

An Investigation into Many-Objective Optimization Problems: A Case Study of the Dial-a-Ride Problem

Renan J. dos S. Viana, Flávio V. Cruzeiro Martins, and Elizabeth F. Wanner

Abstract—Multi-objective optimization problems with more than three objectives are commonly referred to as many-objective optimization problems. Usually, this class of problem brings new and complex challenges to the current optimization methods, mainly maintaining the right balance between convergence and diversity. During the last years, various approaches have been proposed to solve many-objective problems. However, most existing experimental comparative studies are restricted to continuous problems. Few studies have encompassed the most recently proposed state-of-the-art approaches and made an experimental comparison applied to combinatorial optimization problems. Aiming to fill this gap, this paper presents a comparative analysis with eight algorithms covering various categories to solve a many-objective Dial-a-Ride problem. The results show that different observations can be made about the algorithms' behavior when using different test sets. Also, algorithms originally proposed to deal with problems with up to three objectives have overcome recently proposed ones.

Index Terms—Many-objective optimization, dial-a-ride problem, multi-objective evolutionary algorithms, combinatorial optimization.

I. INTRODUÇÃO

Transporte Reativo a Demanda, do inglês *Demand Responsive Transport* (DRT), é um termo utilizado para nomear serviços de transporte flexíveis que operam sob demanda por meio de uma frota de veículos (ônibus, vans, carros, etc.), a qual é programada para transportar passageiros de acordo com suas necessidades [1]. Para solicitar um serviço DRT, os usuários formulam requisições nas quais determinam os locais e horários de coleta e entrega desejados. Geralmente, um serviço DRT é operacionalizado de forma compartilhada. De acordo com [2], esse tipo de serviço é considerado uma forma intermediária de transporte, localizado entre os serviços de transporte convencionais providos por meio de ônibus (compartilhado e geral) e os táxis (individual e personalizado).

Este trabalho aborda o problema de otimização do planejamento tanto do roteamento quanto da programação de um serviço DRT. Em geral, este problema é formulado como um *Dial-a-Ride Problem* (DARP) multiobjetivo. O DARP é um problema clássico de roteirização e programação de veículos no qual a carga transportada consiste em passageiros, geralmente provenientes de diferentes requisições e podendo ser transportados simultaneamente pelo mesmo veículo. O objetivo é projetar rotas e definir horários de atendimento para um conjunto de usuários, os quais formulam requisições de

coleta e entrega entre origens e destinos desejados. Usualmente, o processo de planejamento das rotas busca atender um conjunto de requisições de transporte minimizando os custos operacionais, tradicionalmente associados ao número de rotas e à distância total da viagem. Além disso, precisa satisfazer um conjunto de restrições, tais como: número de veículos disponíveis, capacidade dos veículos, janelas de tempo, tempo máximo de viagem, entre outras [3].

Após [4], [5], o DARP têm recebido crescente atenção da comunidade científica, sendo a grande maioria pesquisas relacionadas às aplicações reais do problema. Revisões extensivas da literatura do DARP e de algumas de suas principais variantes foram realizadas por [6], [7]. Nestes artigos são apresentados os principais estudos, conjuntos de instâncias de teste, metodologias de solução e áreas de aplicação, assim como sugestões de direções para futuras pesquisas.

Entre os diversos trabalhos que abordam o DARP, a grande maioria otimiza, individualmente, funções objetivo relacionadas aos custos operacionais, tais como: minimização do custo total de roteamento [4], minimização da distância total percorrida pela frota de veículos [8], minimização do número de veículos necessários [9], maximização da eficiência de utilização dos veículos [10], maximização da taxa de ocupação de passageiros [11], maximização do lucro total [12] e otimização da carga de trabalho da equipe [13].

A maior parte dos trabalhos disponíveis na literatura, aborda uma formulação mono-objetivo do DARP em que a perspectiva da empresa (custos operacionais) é tratada como a função a ser otimizada e a qualidade do serviço é tratada como restrições do problema [6]. Como exemplos de restrições relacionadas à qualidade do serviço podemos citar: estabelecer janelas de tempo, tempo máximo de viagem do usuário, tempo máximo de espera, entre outros. Versões multiobjetivo, do inglês *Multi-objective Optimization Problem* (MOP) ou com muitos objetivos, do inglês *Many-Objective Optimization Problems* (MaOPs), em que a perspectiva da empresa e a qualidade de serviço são otimizadas simultaneamente são apresentados em [14], [15], [11], [16], [17], [9], [18], [19] e [20]. Nestes trabalhos, o problema é resolvido usando (i) uma escalarização por meio de um conjunto de pesos definido pelo usuário, formando um único objetivo; (ii) uma abordagem lexicográfica na qual os objetivos são otimizados seguindo uma estrutura hierárquica definida com base na importância de cada objetivo; e (iii) uma abordagem multiobjetivo baseada em relação de dominância, as quais utilizam relações de preferência para guiar a busca. Podemos ainda citar exemplos que resolvem o problema com muitos objetivos utilizando técnicas de redução de dimensionalidade por árvore de agregação e clusterização e aplicando algoritmos MOPs clássicos para

Renan J. S. Viana é doutorando do Programa de Pós Graduação em Modelagem Matemática e Computacional do CEFET-MG, Belo Horizonte, Brasil. (renanjviana@gmail.com)

Flávio V. C. Martins e Elizabeth F. Wanner são do Departamento de Computação do CEFET-MG. ({flavio, ewanner}@cefetmg.br)

resolver o problema resultante [21], [22], [23] e usando um algoritmo memético baseado em decomposição especializado para tratar o problema na forma original [24].

De acordo com [25], lidar com MaOPs têm promovido, ao longo da última década, novos e complexos desafios para os pesquisadores da área, dado que otimizá-los é significativamente mais difícil que otimizar MOPs tradicionais. Estes desafios são amplamente discutidos, analisados e explorados em problemas contínuos. Diversos algoritmos foram propostos para resolver tais problemas e os comportamentos de tais algoritmos nos mais diversos problemas são analisados na literatura [26]. Entretanto, problemas combinatórios não têm recebido tanta atenção quanto os problemas contínuos. Na tentativa de preencher esta lacuna, neste trabalho é realizado uma análise comparativa entre oito algoritmos populares utilizados na literatura, e por meio de indicadores de desempenho multiobjetivo avaliamos o desempenho/eficiência destes algoritmos quando aplicados a um DARP formulado como um MaOP.

O restante deste artigo está organizado da seguinte forma. A Seção II descreve o problema abordado. Seção III apresenta os principais conceitos sobre otimização multiobjetivo. Seção IV descreve brevemente os algoritmos utilizados neste estudo, assim como os componentes evolutivos adotados. Em seguida, Seção V apresenta os experimentos computacionais e resultados obtidos. Por fim, a Seção VI expõe as conclusões.

