto de instâncias, e que o mesmo acontece com a versão sistemática para o primeiro grupo.

Diferentes tentativas de integração das características das versões campeãs foram realizadas, porém nenhuma conseguiu igualar as demais para ambos os conjuntos de instâncias. Foram propostos, então, dois novos conceitos para o desenvolvimento de MOACOs: o uso de múltiplas espécies de formigas, que só reconhecem seu próprio feromônio e utilizam cada uma sua própria estratégia de busca, e a regulação por disponibilidade de comida, que permite que as espécies cujas abordagens mostrem-se eficientes possam crescer, recebendo mais recursos computacionais. Os experimentos realizados mostraram que o uso dos conceitos propostos neste trabalho permitem que um único algoritmo gere os melhores conjuntos de aproximação para os diferentes tipos de instâncias testados.

O horizonte de possibilidades aberto para investigações futuras é rico. A literatura

do MSP, por exemplo, carece de uma investigação que possa avaliar mais claramente as virtudes e fraquezas de cada otimizador proposto. É necessário, também, a definição de um banco de instâncias comum mais amplo, tanto em termos de objetivos, como de tamanho e características estruturais. Com relação às abordagens publicadas na literatura concernentes à Colônias de Formigas Multiobjetivo, é necessário pensar como generalizar conceitos encontrados no contexto biobjetivo, como a ordenação de soluções para divisão de regiões. Por fim, para os conceitos propostos neste trabalho, é interessante analisar como interações podem ser promovidas entre as diferentes espécies em um algoritmo, e ponderar diferentes métricas para a avaliação do desempenho das espécies.

Referências

- ALAYA, I.; SOLNON, C.; GHÉDIRA, K. Ant colony optimization for multi-objective optimization problems. In: *IEEE International Conference on Tools with Artificial Intelligence*. [S.l.: s.n.], 2007. p. 450–457.
- ANGUS, D.; WOODWARD, C. Multiple objective ant colony optimisation. *Swarm Intell.*, v. 3, p. 69–85, 2009.
- BEZERRA, L. C. T. et al. GRACE: A generational randomized ACO for the multi-objective shortest path problem. In: TAKAHASHI, R. et al. (Ed.). *Evolutionary Multi-criterion Optimization*. [S.l.]: Springer, 2011, (Lecture Notes in Computer Science, v. 6576). p. 535–549.
- BLEULER, S. et al. Pisa a platform and programming language independent interface for search algorithms. In: FONSECA, C. et al. (Ed.). *EMO 2003*. [S.l.]: Springer Berlin / Heidelberg, 2003, (Lecture Notes in Computer Science, v. 2632). p. 1–1.
- BURIOL, L. S. et al. A hibrid genetic algorithm for the weight setting problem in ospf/is-is routing. *Networks*, N.Y. Print, New York, v. 46, n. 1, p. 36–56, 2005.
- CORNE, D. W. et al. The good of the many outweighs the good of the one: Evolutionary multi-objective optimization. *Connections. The Newsletter of the IEEE Neural Networks Society*, p. 9–13, Fevereiro 2003.
- COSTELLOE, D.; MOONEY, P.; WINSTANSLEY, A. From random walks to pareto optimal paths. In: *Proceedings of the 4th Irish Artificial Intelligence and Cognitive Science Conference*. [S.l.: s.n.], 2001. p. 523–530.
- CZYZAK, P.; JASZKIEWICZ, A. Pareto simulated annealing: A metaheuristic technique for multiple-objective combinatorial optimization. *J. Multi-Criteria Decis. Anal.*, v. 7, p. 34–47, 1998.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans. Evol. Comput.*, v. 6, n. 2, p. 182–197, 2002.
- DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Nr. math.*, 1959.
- DOERNER, K. et al. Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Ann. Oper. Res.*, Springer Netherlands, v. 131, p. 79–99, 2004. ISSN 0254-5330.
- DOERNER, K. et al. Pareto ant colony optimization with ilp preprocessing in multiobjective project portfolio selection. *Eur. J. Oper. Res.*, Elsevier, v. 171, p. 830–841, 2006.

- DOERNER, K.; HARTL, R. F.; REIMANN, M. Are COMPETants more competent for problem solving? - the case of a multiple objective transportation problem. In: *Proceedings of the GECCO'01*. Berlin, Heidelberg: Morgan Kaufmann, 2001. p. 802.
- DORIGO, M. *Optimization, learning and natural algorithms*. Tese (Doutorado) — Dipartimento de Elettronica, Politecnico de Milano, 1992.
- DORIGO, M.; GAMBARDELLA, L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evolut. Comput.*, v. 1, n. 1, p. 53–66, 1997.
- DORIGO, M.; SOCHA, K. *An introduction to Ant Colony Optimization*. [S.l.], 2006.
- GARCÍA-MARTÍNEZ, C.; CORDÓN, O.; HERRERA, F. A taxonomy and empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *Eur. J. Oper. Res.*, v. 180, n. 1, p. 116–148, 2007.
- GHOSEIRI, K.; NADJARI, B. An ant colony optimization algorithm for the bi-objective shortest path problem. *Appl. Soft Comput.*, v. 10, n. 4, p. 1237–1246, 2010.
- GRASSÉ, P.-P. La reconstruction du nid et les coordinations interindividuelles chez *bellicositermes natalensis* et *cubitermes* sp. La théorie de la stigmergie: essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, v. 6, p. 41, 1959.
- GUNTSCH, M.; MIDDENDORF, M. Solving multi-criteria optimization problems with population-based ACO. In: *EMO 2003*. [S.l.]: Springer, 2003, (Lecture Notes in Computer Science, v. 2632). p. 464–478.
- HÄCKEL, S. et al. A multi-objective ant colony approach for pareto-optimization using dynamic programming. In: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO)*. [S.l.]: ACM, 2008. p. 33–40.
- HANSEN, M. Tabu search for multiobjective combinatorial optimization. *Control. and Cybern.*, v. 29, p. 799–818, 2000.
- HANSEN, M. P.; JASZKIEWICZ, A. *Evaluating the quality of approximations to the non-dominated set*. [S.l.], 1998.
- HANSEN, P. Bicriterion path problems. In: FANDEL, G.; GAL, T. (Ed.). *Multiple criteria decision making, theory and application. Proceedings of the 3rd international conference*. [S.l.]: Springer Berlin, 1980, (Lecture Notes in Economics and Mathematical Systems, v. 177). p. 109–127.
- HE, F.; QI, H.; FAN, Q. An evolutionary algorithm for the multi-objective shortest path problem. In: *Proceedings of the Advances in Intelligent Systems Research (ISKE)*. [S.l.: s.n.], 2007.
- IREDI, S.; MERKLE, D.; MIDDENDORF, M. Bi-criterion optimization with multicolony ant algorithms. In: ZITZLER, E. et al. (Ed.). *EMO 2001*. [S.l.]: Springer, 2001, (Lecture Notes in Computer Science, v. 1993). p. 359–372.

ISHIBUCHI, H.; MURATA, T. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Trans. Syst., Man, Cybern., Part C: Appl. and Rev.*, v. 28, n. 3, p. 392–403, 1998.

KNOWLES, J.; CORNE, D. Instance generators and test suites for the multiobjective quadratic assignment problem. Springer Berlin / Heidelberg, v. 2632, p. 295–310, 2003.

KNOWLES, J. D.; THIELE, L.; ZITZLER, E. *A tutorial on the performance assessment of stochastic multiobjective optimizers*. [S.l.], 2006.

KRUSKAL, W. H.; WALLIS, W. A. Use of ranks in one-criterion variance analysis. *J. Am. Stat. Assoc.*, American Statistical Association, v. 47, n. 260, p. pp. 583–621, 1952. ISSN 01621459. Disponível em: <<http://www.jstor.org/stable/2280779>>.

LIN, L.; GEN, M. A bicriteria shortest path routing problems by hybrid genetic algorithm in communication networks. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2007. p. 4577–4582.

LÓPEZ-IBÁÑEZ; STÜTZLE, T. *Automatic configuration of Multi-objective ACO algorithms*. [S.l.], 2000.

LÓPEZ-IBÁÑEZ, M.; PAQUETE, L.; STÜTZLE, T. On the design of ACO for the biobjective quadratic assignment problem. In: DORIGO, M. et al. (Ed.). *ANTS 2004*. [S.l.]: Springer-Verlag, 2004, (Lecture Notes in Computer Science, v. 3172).

LÓPEZ-IBÁÑEZ, M.; STÜTZLE, T. An analysis of algorithmic components for multiobjective ant colony optimization: A case study on the biobjective tsp. In: COLLET, P. et al. (Ed.). *Artificial Evolution*. [S.l.]: Springer Berlin / Heidelberg, 2010, (Lecture Notes in Computer Science, v. 5975). p. 134–145.

LÓPEZ-IBÁÑEZ, M.; STÜTZLE, T. The impact of design choices of multiobjective ant colony optimization algorithms on performance: an experimental study on the biobjective tsp. In: *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2010. (GECCO '10), p. 71–78. ISBN 978-1-4503-0072-8. Disponível em: <<http://doi.acm.org/10.1145/1830483.1830494>>.

MARTINS, E. Q. V. On a multicriteria shortest path problem. *Eur. J. Oper. Res.*, v. 16, p. 236 – 245, 1984.

MICHAEL, C. M.; STEWART, C. V.; KELLY, R. B. Reducing the search time of a steady state genetic algorithm using the immigration operator. In: *Proceedings of the IEEE Congress on Tools for AI*. [S.l.: s.n.], 1991. p. 500–501.

MOONEY, P.; WINSTANSLEY, A. An evolutionary algorithm for multicriteria path optimization problems. *Int. J. Geogr. Inf. Sci.*, v. 20, n. 4, p. 401 – 423, 2006.