II. DEFINIÇÃO DO PROBLEMA

Introduzido em [17] e posteriormente em [9], o serviço DRT pode ser formulado como um *Dial-a-Ride Problem* (DARP) multiobjetivo, o qual otimiza várias funções objetivo, enquanto satisfaz um conjunto de restrições. Baseada na formulação do DARP mono-objetivo proposta em [5], em [27] os autores apresentaram uma formulação matemática para o problema abordado. Originalmente, três funções objetivo são simultaneamente otimizadas. Busca-se minimizar o número de rotas necessárias, a duração total das rotas e o atraso total no desembarque dos passageiros. Entretanto, com o objetivo de analisar a escalabilidade de alguns dos principais algoritmos evolutivos multiobjetivo, do inglês *Multiobjective Evolutionary Algorithms* (MOEAs) existentes, dois novos objetivos foram adicionados: a distância total das rotas e o tempo total de espera dos usuários para o embarque. O problema abordado é uma generalização do clássico DARP e é, portanto, classificado como NP-Difícil.

Neste serviço DRT, todos os n usuários devem ser atendidos por alguma das rotas programadas. Além disso, cada usuário formula uma requisição de transporte r , na qual define o ponto de embarque r^+ , o ponto de desembarque r^- , o número de assentos necessários q_r e o horário de embarque h_{r^+} desejado. A duração da janela de tempo na qual deve ocorrer o embarque em r^+ é proporcional à duração da viagem entre r^+ e r^- (t_{r^+,r^-}), definida como $w_{r^+} = k_w \cdot t_{r^+,r^-}$, sendo k_w um coeficiente que indica a porcentagem da duração da viagem alocada para a janela de tempo. O horário teórico de chegada em r^- é dado pela soma entre h_{r^+} e a duração da viagem entre r^+ e r^- , resultando em $h_{r^-} = h_{r^+} + t_{r^+,r^-}$. O horário máximo de chegada é definido como $h'_{r^-} = h_{r^+} + (k_r \cdot t_{r^+,r^-})$,

sendo k_r um coeficiente de relaxação. Tanto k_w quanto k_r são parâmetros de entrada.

Para ser classificada como factível, uma rota deve satisfazer as seguintes condições: (i) A rota deve ser iniciada e finalizada no ponto associado ao depósito; (ii) Para cada restrição de transporte r , os pontos r^+ e r^- devem pertencer a uma mesma rota e o ponto r^- deve ser alocado após o ponto r^+ . Desta forma, o desembarque do passageiro deve acontecer após o embarque do mesmo; (iii) Em nenhum momento a ocupação do veículo pode exceder a capacidade máxima estabelecida; (iv) O embarque do passageiro no ponto r^+ deve ocorrer dentro do intervalo estabelecido $[h_{r^+}, (h_{r^+} + w_{r^+})]$ e o seu desembarque no ponto r^- deve ocorrer no intervalo $[h_{r^-}, h'_{r^-}]$.

Para este serviço DRT, busca-se planejamentos otimizados para o roteamento (definição das rotas) e para a programação (definição dos horários de atendimento) por meio da minimização dos custos operacionais e maximização da qualidade do serviço prestado.

III. OTIMIZAÇÃO MULTI OBJETIVO

Como o próprio nome sugere, problemas de otimização multiobjetivo (MOPs) envolvem a otimização simultânea de vários objetivos frequentemente conflitantes. Objetivos são considerados conflitantes quando a melhora em um objetivo somente é possível por meio da piora em pelo menos algum dos demais.

Formalmente, um MOP pode ser definido por meio de um conjunto de m funções objetivo $\mathbf{f} = [f_1 \ f_2 \ \dots \ f_m]$, um conjunto X de soluções factíveis no espaço de decisão e um conjunto Z composto por pontos possíveis no espaço de objetivo $Z = f(X)$. Uma solução $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]$ é um vetor de n variáveis de decisão, e para cada solução $\mathbf{x} \in X$, existe um ponto $\mathbf{z} \in Z$ associado, denotado por $\mathbf{f} : X \rightarrow Z$ com $\mathbf{z} = \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x}) \ \dots \ f_m(\mathbf{x})]$ [28].

Geralmente, os algoritmos de otimização multiobjetivo utilizam a relação de dominância Pareto [29] para comparar soluções. Dado um MOP com m funções objetivo, uma solução \mathbf{x} domina uma outra solução \mathbf{x}' ($\mathbf{x} \prec \mathbf{x}'$), se e somente se, as seguintes condições forem satisfeitas: (i) a solução \mathbf{x} não é pior que a solução \mathbf{x}' em todos os objetivos; (ii) a solução \mathbf{x} é estritamente melhor que a solução \mathbf{x}' em pelo menos um dos objetivos considerados.

Se qualquer uma destas duas condições for violada, a solução \mathbf{x} não domina a solução \mathbf{x}' . Além disso, se \mathbf{x} não domina \mathbf{x}' , não implica que \mathbf{x}' domina \mathbf{x} . Uma solução $\mathbf{x}^* \in X$ é definida Pareto-ótimo se não existe uma solução diferente $\mathbf{x} \in X$ em que $\mathbf{x} \prec \mathbf{x}^*$. O conjunto formado por todas soluções Pareto-ótimo é intitulado conjunto Pareto-ótimo e a imagem deste conjunto no espaço de objetivos é denominada fronteira Pareto-ótimo [28].

Geralmente, em vez de encontrar uma única solução ótima global ou uma boa aproximação da mesma, como na otimização de problemas com um único objetivo (problemas mono-objetivo), ao otimizar MOPs, busca-se uma boa aproximação da fronteira Pareto-ótimo com uma grande diversidade.

Algoritmos Evolutivos, do inglês *Evolutionary Algorithms* (EAs) são métodos de otimização que baseiam-se nos princípios da evolução natural para compor procedimentos de busca e otimização [30]. De forma iterativa, simulam a evolução de um conjunto de indivíduos (soluções) utilizando o conceito de competição, o qual garante a sobrevivência dos indivíduos mais aptos. Baseados nos princípios da biologia, os EAs são compostos por mecanismos de seleção, mutação e combinação. Quando projetados para tratar MOPs, EAs são denominados algoritmos evolutivos multiobjetivo (MOEAs). Segundo [31], EAs se destacam na resolução de problemas com múltiplas funções objetivo devido ao fato de manterem um conjunto de indivíduos (população), e ao longo do processo evolutivo combiná-los para a obtenção de novos indivíduos, possivelmente com novas informações genéticas. Portanto, EAs são capazes de obterem vários membros do conjunto Pareto-ótimo em apenas uma única execução, ao contrário de métodos convencionais, que necessitam de várias execuções independentes para atingirem resultados similares.

IV. ABORDAGENS HEURÍSTICAS

Esta seção apresenta os componentes evolutivos gerais utilizados para compor os MOEAs avaliados. Tanto o procedimento de geração da população inicial quanto os operadores de variação empregados foram introduzidos em [27] e posteriormente utilizados em [20]. Em seguida, os oito MOEAs utilizados para resolver o DARP multiobjetivo são brevemente apresentados.

A. Representação da Solução

Neste trabalho, uma solução candidata é representada por meio de um vetor de rotas. Computacionalmente, um cromossomo (indivíduo) é codificado como um vetor bi-dimensional. Basicamente, cada rota é representada por meio de um vetor composto por identificadores de requisições de transporte. Cada requisição de transporte está associada a um par de pontos, os quais representam os locais de embarque e de desembarque desejados. A primeira e a segunda ocorrência do identificador da requisição em uma rota representam os pontos de embarque e desembarque, respectivamente. Porém, para facilitar a identificação da ação (embarque ou desembarque) que irá ocorrer em cada parada do veículo, a ocorrência do identificador que representa o embarque será precedida pelo sinal positivo (+) e a ocorrência que representa o desembarque será precedida pelo sinal negativo (-). Note que os identificadores em conjunto com os sinais correspondentes são armazenados no vetor (rota) na ordem em que os atendimentos deverão ser executados pelo motorista do veículo. A Figura 1 apresenta um exemplo de solução para uma instância de teste fictícia composta por 8 requisições de transporte, as quais foram distribuídas em 4 rotas.

Um indivíduo é avaliado utilizando a função de avaliação. Esta função retorna um vetor de objetivo com cinco valores, $f = [f_1 f_2 f_3 f_4 f_5]$, o qual representa a solução no espaço de objetivo. Uma vez que tanto o procedimento de geração da população inicial, quanto os operadores de variação, garantem a factibilidade dos indivíduos durante todo o processo de otimização, não foram utilizadas estratégias de penalização.

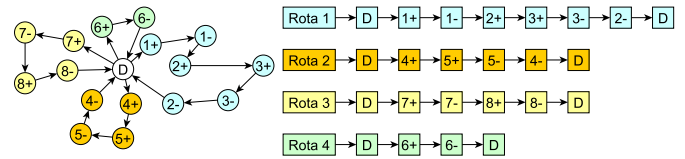


Fig. 1. Esquema de representação de uma solução.

B. Geração da População Inicial

O procedimento de geração da população inicial empregado considera o horário desejado de coleta (h_{i+}) informado pelos usuários e o tempo necessário para um veículo mover-se entre o depósito e o ponto de embarque ($t_{D,i+}$) para definir as requisições de transporte mais urgentes. Desta forma, as requisições de transporte que demandam atendimentos que exigem maior antecedência da partida dos veículos serão priorizados. Para criação de uma solução, o procedimento utilizado define a ordem das requisições de forma puramente gulosa e realiza a atribuição às rotas de forma aleatória.

C. Operador de Recombinação (Crossover)

Projetado com base no operador clássico de recombinação com um ponto de corte, o operador de recombinação utiliza dois indivíduos pais, selecionados da população corrente, com o objetivo de gerar dois indivíduos filhos C_1 e C_2 . O procedimento proposto é composto por cinco etapas, as quais foram exemplificadas na Figura 2.

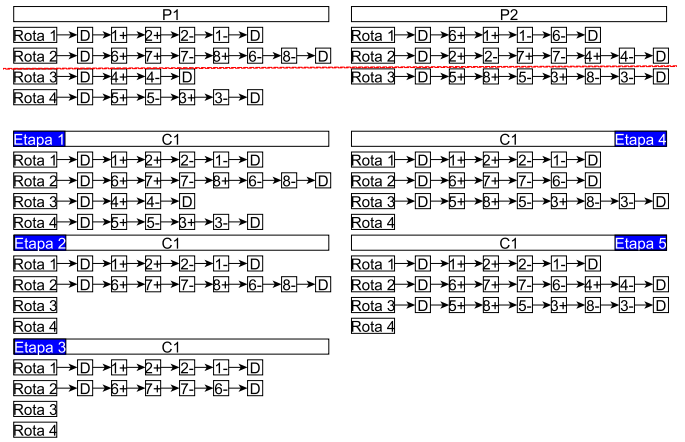


Fig. 2. Esquema de aplicação do operador de recombinação.

Dado dois indivíduos pais P_1 e P_2 , para a geração de um filho, C_1 , é realizado uma cópia do indivíduo P_1 (primeira etapa). Para modificar C_1 , seleciona-se um ponto de corte l entre 1 e $\min(|P_1|, |P_2|)$. Posteriormente, remove-se todas as rotas no vetor de rotas de C_1 que estão indexadas após l (segunda etapa). Com o objetivo de evitar dados duplicados (duplo atendimento), remove-se de C_1 todas as requisições de transporte em P_2 que estão atribuídas à alguma das rotas indexadas após l e que já estão alocadas à alguma das rotas de C_1 (terceira etapa). Em seguida, todas as rotas em P_2 que estão indexadas após l são incluídas em C_1 (quarta etapa). Por fim, todas as requisições de transporte que ainda não

foram atribuídas a alguma das rotas de C_1 , são inseridas na rota e na posição que possibilita o menor custo de inserção possível (quinta etapa). Por menor custo de inserção possível entende-se menor acréscimo na duração total das rotas. O indivíduo C_2 é produzido da mesma maneira invertendo os papéis de P_1 e P_2 . Para controlar a frequência de execução dos cruzamentos, é definida uma probabilidade Pr intitulada taxa de recombinação.

D. Operador de Mutação

Com o objetivo de introduzir diversidade à população, o operador de mutação utilizado seleciona de forma aleatória duas rotas diferentes do indivíduo corrente ($r1$ e $r2$). Em seguida, seleciona aleatoriamente e remove de $r1$ uma requisição req . O processo de reinserção de req no indivíduo é iniciado em $r2$. Caso não seja possível inserir a requisição de forma que mantenha a factibilidade do indivíduo, é realizada uma nova tentativa na rota subsequente. Este processo continua enquanto não for possível realizar uma inserção factível e houver rotas. Se além de $r1$, nenhuma outra rota do indivíduo corrente for capaz de atender a requisição de forma factível, uma nova rota (veículo) é alocada para efetuar o atendimento. O procedimento que analisa a viabilidade da inserção inicia as tentativas sempre ao final da rota, afetando poucos atendimentos e, conseqüentemente, resultando em um menor acréscimo no atraso total dos desembarques. Para controlar a frequência em que as mutações ocorrem, é definida uma probabilidade Pm intitulada taxa de mutação.

E. MOEAs Utilizados

Oito MOEAs foram selecionados de modo que as principais classes de algoritmos evolutivos fossem contempladas. Para representar as abordagens baseadas na relação de dominância Pareto, foram utilizados os algoritmos *Non-Dominated Sorting Genetic Algorithm II* (NSGA-II) [32] e *Strength Pareto Evolutionary Algorithm 2* (SPEA2) [33]. O algoritmo *Multiobjective Evolutionary Algorithm Based on Decomposition* (MOEA/D) [34] e sua variante *MOEA/D with global replacement* (MOEA/D-GR) [35] foram selecionados para representar as abordagens de decomposição baseadas em agregação. O algoritmo *Indicator-based Multi-objective Evolutionary Algorithm* (IBEA) [36] representa as abordagens baseadas em indicador de qualidade. Os algoritmos *Non-dominated Sorting Genetic Algorithm III* (NSGA-III) [37] e *Strength Pareto Evolutionary Algorithm Based on Reference Direction* (SPEA/R) [38] representam as abordagens de decomposição baseadas em conjuntos de referência. Por fim, o algoritmo *Two Arch Algorithm 2* (Two_Arch2) [39] representa as abordagens baseadas em dois arquivos. Para maiores detalhes sobre o funcionamento dos algoritmos usados, as referências originais devem ser consultadas.