MSPP. 2010. www.mat.uc.pt/~zeluis/INVESTIG/MSPP/mspp.htm.

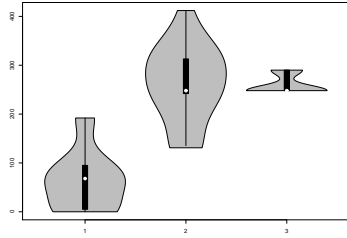
PANGILINAN, J. M. A.; JANSEENS, G. K. Evolutionary algorithms for the multiobjective shortest path problem. *Int. J. Comput. Inf. Sci. Eng.*, v. 1, 2007.

- PEREIRA, C. M. N. A. Evolutionary multicriteria optimization in core designs: basic investigation and case study. *Annals of Nuclear Energy*, Elsevier, v. 31, p. 1251–1331, 2004.
- RAITH, A.; EHRGOTT, M. A comparison of solution strategies for the biobjective shortest path problem. *Comput. & Oper. Res.*, v. 36, n. 4, p. 1299–1331, 2009.
- SERAFINI, P. Some considerations about computational complexity for multi objective combinatorial problems. In: JAHN, J.; KRABS, W. (Ed.). *Recent advances and historical development of vector optimization*. Berlin: Springer Verlag, 1986, (Lecture Notes in Economics and Mathematical Systems, v. 294).
- SKRIVER, A. J. V. A classification of bicriterion shortest path (bsp) algorithms. *Asia-Pac. J. Oper. Res.*, v. 17, p. 199 – 212, 2000.
- SKRIVER, A. J. V.; ANDERSEN, K. A. A label correcting approach for solving bicriterion shortest-path problems. *Comput. & Oper. Res.*, v. 27, n. 6, p. 507 – 524, 2000.
- SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolut. Comput.*, v. 2, n. 3, p. 221–248, 1994.
- STEINER, S.; RADZIK, T. *Solving the biobjective minimum spanning tree problem using a k-best algorithm*. [S.l.], 2003.
- STÜTZLE, T.; HOOS, H. H. Max-min ant system. *Future Gener. Comput. Syst.*, v. 16, p. 889–914, 2000.
- TAILLARD, E.; WAELTI, P.; ZUBER, J. Few statistical tests for proportions comparison. *Eur. J. Oper. Res.*, v. 185, p. 1336–1350, 2009.
- TAILLARD, E. D. Robust taboo search for the quadratic assignment problem. *Parallel Comput.*, v. 17, p. 443–455, 1991.
- TARAPATA, Z. Military route planning in battlefield simulation: effectiveness problems and potential solutions. *J. Telecomm. Inf. Technol.*, n. 4, p. 47–56, 2003.
- ULUNGU, E. L.; TEGHEM, J. The two phases method: an efficient procedure to solve bi-objective combinatorial optimization problems. *Foundations of Computing and Decision Sciences*, v. 20, p. 149–165, 1995.
- VELDHUIZEN, D. A. V. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Tese (Doutorado) — Graduate School of Engineering of the Air Force Institute of Technology, Air University, Junho 1999.
- WILCOXON, F. Individual comparisons by ranking methods. *Biom. bull.*, v. 1, n. 6, p. 80–83, 1945.
- ZITZLER, E. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Tese (Doutorado) — Institut für Technische Informatik und Kommunikationsnetze, November 1999.
- ZITZLER, E.; LAUMANN, M.; THIELE, L. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. [S.l.], 2001.

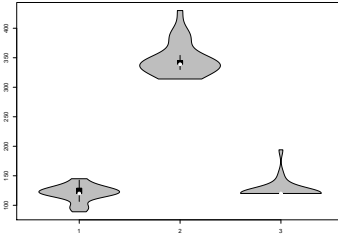
ZITZLER, E.; THIELE, L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto evolutionary algorithm. *IEEE Trans. Evolut. Comput.*, v. 3, n. 4, p. 257–271, 1999.

ZITZLER, E. et al. *Performance assessment of multiobjective optimizers: an analysis and review*. [S.l.], 2002.

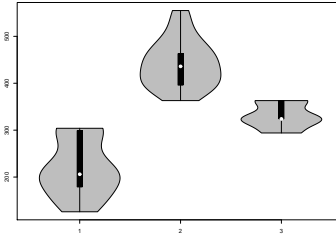
Anexo 1 – Vioplots Cap. 4



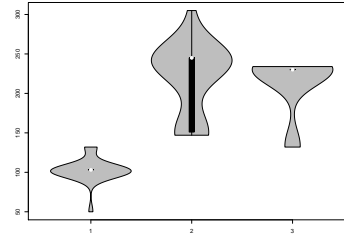
(a) CompleteN-small/n25s45



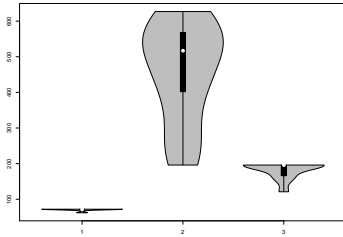
(b) CompleteN-small/n50s12



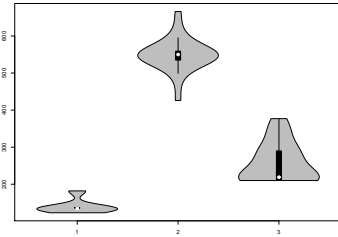
(c) CompleteN-small/n100s13



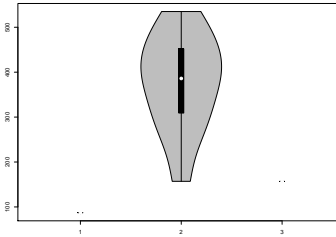
(d) CompleteN-medium/n40s18



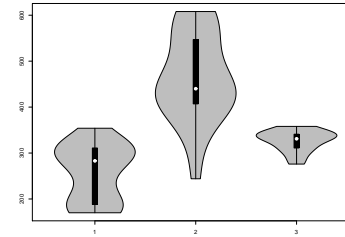
(e) CompleteN-medium/n120s14



(f) CompleteN-medium/n200s21

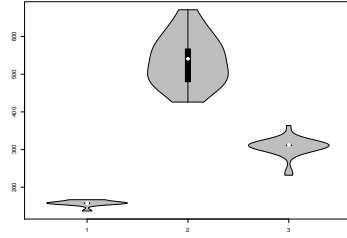


(g) CompleteN-large/n100s1

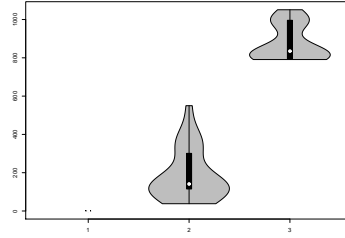


(h) CompleteN-large/n150s10

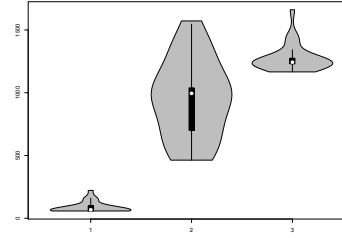
Figura 5: Violin plots para o indicador $I_{\epsilon+}$ da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO.



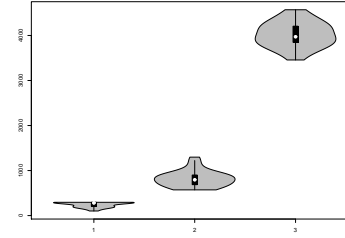
(a) CompleteN-large/n200s1



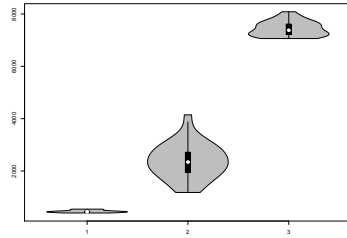
(b) GridN-small/n8s14



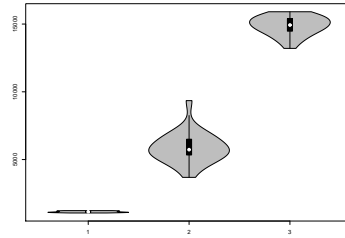
(c) GridN-small/n12s1



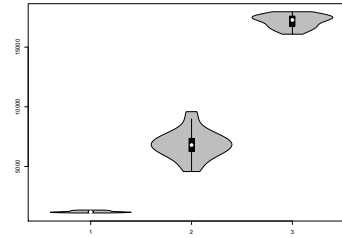
(d) GridN-small/n16s26



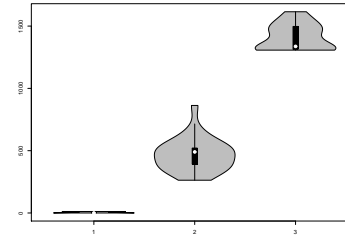
(e) GridN-medium/n22s1



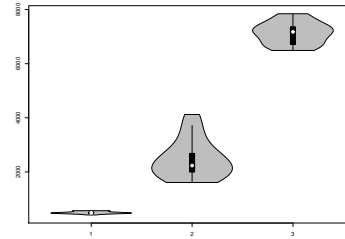
(f) GridN-medium/n31s40



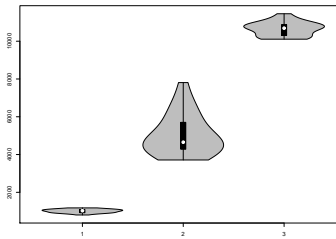
(g) GridN-medium/n35s2



(h) GridN-large/n11s41



(i) GridN-large/n22s42



(j) GridN-large/n30s43

Figura 6: Violin plots para o indicador I_{ϵ_+} da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO.