V. EXPERIMENTOS COMPUTACIONAIS

Nesta seção será apresentada uma análise de desempenho, a qual considera oito MOEAs projetados com base em diferentes estratégias. Aplicada à uma formulação do DARP com

cinco objetivos, esta análise busca avaliar comportamentos e comparar o desempenho dos oito MOEAs quando aplicados a um MaOP combinatório.

Os testes computacionais foram executados em um computador com processador Intel Xeon E5-2630 v4, com 2.2 GHz de frequência, com 10 cores, e com 32GB de memória RAM, executando o sistema operacional *CentOS 6.6*. Além disso, os MOEAs utilizados foram codificados em C++.

Para este estudo foram utilizados dois conjuntos de instâncias de teste com diferentes características, os quais foram introduzidos em [17] e, posteriormente, utilizados em [9]. Ambos conjuntos são compostos por 10 instâncias de teste, com 100 usuários (requisições de transporte) em cada. O primeiro conjunto, intitulado “*Rnd100*”, é composto por instâncias de teste que possuem uma distribuição quase homogênea de usuários (janelas de tempo conflitantes). Neste conjunto, as requisições de transporte foram geradas aleatoriamente. Já o segundo conjunto utilizado, nomeado “*Gravit100*”, contém instâncias de teste com uma distribuição heterogênea de usuários. Neste conjunto, as requisições de transporte foram geradas utilizando um modelo geográfico de pessoas ou de fluxos de mercadorias.

Dado que a formulação original do problema abordado considera somente o tempo de viagem ($t_{i,j}$) de cada aresta $(i,j) \in A$, tanto em [17] quanto em [9], não foram definidos valores relacionados à distância entre vértices ($d_{i,j}$). Portanto, para este estudo, embora não corresponda à realidade, foi estabelecido que o valor associado à distância entre cada par de vértices é igual ao tempo de viagem correspondente ($t_{i,j} = d_{i,j}$). Note que para problemas do mundo real, o tempo de viagem entre dois vértices provavelmente não estará relacionado à distância na forma aqui estabelecida. Entretanto, valores associados à distância foram adicionados ao problema teste original para que fosse possível a utilização do mesmo no processo de otimização da formulação DARP com muitos objetivos.

A. Indicadores de Qualidade e Análise de Desempenho

Para mensurar a qualidade dos conjuntos de aproximação gerados durante os experimentos computacionais, o protocolo proposto em [40] foi seguido. Inicialmente, para cada instância de teste γ , é gerado o conjunto Z_γ^{all} a partir da união de todos os conjuntos de aproximação obtidos pelos MOEAs avaliados. Em seguida, considerando todos os pontos contidos em Z_γ^{all} , são definidos os pontos $z^{min} = [z_1^{min} \dots z_m^{min}]$ e $z^{max} = [z_1^{max} \dots z_m^{max}]$, nos quais z_k^{min} e z_k^{max} representam os limites inferior e superior do k -ésimo objetivo otimizado, respectivamente. Por fim, cada ponto $z \in Z_\gamma^{all}$ é normalizado por meio da seguinte equação:

$$z_k = \frac{z_k - z_k^{min}}{z_k^{max} - z_k^{min}}, \forall k \in \{1, \dots, m\}. \quad (1)$$

Após a execução do procedimento de normalização, todos os valores de função objetivo de cada ponto em Z_γ^{all} irá variar dentro do intervalo $[0, 1]$. Por fim, para cada instância de teste γ , é gerado um conjunto de referência Z_γ^* contendo todos os pontos não-dominados do conjunto Z_γ^{all} correspondente.

Proposto em [41], o indicador Hipervolume (I_{HV}) calcula o hiper-volume da porção do espaço de objetivo que é fracamente dominada por um conjunto de aproximação A . Quanto maior o valor de I_{HV} , melhor é a qualidade das soluções em A . Portanto, I_{HV} deve ser maximizado. Para o cálculo do I_{HV} é necessário definir um ponto de referência que seja pelo menos fracamente dominado por todos os pontos de Z_γ^{all} [40]. Dado que os pontos de Z_γ^{all} são normalizados, o ponto (1.1, 1.1, 1.1, 1.1, 1.1) foi utilizado como ponto de referência. O indicador I_{HV} é estritamente monotônico em relação à dominância Pareto. Além disso, o I_{HV} é capaz de quantificar informações tanto de convergência quanto de diversidade do conjunto avaliado.

Proposto em [42], o *Binary Additive ϵ -Indicator* ($I_{\epsilon+}$) é um indicador de desempenho que mensura convergência. Baseado no conceito de ϵ -dominância, $I_{\epsilon+}(A, B)$ calcula o fator mínimo ϵ necessário para qualquer vetor de objetivo do conjunto B seja ϵ -dominado por pelo menos um vetor de objetivo do conjunto A . Dado que o conjunto Pareto-ótimo do problema em questão não é conhecido para nenhuma das instâncias de teste utilizadas, neste estudo foi aplicado a variante unária desta métrica [43]. Desta forma, em cada instância γ de teste, um conjunto de aproximação A é avaliado utilizando o conjunto referência Z_γ^* composto por todas soluções não-dominadas obtidas ao longo de todos experimentos computacionais realizados. O conjunto Z_γ^* foi utilizado como uma aproximação ao conjunto Pareto-ótimo para avaliar o desempenho dos algoritmos avaliados. Quanto menor o valor de $I_{\epsilon+}(A, Z_\gamma^*)$, melhor é a qualidade de A .

Com base nos valores de I_{HV} e $I_{\epsilon+}$, análises estatísticas foram conduzidas utilizando um procedimento estatístico não paramétrico uma vez que a hipótese de normalidade dos dados foi rejeitada em todos os casos. Conforme proposto em [44], para cada instância de teste, uma comparação par-a-par entre algoritmos é realizada aplicando o teste de *Wilcoxon-Mann-Whitney* (*Wilcoxon rank-sum test*). A hipótese nula estabelece que os valores de indicadores de qualidade obtidos pelos dois algoritmos avaliados não diferem em localização, sendo a diferença nos resultados puramente ocasionada ao acaso. A hipótese alternativa afirma que estes valores diferem significativamente em localização. Os testes de hipóteses foram realizados usando um nível de significância de 95%. Desta forma, suponha que para cada instância de teste, n algoritmos $A_1 \dots A_n$ são comparados segundo valores de um indicador de qualidade qualquer. Para cada $i \in \{1, \dots, n\}$ e $j \in \{1, \dots, n\} \setminus \{i\}$, δ_{ij} assumirá o valor 1, se o algoritmo A_j supera significativamente o algoritmo A_i com base no teste de *Wilcoxon-Mann-Whitney*. Caso contrário, δ_{ij} assume 0. Desta forma, a pontuação de desempenho para o algoritmo A_i é computada da seguinte maneira:

$$P(A_i) = \sum_{j \in \{1, \dots, n\} \setminus \{i\}} \delta_{ij}, \quad (2)$$

na qual $P(A_i)$ representa quantos algoritmos superam A_i em uma determinada instância de teste. Quanto menor este valor, melhor é o algoritmo avaliado. $P(A_i) = 0$ significa que nenhum dos demais algoritmos testados superou estatisticamente

seus resultados. Já para $P(A_i) = 1$ significa que apenas 1 algoritmo superou seus resultados, e assim por diante. Para o caso em que não há diferença estatística entre os algoritmos nenhum deles recebe pontuação.

B. Ajuste dos Parâmetros

Com o objetivo de prover equidade à análise comparativa, configurações de parâmetros associadas à um MOEA que influenciam diretamente no consumo computacional, como tamanho da população e critério de parada foram estabelecidos iguais para todos os algoritmos avaliados. Devido a natureza estocástica dos MOEAs, 31 execuções independentes foram realizadas para cada algoritmo em cada instância de teste.

Para geração do conjunto de pontos de referência ou de direções de referência, elemento essencial para as abordagens de decomposição, foi utilizado a abordagem sistemática proposta em [45]. Para sua utilização é necessário definir quantas divisões p serão utilizadas e o número de objetivos m considerados. Conforme mostrado em [37], caso tenha $p \geq m$, não serão gerados pontos intermediários no hiperplano normalizado. Portanto, dado que o MaOP abordado possui 5 objetivos e foi estabelecido 6 divisões, 210 pontos serão gerados. Deste modo, para os algoritmos MOEA/D e MOEA/D-GR foram utilizadas populações de tamanho $N = 210$. Dado que os demais MOEAs avaliados adotam o operador de seleção por torneio para seleção dos indivíduos pais, para estes foram utilizadas populações de tamanho igual ao menor múltiplo de quatro maior que 210, logo $N = 212$. Além disso, para cada MOEA, todas as execuções foram encerradas após 100 mil avaliações de função objetivo.

Parâmetros associados aos operadores genéticos, como as taxas de recombinação e de mutação foram calibrados por meio de experimentos computacionais com o intuito de identificar as condições nas quais as abordagens de busca avaliadas apresentam melhor desempenho. Com base em experimentos computacionais preliminares, as taxas de recombinação $P_r = 1.0$ e de mutação $P_m = 0.2$ permitiram que todos os MOEAs avaliados alcançassem seus melhores desempenhos. Portanto, estes valores foram utilizados em todos os experimentos computacionais posteriores.

Para os parâmetros específicos de cada MOEA foram utilizados os valores definidos nos estudos em que foram introduzidos. Abaixo são especificados tais parâmetros e os valores correspondentes:

- **MOEA/D e MOEA/D-GR:** a abordagem *Tchebychev* foi selecionada como método de escalarização, e o tamanho da vizinhança foi definido como $N \times 0.1$;
- **IBEA $\epsilon+$:** o parâmetro k utilizado no cálculo da aptidão foi definido como 0.05;
- **SPEA2:** tamanho do arquivo definido como 100;
- **Two_ARCH2:** tamanho de CA definido como 100 e o p utilizado na norma L_p definido como $1/M$.

Por fim, alguns parâmetros associados ao problema otimizado e seus correspondentes valores são definidos:

- frota de veículos homogênea com a capacidade máxima de quatro pessoas, além do motorista;

- baseado em [9], a duração de serviço nos pontos de atendimento foi definida como zero ($d_i = 0, \forall i \in V^+ \cup V^-$);
- baseado em [9], as constantes utilizadas para os cálculos das janelas de tempo e das relaxações foram definidas 0.2 e 1.5, respectivamente.

C. Resultados e Análises Experimentais

As Tabelas I e II apresentam os valores médio dos indicadores I_{HV} e I_{ϵ^+} , respectivamente, obtidos em 31 execuções de cada MOEA avaliado para as instâncias de teste do conjunto “*Gravit100*”. As Tabelas III e IV apresentam os valores médio dos indicadores I_{HV} e I_{ϵ^+} , respectivamente, obtidos em 31 execuções de cada MOEA avaliado para as instâncias de teste do conjunto “*Rnd100*”. O valor entre parêntese representa a classificação obtida pelo algoritmo baseado em sua pontuação de desempenho (Equação 2). Quanto menor $P(A_i)$, melhor a classificação atribuída ao algoritmo. Algoritmos com o mesmo $P(A_i)$ recebem a mesma classificação, o que seria considerando um “empate” segundo esse índice. O(s) algoritmo(s) melhores classificados estão destacados em negrito.

De acordo com os resultados apresentados nas Tabelas I e II, os algoritmos Two_ARCH2 e IBEA obtiveram os melhores resultados com relação aos indicadores (I_{HV} e I_{ϵ^+}), portanto obtiveram os melhores conjuntos de aproximação em termos de convergência e diversidade. Com destaque para o Two_ARCH2, que para algumas instâncias de teste obteve melhor classificação em relação ao IBEA. Note que o processo de busca desempenhado pelos algoritmos Two_ARCH2 (em uma de suas populações) e IBEA são baseados no indicador I_{ϵ^+} , o que de certa forma proporciona alguma vantagem a estes algoritmos em relação aos demais quando utilizado o mesmo indicador para comparação de desempenho. Logo em seguida aparecem os dois representantes da estratégia de decomposição baseada em agregação (MOEA/D e MOEA/D-GR), o que provavelmente significa que a distribuição dos vetores de peso gerados foi adequada à geometria da fronteira Pareto do problema. Dentre todos os MOEAs avaliados, o NSGA-III obteve desempenho intermediário para este conjunto de teste. Os algoritmos SPEA-R, NSGA-II e SPEA2 apresentaram as piores performances. Em relação aos dois últimos, o desempenho ruim pode ser explicado pela incapacidade das abordagens baseadas na relação de dominância Pareto em lidar com formulações com alto número de objetivos.

Conforme os resultados apresentados nas Tabelas III e IV, para o conjunto de instâncias “*Rnd100*”, os algoritmos Two_ARCH2 e IBEA novamente obtiveram as melhores classificações para ambos os indicadores de qualidade. Novamente, com destaque para o algoritmo Two_ARCH2, que obteve a melhor classificação em todas as instâncias do conjunto. No geral, os algoritmos MOEA/D, NSGA-II e NSGA-III obtiveram desempenhos inferiores somente ao Two_ARCH2 e IBEA. Considerando o indicador I_{ϵ^+} , o algoritmo MOEA/D-GR apresentou uma queda de desempenho. Em contrapartida, o algoritmo SPEA2 obteve uma significativa melhora, destacando-se pela capacidade de convergência. Por fim, para ambos os indicadores, o SPEA-R apresentou o pior desempenho dentre os demais algoritmos.