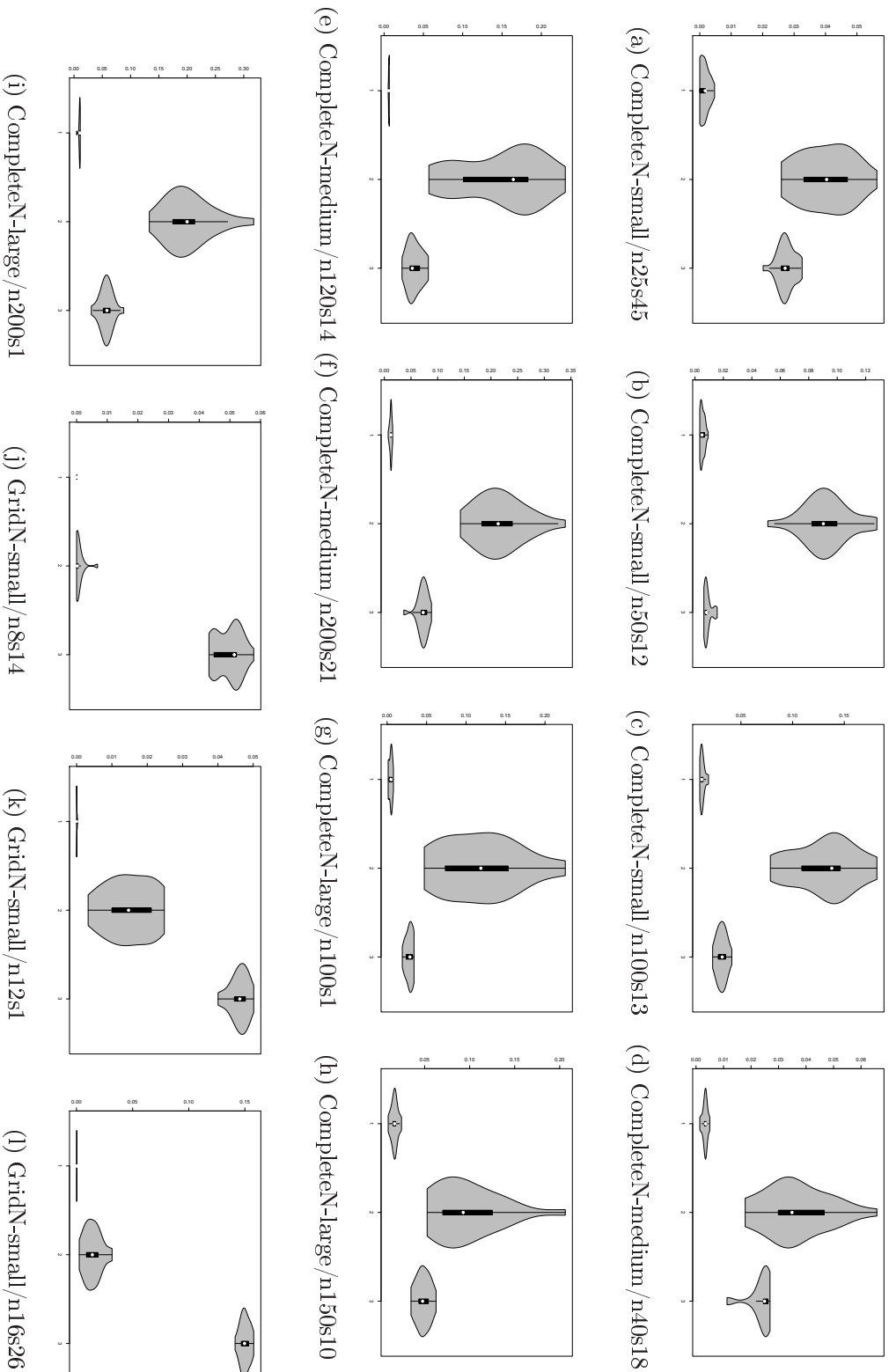


Figura 7: Violin plots para o indicador I_H da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO.

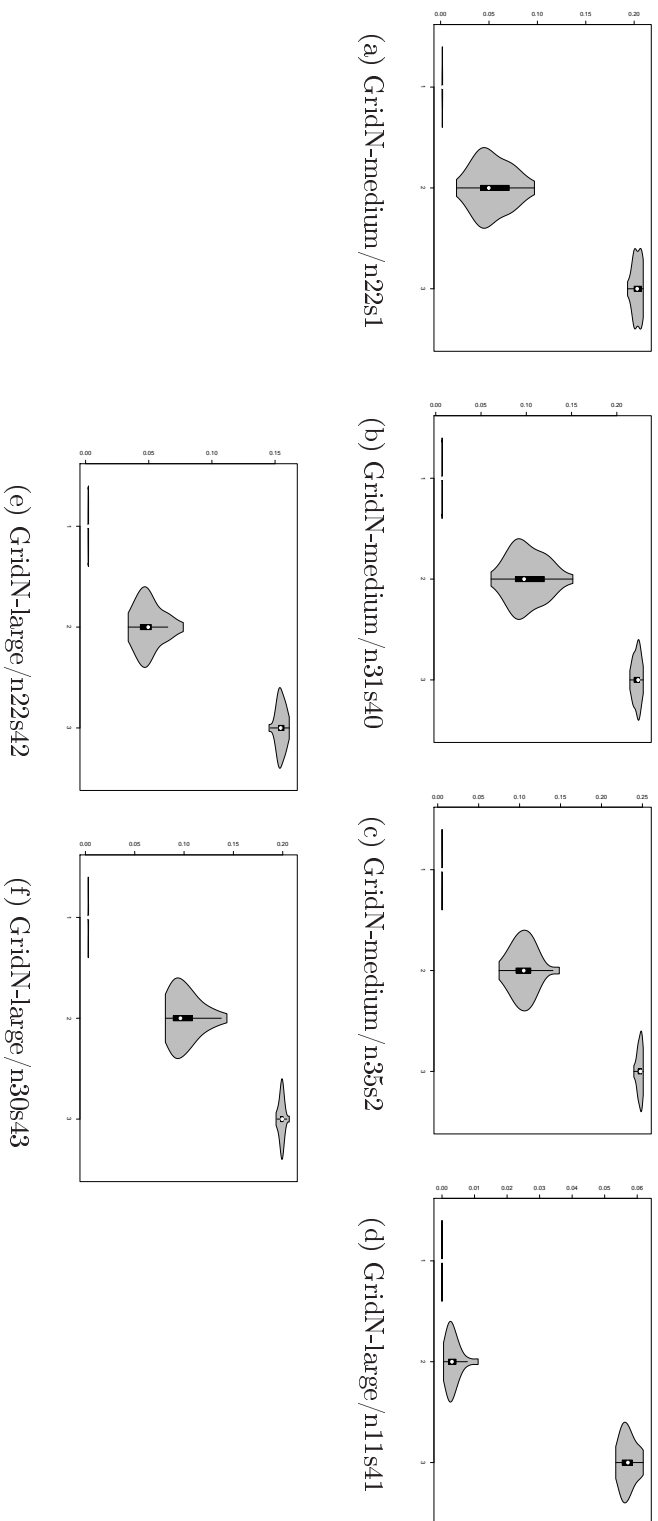
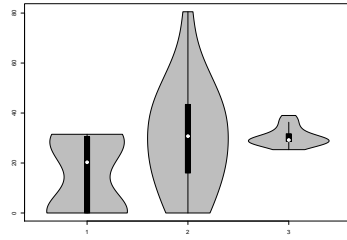
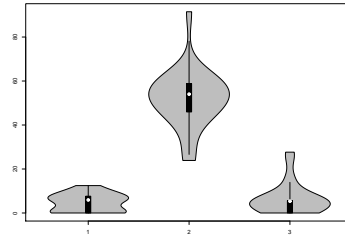


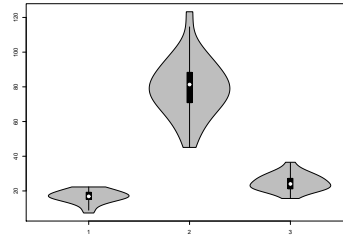
Figura 8: Violin plots para o indicador I_H da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO.