VI. CONCLUSÃO

Neste trabalho foi apresentada uma análise comparativa aplicada à uma formulação com muitos objetivos do problema combinatório *Dial-a-Ride Problem*. Esta análise foi realizada com o objetivo de comparar o desempenho de oito MOEAs, projetados com base em diferentes estratégias, ao serem aplicados à um complexo problema combinatório. Os resultados obtidos sustentaram um fato já evidenciado na literatura (majoritariamente para problemas contínuos) que abordagens baseadas na relação de dominância Pareto apresentam dificuldades quando lidam com um alto número de objetivos. Além disso, os resultados mostram que para uma mesma formulação do problema, diferentes observações podem ser feitas sobre o comportamento dos algoritmos quando aplicados a conjuntos de teste com características diferentes. Portanto, além do número de objetivos, outras propriedades são capazes de influenciar o desempenho dos MOEAs. Além disso, o ótimo desempenho dos algoritmos Two_ARCH2 e IBEA evidencia a importância de se utilizar indicadores de qualidade (com baixo custo computacional) para condução do processo de busca substituindo ou combinado à tradicional relação de dominância Pareto. Por fim, embora o algoritmo IBEA seja originalmente proposto para lidar com problemas de otimização multiobjetivo com até três objetivos, o mesmo superou, com grande margem, alguns MOEAs projetados recentemente para tratar MaOPs. Apesar de guiar sua busca por meio do indicador I_{ϵ^+} , a qual foi utilizada como critério de comparação na análise realizada, o mesmo obteve bons resultados também em termos do indicador I_{HV} . É importante destacar que o bom desempenho do algoritmo IBEA ao lidar com MaOps também foi evidenciado em outros trabalhos recentes [46], [47].

TABELA I

RESULTADOS OBTIDOS PELOS OITO MOEAS PARA AS INSTÂNCIAS DE TESTE DO CONJUNTO “Gravit100” UTILIZANDO O I_{HV} .

Instância	IBEA	MOEA/D	MOEA/D-GR	NSGA-II	NSGA-III	SPEA2	SPEA-R	Two_ARCH2
Gravit100_0	0,9565 (2)	0,9028 (3)	0,8831 (4)	0,7991 (7)	0,8478 (5)	0,7285 (8)	0,813 (6)	0,9763 (1)
Gravit100_1	1,0017 (2)	0,9293 (3)	0,932 (3)	0,7777 (7)	0,8719 (5)	0,7142 (8)	0,8214 (6)	1,0229 (1)
Gravit100_2	0,9538 (1)	0,8957 (3)	0,8942 (3)	0,7784 (7)	0,8488 (5)	0,7311 (8)	0,7951 (6)	0,9458 (1)
Gravit100_3	0,979 (2)	0,9263 (3)	0,9279 (3)	0,8332 (7)	0,8636 (5)	0,765 (8)	0,8556 (5)	1,0171 (1)
Gravit100_4	1,0689 (1)	0,9825 (3)	0,9727 (3)	0,886 (7)	0,9443 (5)	0,8172 (8)	0,9142 (6)	1,0711 (1)
Gravit100_5	1,0066 (2)	0,9398 (3)	0,9398 (3)	0,8297 (7)	0,8972 (5)	0,791 (8)	0,8461 (6)	1,0316 (1)
Gravit100_6	0,9514 (1)	0,8835 (4)	0,8947 (3)	0,781 (6)	0,8329 (5)	0,7256 (8)	0,7942 (6)	0,9535 (1)
Gravit100_7	1,0215 (1)	0,9788 (3)	0,9588 (4)	0,869 (7)	0,9204 (5)	0,8078 (8)	0,8939 (6)	1,0329 (1)
Gravit100_8	1,0866 (1)	1,0262 (3)	1,0196 (3)	0,9024 (7)	0,9662 (5)	0,8433 (8)	0,9528 (6)	1,0976 (1)
Gravit100_9	1,0466 (2)	0,9667 (3)	0,9755 (3)	0,8441 (7)	0,932 (5)	0,7681 (8)	0,875 (6)	1,0677 (1)

TABELA II

RESULTADOS OBTIDOS PELOS OITO MOEAS PARA AS INSTÂNCIAS DE TESTE DO CONJUNTO “Gravit100” UTILIZANDO O $I_{\epsilon+}$.

Instância	IBEA	MOEA/D	MOEA/D-GR	NSGA-II	NSGA-III	SPEA2	SPEA-R	Two_ARCH2
Gravit100_0	0,1123 (1)	0,1317 (3)	0,1411 (4)	0,1892 (6)	0,1504 (5)	0,2072 (7)	0,2195 (8)	0,1119 (1)
Gravit100_1	0,0938 (2)	0,1428 (4)	0,1364 (3)	0,2028 (6)	0,1484 (4)	0,2152 (7)	0,2133 (7)	0,0881 (1)
Gravit100_2	0,1056 (1)	0,1389 (3)	0,1385 (3)	0,1679 (6)	0,145 (5)	0,1846 (7)	0,2126 (8)	0,1154 (2)
Gravit100_3	0,1059 (1)	0,136 (3)	0,1337 (3)	0,1916 (6)	0,1496 (5)	0,1961 (6)	0,2162 (8)	0,1032 (1)
Gravit100_4	0,1024 (1)	0,1446 (3)	0,1534 (4)	0,1847 (6)	0,1581 (4)	0,1981 (7)	0,208 (8)	0,1032 (1)
Gravit100_5	0,0957 (2)	0,1369 (3)	0,135 (3)	0,1761 (6)	0,1538 (5)	0,1927 (7)	0,2221 (8)	0,0864 (1)
Gravit100_6	0,1043 (1)	0,1471 (3)	0,1413 (3)	0,1748 (6)	0,1566 (5)	0,205 (7)	0,2102 (7)	0,1009 (1)
Gravit100_7	0,0953 (1)	0,1212 (3)	0,1229 (3)	0,1667 (6)	0,1371 (5)	0,1797 (7)	0,1887 (7)	0,095 (1)
Gravit100_8	0,095 (1)	0,1359 (3)	0,1303 (3)	0,1823 (6)	0,1556 (5)	0,2027 (8)	0,1948 (7)	0,0989 (1)
Gravit100_9	0,0994 (1)	0,1376 (3)	0,1352 (3)	0,1861 (6)	0,1429 (3)	0,2072 (7)	0,199 (7)	0,0942 (1)

TABELA III

RESULTADOS OBTIDOS PELOS OITO MOEAS PARA AS INSTÂNCIAS DE TESTE DO CONJUNTO “Rnd100” UTILIZANDO O I_{HV} .

Instância	IBEA	MOEA/D	MOEA/D-GR	NSGA-II	NSGA-III	SPEA2	SPEA-R	Two_ARCH2
Rnd100_0	0,8786 (2)	0,7991 (3)	0,7885 (4)	0,7837 (4)	0,7836 (4)	0,7787 (7)	0,7307 (8)	0,907 (1)
Rnd100_1	0,8464 (2)	0,7547 (3)	0,7436 (5)	0,7262 (6)	0,7553 (3)	0,7184 (7)	0,7015 (8)	0,8648 (1)
Rnd100_2	0,8428 (2)	0,7614 (3)	0,7505 (4)	0,7394 (6)	0,7541 (4)	0,721 (7)	0,6775 (8)	0,8616 (1)
Rnd100_3	0,8277 (2)	0,7492 (3)	0,7427 (4)	0,7141 (6)	0,7392 (4)	0,7101 (6)	0,6623 (8)	0,8472 (1)
Rnd100_4	0,888 (2)	0,8293 (4)	0,8211 (7)	0,8285 (4)	0,8247 (6)	0,8364 (3)	0,7611 (8)	0,9063 (1)
Rnd100_5	0,851 (2)	0,7813 (3)	0,7728 (4)	0,7736 (4)	0,7649 (6)	0,7577 (6)	0,7026 (8)	0,8684 (1)
Rnd100_6	0,8149 (2)	0,7584 (3)	0,7404 (4)	0,6935 (7)	0,7387 (4)	0,7249 (6)	0,6874 (7)	0,8381 (1)
Rnd100_7	0,8085 (2)	0,7504 (3)	0,7404 (4)	0,7021 (6)	0,7172 (5)	0,7044 (6)	0,6756 (8)	0,8266 (1)
Rnd100_8	0,8495 (2)	0,7667 (3)	0,749 (4)	0,7278 (6)	0,7366 (5)	0,724 (6)	0,696 (8)	0,8726 (1)
Rnd100_9	0,8901 (2)	0,828 (3)	0,8166 (5)	0,8223 (4)	0,8096 (6)	0,8014 (7)	0,7205 (8)	0,8995 (1)