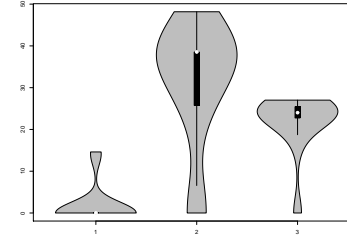
(a) CompleteN-small/n25s45



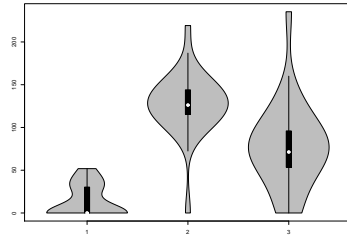
(b) CompleteN-small/n50s12



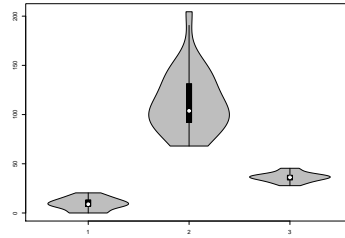
(c) CompleteN-small/n100s13



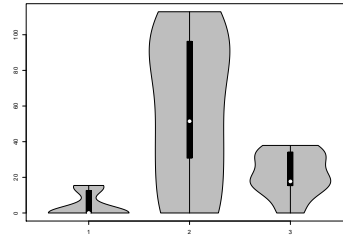
(d) CompleteN-medium/n40s18



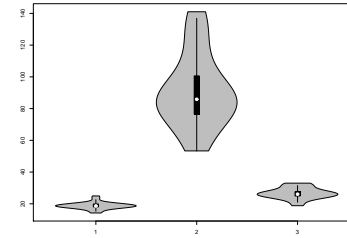
(e) CompleteN-medium/n120s14



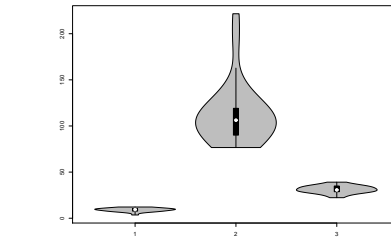
(f) CompleteN-medium/n200s21



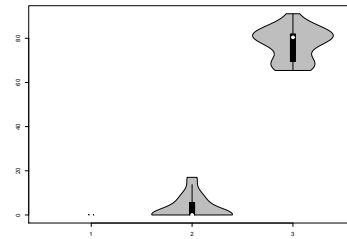
(g) CompleteN-large/n100s1



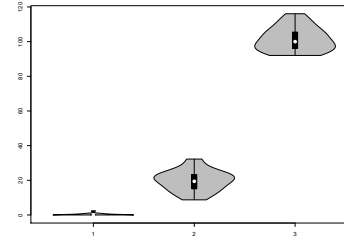
(h) CompleteN-large/n150s10



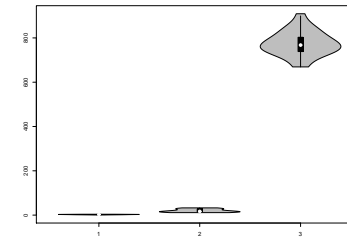
(i) CompleteN-large/n200s1



(j) GridN-small/n8s14

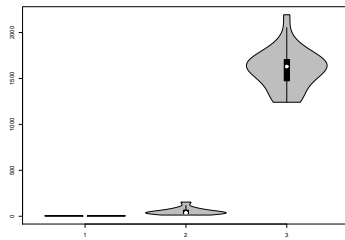


(k) GridN-small/n12s1

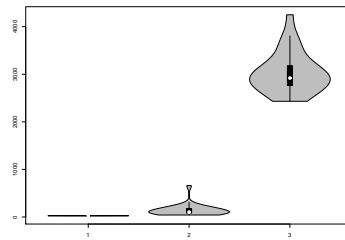


(l) GridN-small/n16s26

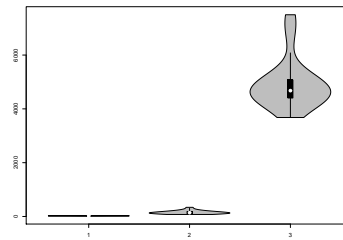
Figura 9: Violin plots para a distância geracional da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO.



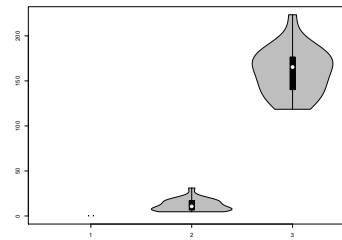
(a) GridN-medium/n22s1



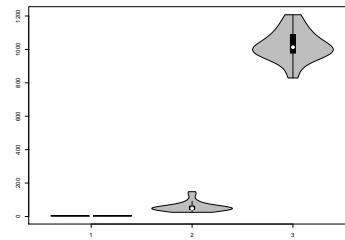
(b) GridN-medium/n31s40



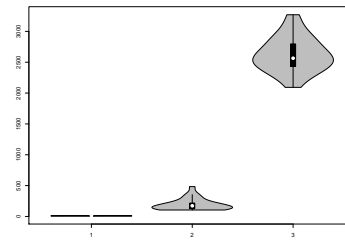
(c) GridN-medium/n35s2



(d) GridN-large/n11s41

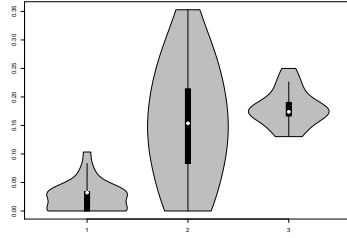


(e) GridN-large/n22s42

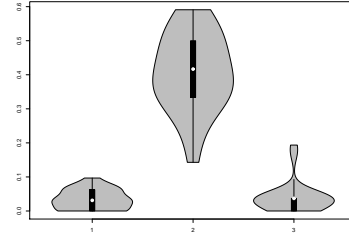


(f) GridN-large/n30s43

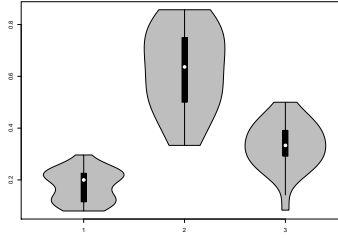
Figura 10: Violin plots para a distância geracional da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO.



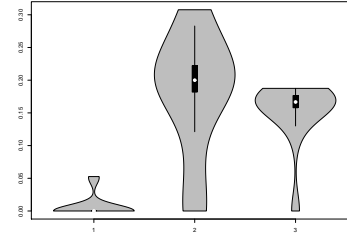
(a) CompleteN-small/n25s45



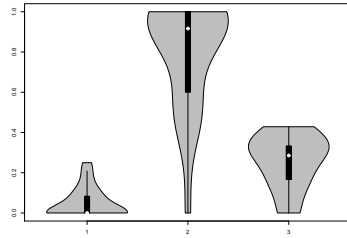
(b) CompleteN-small/n50s12



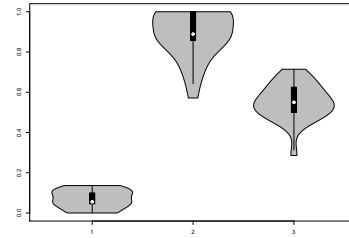
(c) CompleteN-small/n100s13



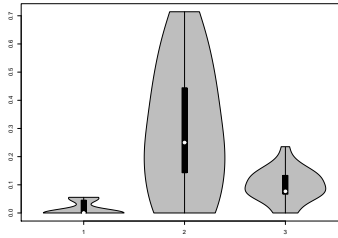
(d) CompleteN-medium/n40s18



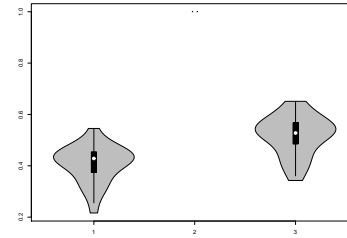
(e) CompleteN-medium/n120s14



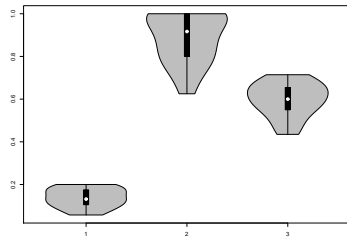
(f) CompleteN-medium/n200s21



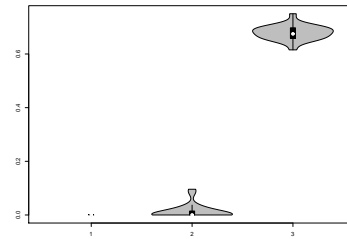
(g) CompleteN-large/n100s1



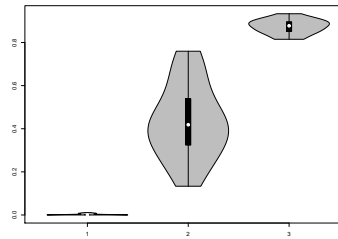
(h) CompleteN-large/n150s10



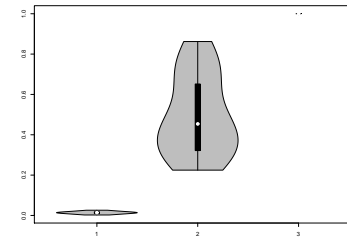
(i) CompleteN-large/n200s1



(j) GridN-small/n8s14

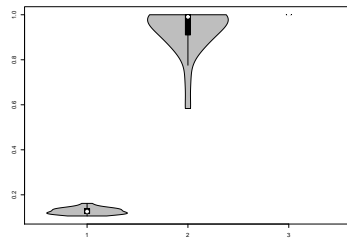


(k) GridN-small/n12s1

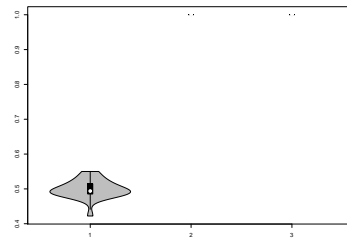


(l) GridN-small/n16s26

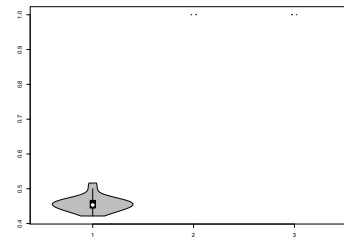
Figura 11: Violin plots para a taxa de erro da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO.



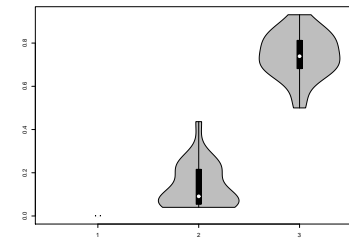
(a) GridN-medium/n22s1



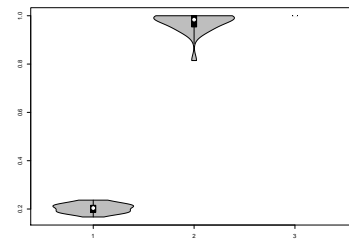
(b) GridN-medium/n31s40



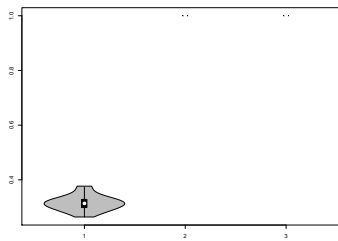
(c) GridN-medium/n35s2



(d) GridN-large/n11s41

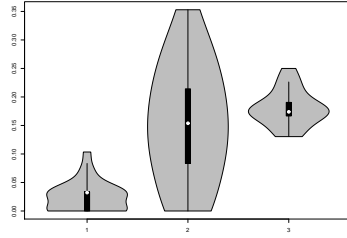


(e) GridN-large/n22s42

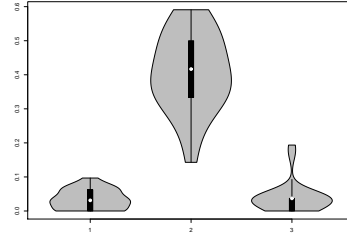


(f) GridN-large/n30s43

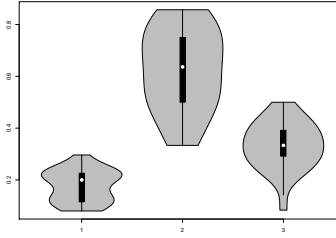
Figura 12: Violin plots para a taxa de erro da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO.



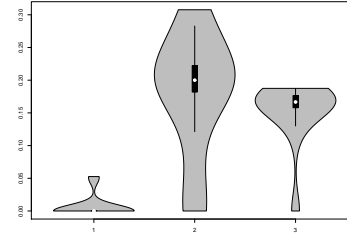
(a) CompleteN-small/n25s45



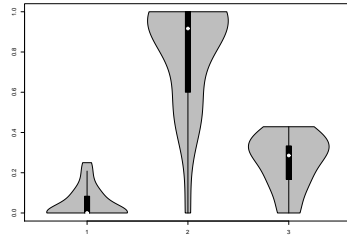
(b) CompleteN-small/n50s12



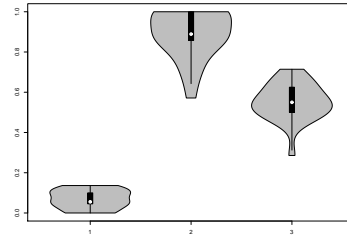
(c) CompleteN-small/n100s13



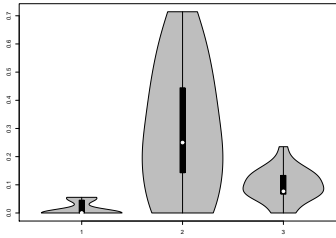
(d) CompleteN-medium/n40s18



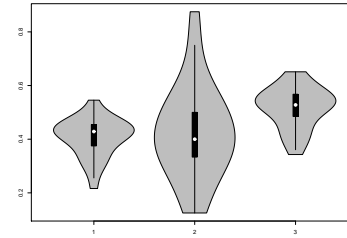
(e) CompleteN-medium/n120s14



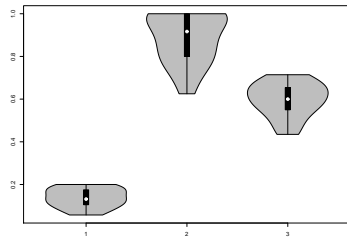
(f) CompleteN-medium/n200s21



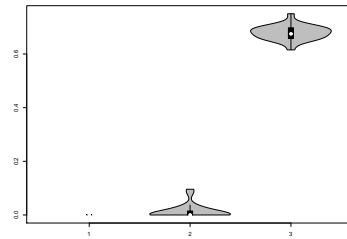
(g) CompleteN-large/n100s1



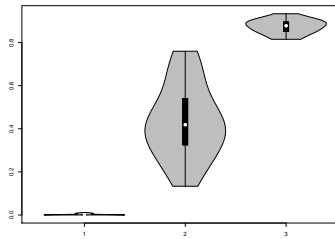
(h) CompleteN-large/n150s10



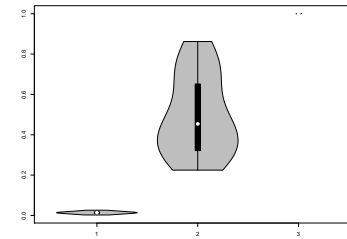
(i) CompleteN-large/n200s1



(j) GridN-small/n8s14

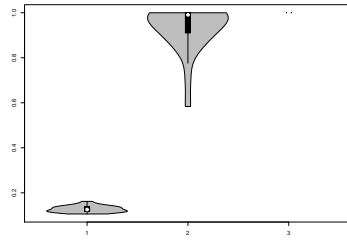


(k) GridN-small/n12s1

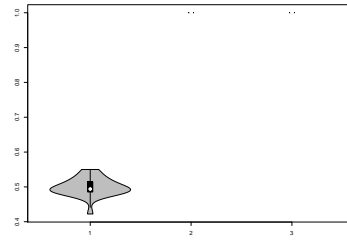


(l) GridN-small/n16s26

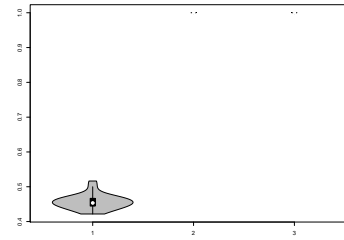
Figura 13: Violin plots para a métrica \mathcal{C}_\prec da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO.



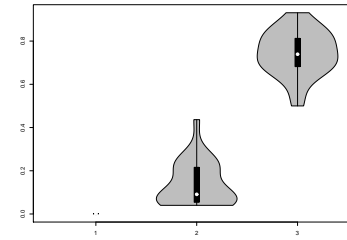
(a) GridN-medium/n22s1



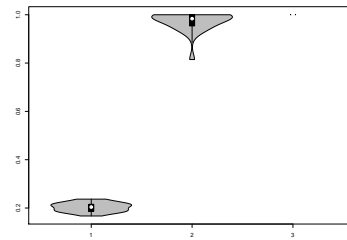
(b) GridN-medium/n31s40



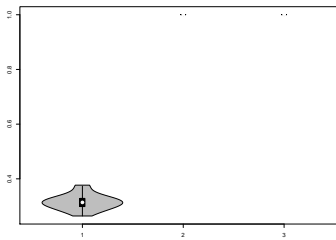
(c) GridN-medium/n35s2



(d) GridN-large/n11s41



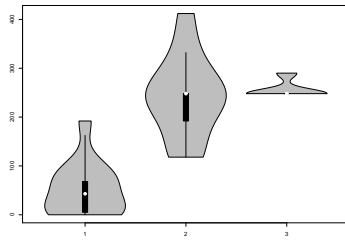
(e) GridN-large/n22s42



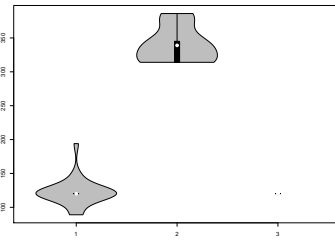
(f) GridN-large/n30s43

Figura 14: Violin plots para a métrica \mathcal{C}_\prec da comparação entre o algoritmo GRACE original, o NSGA-II e o HACO.

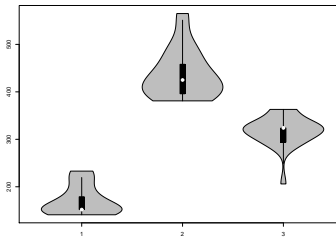
Anexo 2 – Vioplots Cap. 6



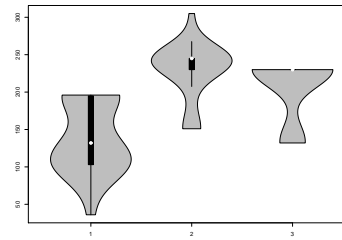
(a) CompleteN-small/n25s45



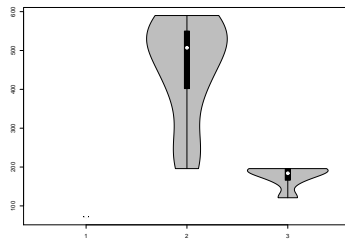
(b) CompleteN-small/n50s12



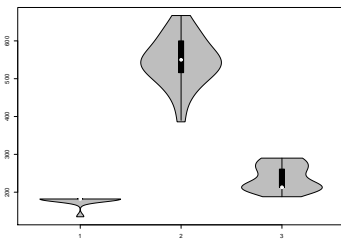
(c) CompleteN-small/n100s13



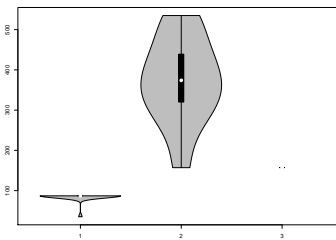
(d) CompleteN-medium/n40s18



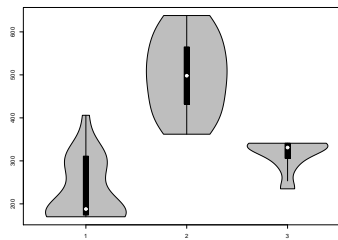
(e) CompleteN-medium/n120s14



(f) CompleteN-medium/n200s21



(g) CompleteN-large/n100s1



(h) CompleteN-large/n150s10

Figura 15: Violin plots para o indicador $I_{\epsilon+}$ da comparação entre a versão R&B-FRd, o NSGA-II e o HACO.