TABELA IV

RESULTADOS OBTIDOS PELOS OITO MOEAS PARA AS INSTÂNCIAS DE TESTE DO CONJUNTO “Rnd100” UTILIZANDO O $I_{\epsilon+}$.

Instância	IBEA	MOEA/D	MOEA/D-GR	NSGA-II	NSGA-III	SPEA2	SPEA-R	Two_ARCH2
Rnd100_0	0,0458 (2)	0,1304 (4)	0,1429 (7)	0,1364 (4)	0,1309 (4)	0,1197 (3)	0,2441 (8)	0,0422 (1)
Rnd100_1	0,0441 (2)	0,1424 (5)	0,1532 (6)	0,1585 (6)	0,1318 (3)	0,1293 (3)	0,2421 (8)	0,038 (1)
Rnd100_2	0,0463 (2)	0,1118 (4)	0,1226 (5)	0,137 (7)	0,1077 (3)	0,1211 (5)	0,2331 (8)	0,0418 (1)
Rnd100_3	0,0428 (2)	0,1224 (3)	0,1502 (7)	0,1206 (3)	0,1351 (6)	0,1173 (3)	0,2435 (8)	0,037 (1)
Rnd100_4	0,0417 (2)	0,13 (5)	0,1495 (7)	0,1083 (4)	0,1306 (5)	0,0889 (3)	0,2311 (8)	0,0296 (1)
Rnd100_5	0,0474 (1)	0,1108 (3)	0,1313 (6)	0,1253 (6)	0,1121 (3)	0,1177 (3)	0,2187 (8)	0,0496 (1)
Rnd100_6	0,0611 (2)	0,1323 (5)	0,1427 (6)	0,1443 (6)	0,1205 (3)	0,1158 (3)	0,2373 (8)	0,0431 (1)
Rnd100_7	0,0494 (2)	0,1333 (3)	0,1408 (6)	0,1523 (7)	0,1301 (3)	0,1277 (3)	0,229 (8)	0,0469 (1)
Rnd100_8	0,0466 (1)	0,1282 (4)	0,135 (6)	0,1511 (7)	0,1183 (3)	0,1278 (4)	0,2364 (8)	0,0448 (1)
Rnd100_9	0,0396 (2)	0,1115 (6)	0,1296 (7)	0,1062 (3)	0,1076 (3)	0,1073 (3)	0,2393 (8)	0,0366 (1)

ACKNOWLEDGMENT

Os autores agradecem a FAPEMIG, o CNPq e a CAPES pelo financiamento da pesquisa.

REFERÊNCIAS

[1] J. Mageean and J. D. Nelson, “The evaluation of demand responsive transport services in europe,” *Journal of Transport Geography*, vol. 11, no. 4, pp. 255–270, 2003.

[2] G. Ambrosino, J. Nelson, and M. Romanazzo, *Demand responsive transport services: Towards the flexible mobility agency*. ENEA, Italian National Agency for New Technologies, Energy and the Environment, 2004.

[3] J.-F. Cordeau and G. Laporte, “The dial-a-ride problem: models and algorithms,” *Annals of Operations Research*, vol. 153, no. 1, pp. 29–46, 2007.

[4] —, “A tabu search heuristic for the static multi-vehicle dial-a-ride problem,” *Transportation Research Part B: Methodological*, vol. 37, no. 6, pp. 579–594, 2003.

[5] J.-F. Cordeau, “A branch-and-cut algorithm for the dial-a-ride problem,” *Operations Research*, vol. 54, no. 3, pp. 573–586, 2006.

[6] Y. Molenbruch, K. Braekers, and A. Caris, “Typology and literature review for dial-a-ride problems,” *Annals of Operations Research*, vol. 259, no. 1–2, pp. 295–325, 2017.

[7] S. C. Ho, W. Szeto, Y.-H. Kuo, J. M. Leung, M. Petering, and T. W. Tou, “A survey of dial-a-ride problems: Literature review and recent developments,” *Transportation Research Part B: Methodological*, 2018.