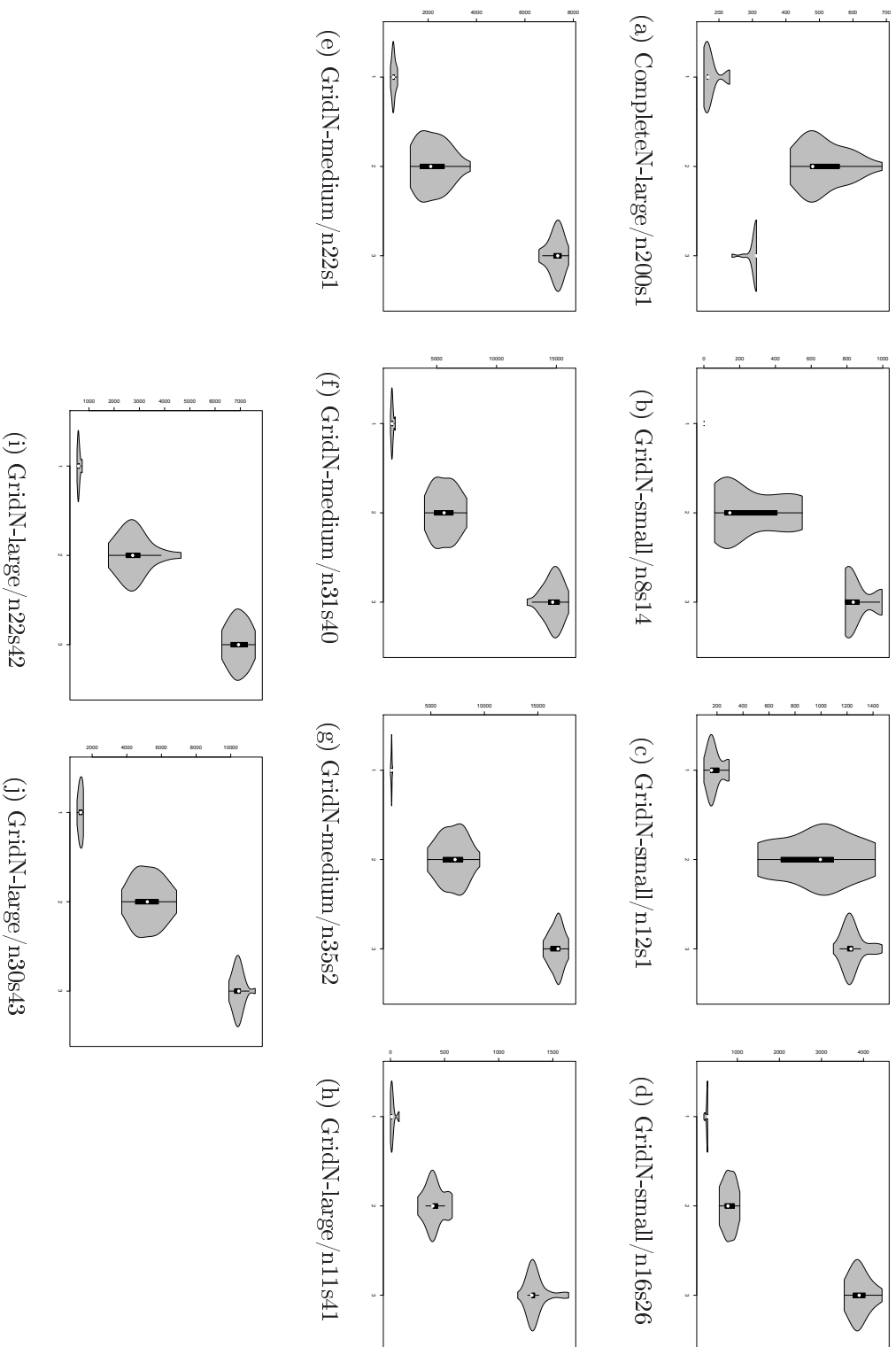


Figura 16: Violin plots para o indicador $I_{\epsilon+}$ da comparação entre a versão R&B-FRd, o NSGA-II e o HACO.

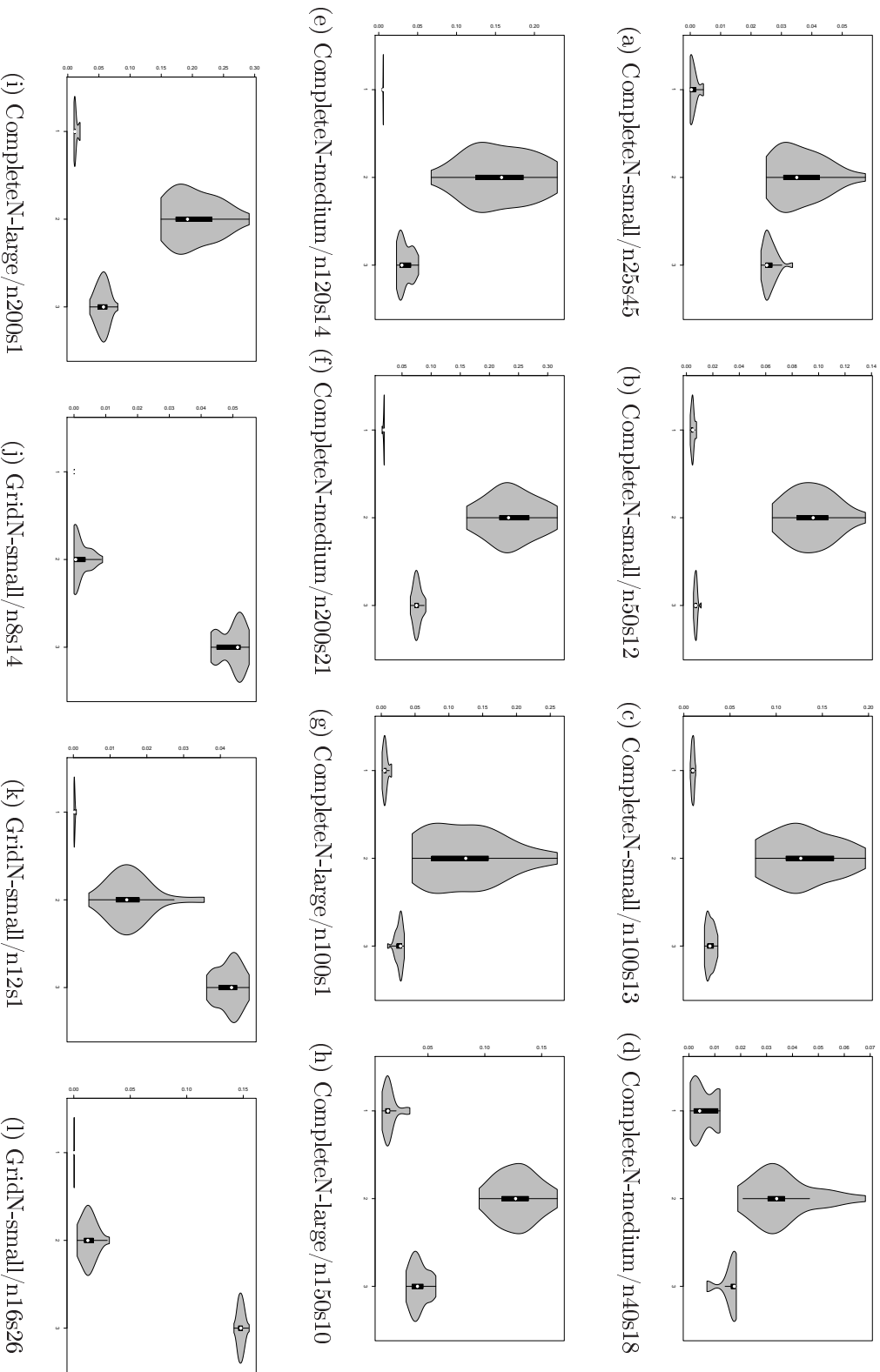


Figura 17: Violin plots para o indicador I_H^- da comparação entre a versão R&B-FRd, o NSGA-II e o HACO.

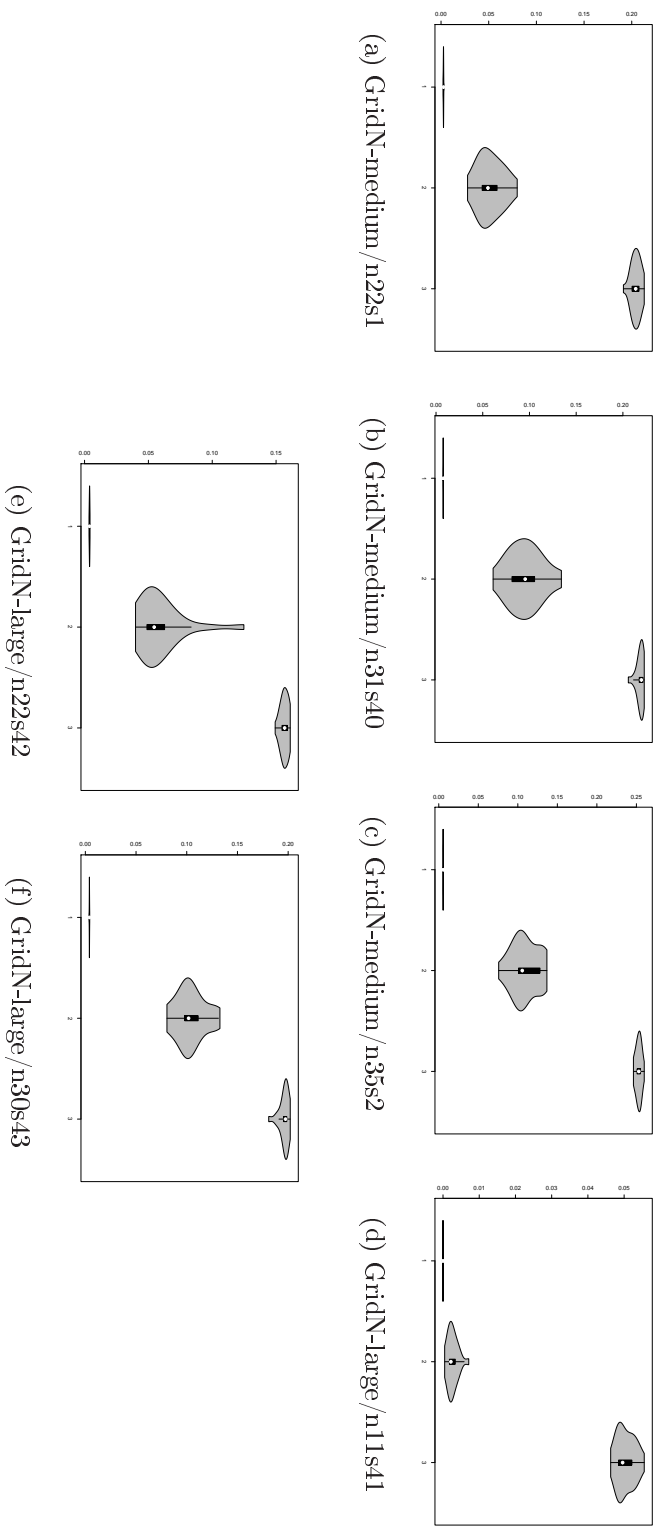
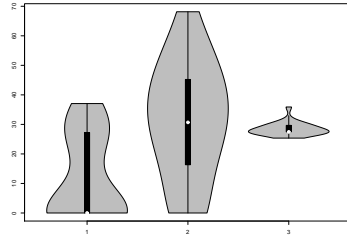
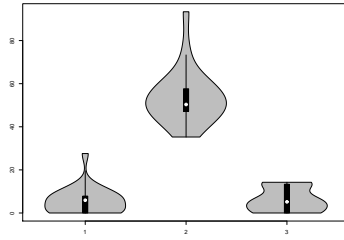


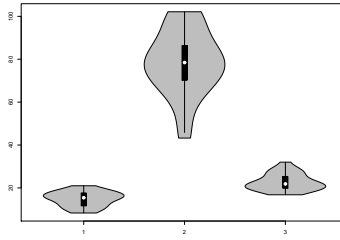
Figura 18: Violin plots para o indicador I_H da comparação entre a versão R&B-FRd, o NSGA-II e o HACO.



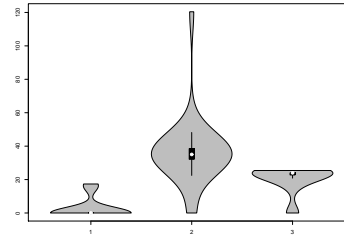
(a) CompleteN-small/n25s45



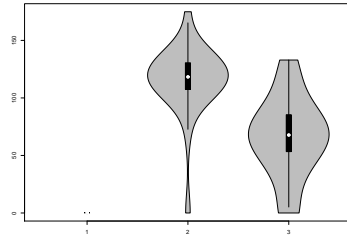
(b) CompleteN-small/n50s12



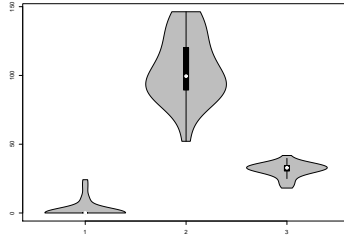
(c) CompleteN-small/n100s13



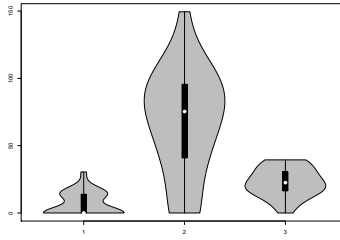
(d) CompleteN-medium/n40s18



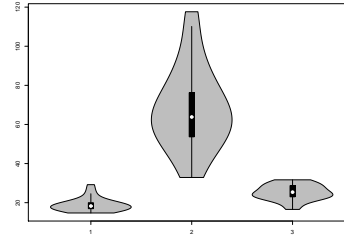
(e) CompleteN-medium/n120s14



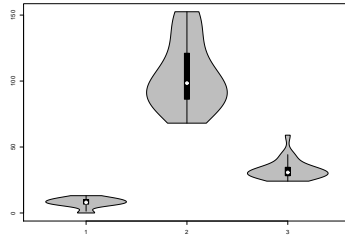
(f) CompleteN-medium/n200s21



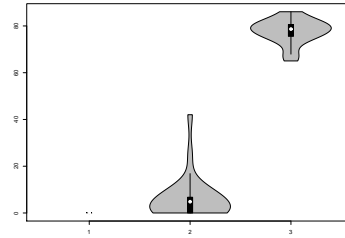
(g) CompleteN-large/n100s1



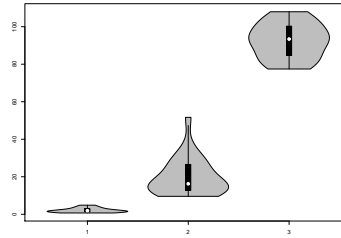
(h) CompleteN-large/n150s10



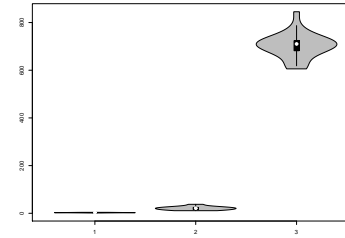
(i) CompleteN-large/n200s1



(j) GridN-small/n8s14

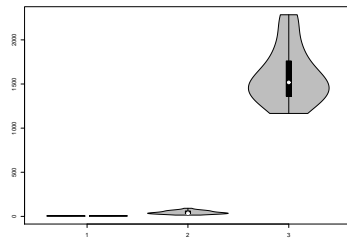


(k) GridN-small/n12s1

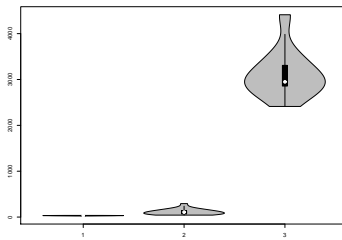


(l) GridN-small/n16s26

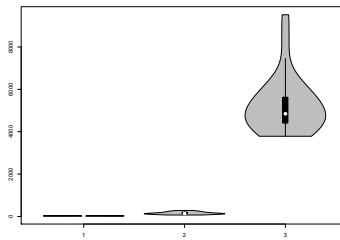
Figura 19: Violin plots para a distância geracional da comparação entre a versão R&B-FRd, o NSGA-II e o HACO.



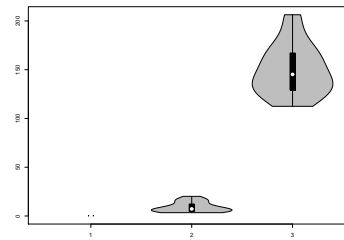
(a) GridN-medium/n22s1



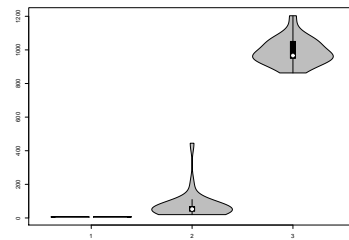
(b) GridN-medium/n31s40



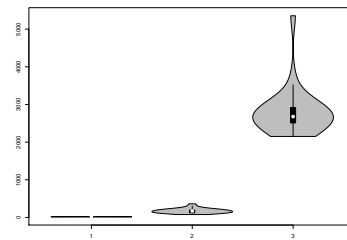
(c) GridN-medium/n35s2



(d) GridN-large/n11s41



(e) GridN-large/n22s42



(f) GridN-large/n30s43

Figura 20: Violin plots para a distância geracional da comparação entre a versão R&B-FRd, o NSGA-II e o HACO.

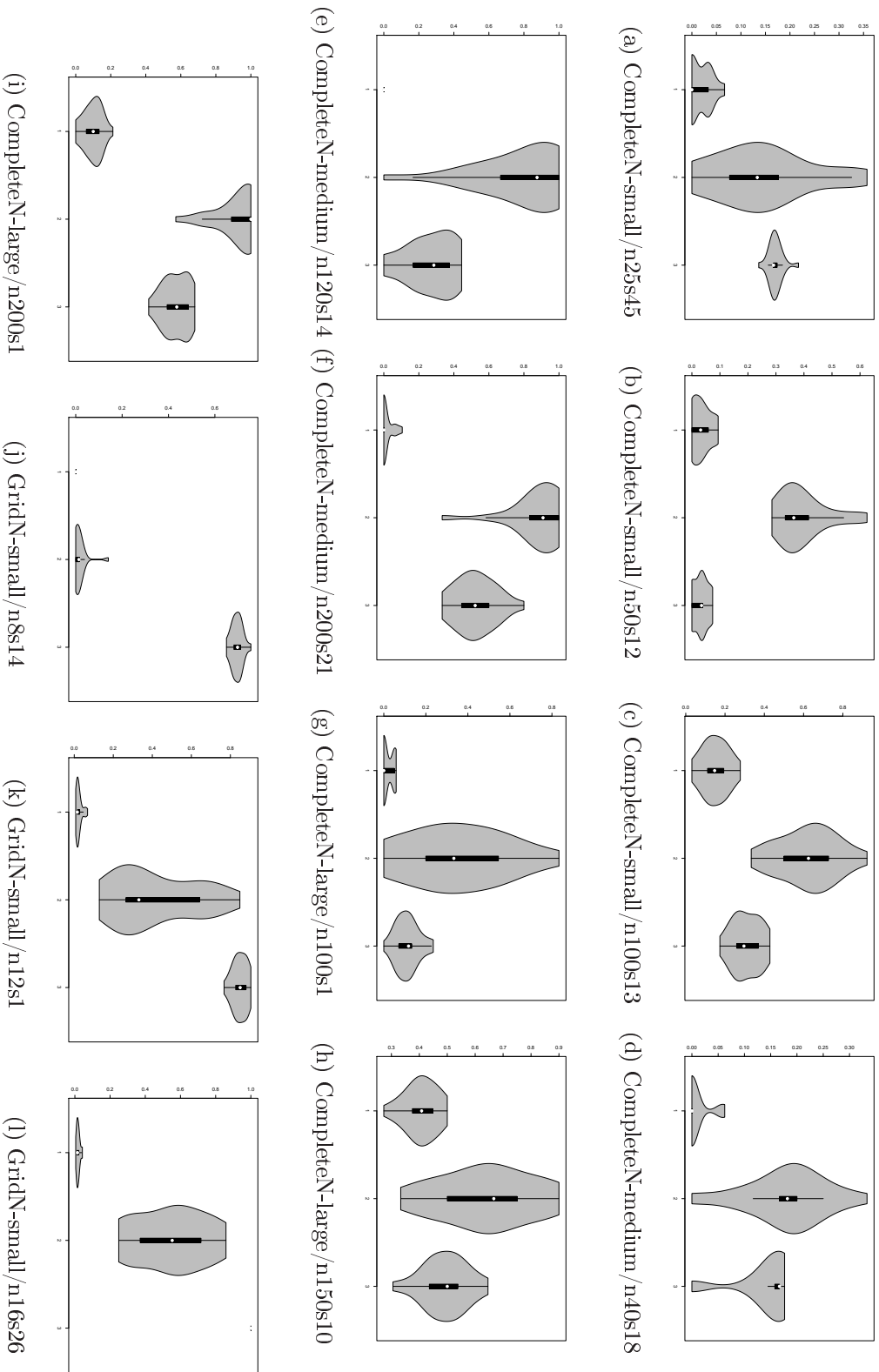
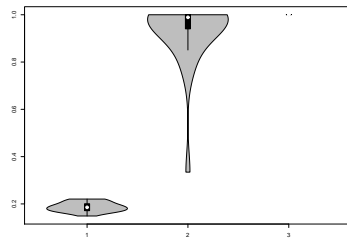
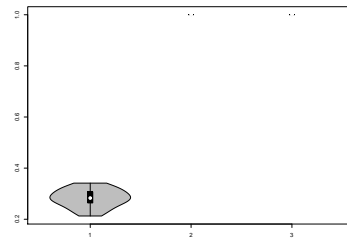