[8] U. Ritzinger, J. Puchinger, and R. F. Hartl, “Dynamic programming

- based metaheuristics for the dial-a-ride problem,” *Annals of Operations Research*, vol. 236, no. 2, pp. 341–358, 2016.
- [9] R. Chevrier, A. Liefvooghe, L. Jourdan, and C. Dhaenens, “Solving a dial-a-ride problem with a hybrid evolutionary multi-objective approach: Application to demand responsive transport,” *Applied Soft Computing*, vol. 12, no. 4, pp. 1247–1258, 2012.
- [10] B. Rekiek, A. Delchambre, and H. A. Saleh, “Handicapped person transportation: An application of the grouping genetic algorithm,” *Engineering Applications of Artificial Intelligence*, vol. 19, no. 5, pp. 511–520, 2006.
- [11] T. Garaix, C. Artigues, D. Feillet, and D. Josselin, “Optimization of occupancy rate in dial-a-ride problems via linear fractional column generation,” *Computers & Operations Research*, vol. 38, no. 10, pp. 1435–1442, 2011.
- [12] S. N. Parragh, J. Pinho de Sousa, and B. Almada-Lobo, “The dial-a-ride problem with split requests and profits,” *Transportation Science*, vol. 49, no. 2, pp. 311–334, 2014.
- [13] A. Lim, Z. Zhang, and H. Qin, “Pickup and delivery service with manpower planning in hong kong public hospitals,” *Transportation Science*, vol. 51, no. 2, pp. 688–705, 2016.
- [14] R. M. Jorgensen, J. Larsen, and K. B. Bergvinsdottir, “Solving the dial-a-ride problem using genetic algorithms,” *Journal of the operational research society*, vol. 58, no. 10, pp. 1321–1331, 2007.
- [15] D. Kirchler and R. W. Calvo, “A granular tabu search algorithm for the dial-a-ride problem,” *Transportation Research Part B: Methodological*, vol. 56, pp. 120–135, 2013.
- [16] M. Schilde, K. F. Doerner, and R. F. Hartl, “Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem,” *European journal of operational research*, vol. 238, no. 1, pp. 18–30, 2014.
- [17] R. Chevrier, A. Liefvooghe, L. Jourdan, and C. Dhaenens, “On optimizing a demand responsive transport with an evolutionary multi-objective approach,” in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 2010, pp. 575–580.
- [18] J. Paquette, J.-F. Cordeau, G. Laporte, and M. M. Pascoal, “Combining multicriteria analysis and tabu search for dial-a-ride problems,” *Transportation Research Part B: Methodological*, vol. 52, pp. 1–16, 2013.
- [19] Y. Molenbruch, K. Braekers, A. Caris, and G. V. Berghe, “Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation,” *Computers & Operations Research*, vol. 77, pp. 58–71, 2017.
- [20] R. J. S. Viana, A. G. Santos, F. V. C. Martins, and E. F. Wanner, “Optimization of a demand responsive transport service using multi-objective evolutionary algorithms,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2064–2067. [Online]. Available: <https://doi.org/10.1145/3319619.3328528>
- [21] R. S. Mendes, D. S. Miranda, E. F. Wanner, J. F. M. Sarubbi, and F. V. C. Martins, “Multiobjective approach to the vehicle routing problem with demand responsive transport,” in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 3761–3768.
- [22] R. Mendes, E. Wanner, F. Martins, and J. Sarubbi, “Dimensionality reduction approach for many-objective vehicle routing problem with demand responsive transport,” in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2017, pp. 438–452.
- [23] R. S. Mendes, V. Lush, E. F. Wanner, F. V. Martins, J. F. Sarubbi, and K. Deb, “Online clustering reduction based on parametric and non-parametric correlation for a many-objective vehicle routing problem with demand responsive transport,” *Expert Systems with Applications*, vol. 170, p. 114467, 2021. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417420311180>
- [24] A. A. Anwar and I. Younas, “Optimization of many objective pickup and delivery problem with delay time of vehicle using memetic decomposition based evolutionary algorithm,” *International Journal on Artificial Intelligence Tools*, vol. 29, no. 01, 2020.
- [25] H. Zhao, C. Zhang, J. Ning, B. Zhang, P. Sun, and Y. Feng, “A comparative study of the evolutionary many-objective algorithms,” *Progress in Artificial Intelligence*, vol. 8, no. 1, pp. 15–43, 2019.
- [26] B. Li, J. Li, K. Tang, and X. Yao, “Many-objective evolutionary algorithms: A survey,” *ACM Comput. Surv.*, vol. 48, no. 1, Sep. 2015. [Online]. Available: <https://doi.org/10.1145/2792984>
- [27] R. J. D. S. VIANA, “Abordagens heurísticas para otimização de um serviço de transporte reativo a demanda,” Master’s thesis, Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Viçosa, 2016.
- [28] C. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.
- [29] V. Pareto, “Cours deconomie, vol. I and II, f,” *Rouge, Lausanne*, 1896.
- [30] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.
- [31] A. Abraham and L. Jain, *Evolutionary multiobjective optimization*. Springer, 2005.
- [32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.
- [33] E. Zitzler, M. Laumanns, L. Thiele, E. Zitzler, E. Zitzler, L. Thiele, and L. Thiele, “SPEA2: Improving the strength pareto evolutionary algorithm,” Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK), Tech. Rep., 2001.
- [34] Q. Zhang and H. Li, “Moea/d: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [35] Z. Wang, Q. Zhang, M. Gong, and A. Zhou, “A replacement strategy for balancing convergence and diversity in moea/d,” in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 2132–2139.
- [36] E. Zitzler and S. Künzli, “Indicator-based selection in multiobjective search,” in *Parallel Problem Solving from Nature-PPSN VIII*. Springer, 2004, pp. 832–842.
- [37] K. DEB and H. JAIN, “An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints,” *IEEE transactions on evolutionary computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [38] S. Jiang and S. Yang, “A strength pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 329–346, 2017.
- [39] H. Wang, L. Jiao, and X. Yao, “Two_arch2: An improved two-archive algorithm for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 524–541, 2015.
- [40] J. D. Knowles, L. Thiele, and E. Zitzler, “A tutorial on the performance assessment of stochastic multiobjective optimizers,” *TIK-Report*, vol. 214, 2006.
- [41] E. Zitzler and L. Thiele, “Multiobjective optimization using evolutionary algorithms - a comparative case study,” in *Parallel problem solving from nature - PPSN V*. Springer, 1998, pp. 292–301.
- [42] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, “Performance assessment of multiobjective optimizers: An analysis and review,” *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 2, pp. 117–132, 2003.
- [43] C. M. Fonseca, J. D. Knowles, L. Thiele, and E. Zitzler, “A tutorial on the performance assessment of stochastic multiobjective optimizers,” in *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, vol. 216, 2005, p. 240.
- [44] J. Bader and E. Zitzler, “Hype: An algorithm for fast hypervolume-based many-objective optimization,” *Evolutionary computation*, vol. 19, no. 1, pp. 45–76, 2011.
- [45] I. Das and J. E. Dennis, “Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems,” *SIAM journal on optimization*, vol. 8, no. 3, pp. 631–657, 1998.
- [46] R. Tanabe, H. Ishibuchi, and A. Oyama, “Benchmarking multi- and many-objective evolutionary algorithms under two optimization scenarios,” *IEEE Access*, vol. 5, pp. 19 597–19 619, 2017.
- [47] L. C. Bezerra, M. López-Ibáñez, and T. Stützle, “A large-scale experimental evaluation of high-performing multi-and many-objective evolutionary algorithms,” *Evolutionary computation*, vol. 26, no. 4, pp. 621–656, 2018.



Renan J. dos S. Viana possui graduação em Sistemas de Informação pela Universidade Federal de Ouro Preto (2013), mestrado em Ciência da Computação pela Universidade Federal de Viçosa (2016). É doutorando no Programa de Pós-Graduação em Modelagem Matemática Computacional do Centro Federal de Educação Tecnológica de Minas Gerais. Tem interesse na área de otimização, nos seguintes temas: algoritmos evolutivos, otimização multiobjetivo e otimização combinatória.



Flávio V. Cruzeiro Martins possui graduação em Ciência da Computação (2007), mestrado (2009) e doutorado (2012) em Engenharia Elétrica ambos pela Universidade Federal de Minas Gerais. É professor no Centro Federal de Educação Tecnológica de Minas Gerais e Bolsista de Produtividade Desenvolvimento Tecnológico e Extensão Inovadora do CNPq, Nível 2. Tem experiência na área de otimização, atuando principalmente nos seguintes temas: algoritmos evolutivos, otimização mono e multiobjetivo e otimização combinatória.



Elizabeth F. Wanner possui mestrado em Matemática (2002), doutorado em Engenharia Elétrica (2006) ambos pela Universidade Federal de Minas Gerais e pós-doutorado em Ciência da Computação pela Universidade do Algarve, Portugal (2009). É professor do Departamento de Computação do Centro Federal de Educação Tecnológica de Minas Gerais. Tem experiência na área de Matemática Aplicada, atuando principalmente nos seguintes temas: algoritmos populacionais, aproximações quadráticas, restrições de igualdade, medidas de desempenho e

otimização mono e multiobjetivo.