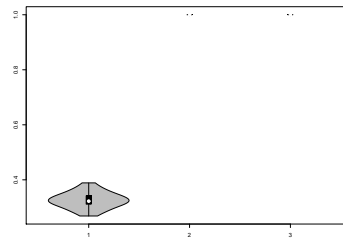
Figura 21: Violin plots para a taxa de erro da comparação entre a versão R&B-FRd, o NSGA-II e o HACO.



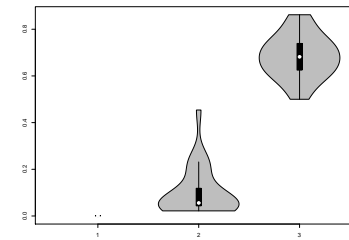
(a) GridN-medium/n22s1



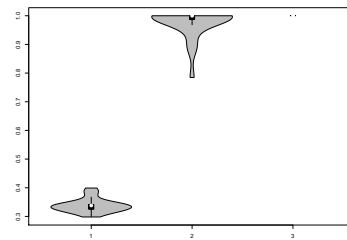
(b) GridN-medium/n31s40



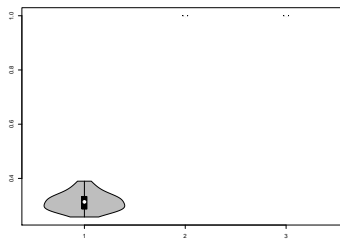
(c) GridN-medium/n35s2



(d) GridN-large/n11s41



(e) GridN-large/n22s42



(f) GridN-large/n30s43

Figura 22: Violin plots para a taxa de erro da comparação entre a versão R&B-FRd, o NSGA-II e o HACO.

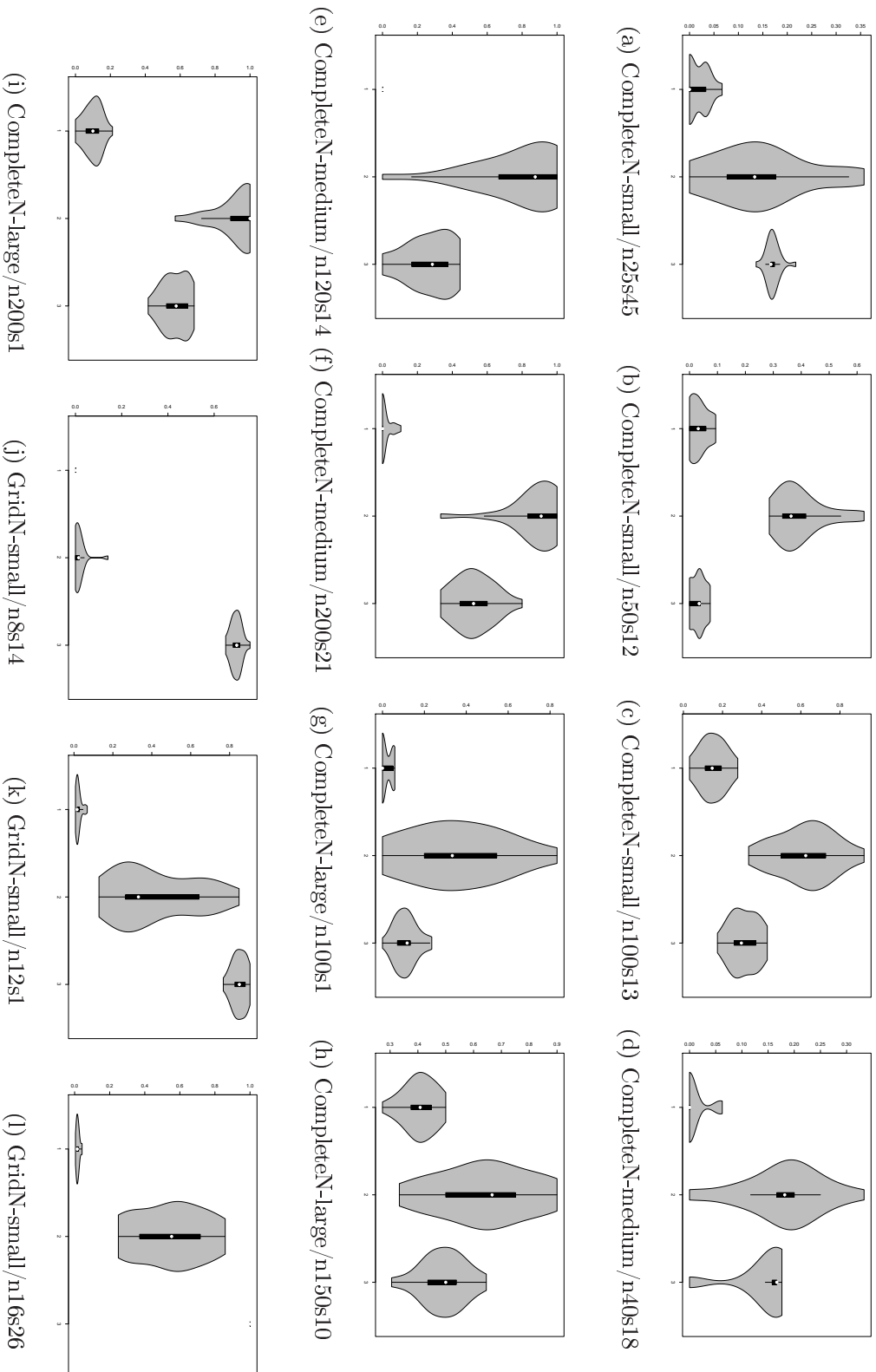
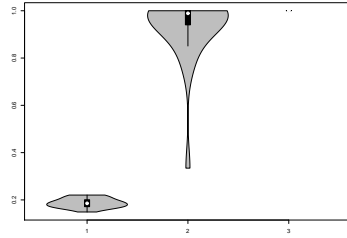
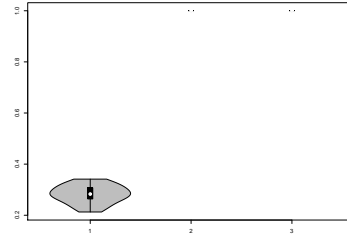


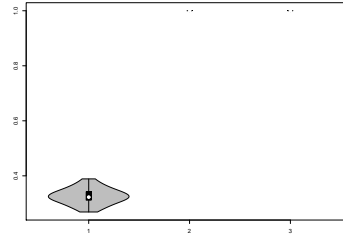
Figura 23: Violin plots para a métrica C_z da comparação entre a versão R&B-FRd, o NSGA-II e o HACO.



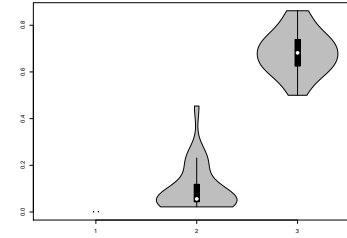
(a) GridN-medium/n22s1



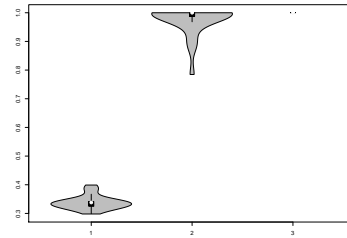
(b) GridN-medium/n31s40



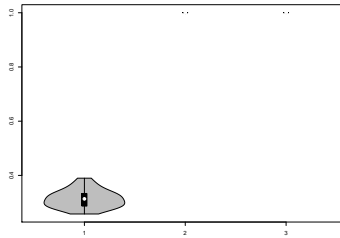
(c) GridN-medium/n35s2



(d) GridN-large/n11s41



(e) GridN-large/n22s42



(f) GridN-large/n30s43

Figura 24: Violin plots para a métrica \mathcal{C}_\prec da comparação entre a versão R&B-FRd, o NSGA-II e o HACO.