

# ABORDAGEM MULTIOBJETIVO PARA UM PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM SERVIÇOS DE COLETA E ENTREGA

**André L. Maravilha, Luciana Assis, Alessandro Vivas**

Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM)  
{msilva.andreluiz,lupassis,alessandro.vivas}@gmail.com

**Jaime A. Ramírez**

Universidade Federal de Minas Gerais (UFMG)  
jramirez@ufmg.br

**Resumo** – Este artigo apresenta uma abordagem multiobjetivo para um problema de roteamento de veículos com entregas obrigatórias e coletas opcionais. Este problema tem como objetivo definir rotas com custo mínimo e, ao mesmo tempo, atender o maior número de demandas de coleta. O algoritmo proposto para solucionar o problema é uma adaptação do *Iterated Local Search* que retorna um conjunto de soluções não dominadas. Para avaliar a eficiência deste algoritmo, foram utilizadas instâncias do Problema de Roteamento de Veículos com Coletas e Entregas Simultâneas mono-objetivo. Os resultados mostraram que o algoritmo proposto apresenta um bom desempenho quanto ao número de soluções retornadas, extensão e distribuição das soluções Pareto. A qualidade dos resultados foi analisada também considerando os objetivos individualmente. Neste caso, os resultados também foram satisfatórios e se aproximam dos resultados da literatura.

**Palavras-chave** – Logística & Transportes, Metaheurísticas, Otimização Combinatória.

**Abstract** – This paper presents an approach to a multiobjective vehicle routing problem with deliveries and selective pickups. This problem aims to define routes with minimum cost and, at the same time, attend as many demands of collection. The proposed algorithm is an adaptation of Iterated Local Search algorithm, which returns not only a solution but a set of nondominated solutions. To evaluate the efficiency of this algorithm, we used instances of the mono-objective vehicle routing problem with simultaneous pickup and delivery. The results indicate that the proposed algorithm presents a good performs on the number of solutions returned, extent and uniform distribution of Pareto solutions. Another form used in this study to evaluate the quality of the solution was to analyze the objectives individually. In this case, the results were also satisfactory and closer to the ones available in the literature.

**Keywords** – Logistics & Transport, Metaheuristics, Combinatorial Optimization.

## 1 INTRODUÇÃO

O estudo de roteamento de veículos tem um papel importante em indústrias que trabalham com um fluxo intenso de materiais. Em regiões com grande área territorial, o custo com o transporte pode impactar diretamente no custo final do produto. Segundo a Associação Brasileira de Logística [1], os custos de transporte correspondem a 60% dos custos logísticos das empresas. Os custos logísticos no Brasil atingiram, em 2008, um valor equivalente a 11,6% do Produto Interno Bruto (PIB), ou seja, R\$ 349,0 bilhões. O item de maior representatividade foi o transporte, com 6,9% do PIB (R\$207,0 bilhões).

Em diversos casos, além de realizar a entrega de produtos, é necessário realizar a coleta dos resíduos gerados após o consumo destes produtos para reutilização ou armazenagem em local adequado. O estudo deste fluxo de materiais está inserido no contexto da logística reversa que planeja toda a atividade de entrega dos produtos, e também o caminho inverso, o retorno dos produtos à central de distribuição.

Na otimização combinatória, um problema básico de logística reversa é o Problema de Roteamento de Veículos com Coletas e Entregas Simultâneas (VRPSPD, do inglês *Vehicle Routing Problem with Simultaneous Pickup and Delivery*). O objetivo deste problema, proposto por Min [2], é minimizar o custo do transporte, atendendo às demandas de coleta e entrega simultaneamente. Cada cliente deve ser visitado uma única vez e por um único veículo, onde se tem disponível uma frota de veículos homogênea e limitada. É fácil perceber que, se nenhum consumidor necessita do serviço de coleta, o VRPSPD pode ser reduzido ao Problema de Roteamento de Veículos (VRP, do inglês *Vehicle Routing Problem*), mostrando que é um problema NP-Difícil.

Desde que foi proposto, vários são os estudos disponíveis na literatura para o VRPSPD. Dell'Amico [3] apresenta uma modelagem matemática para o problema e faz uso da técnica *Branch-and-Price* para solucioná-la. Para a fase de avaliação (*pricing*) são apresentadas duas técnicas, uma utilizando programação dinâmica e outra com relaxação do espaço de estados.

Diversos autores apresentam algoritmos baseados nas metaheurísticas *Tabu Search* [4–7], *Greedy Random Adaptative Search Procedure* (GRASP) [8], *Iterated Local Search* (ILS) [9, 10] e *Variable Neighborhood Descent* (VND) [8–10], pois estas apresentam bons resultados na solução do VRPSPD. Nestes trabalhos, além do uso destas metaheurísticas, são apresentadas, também, diversas técnicas geração de soluções de partida dos algoritmos.

Subramanian et al [10] apresenta uma abordagem paralela para solucionar o VRPSPD, na qual é utilizado a metaheurística *Iterated Local Search* (ILS) em conjunto com um procedimento *Variable Neighborhood Descent* com ordenação aleatória (RVND). Até então, este detém os melhores resultados para grande parte das instâncias (problemas teste) do VRPSPD.

Abordagens bio-inspiradas vem se mostrando como uma boa opção na resolução de problemas de otimização combinatória. Nos trabalhos [11, 12] são propostos algoritmos baseados em colônia de formigas para solução do VRPSPD. Estas abordagens apresentam resultados de boa qualidade, principalmente por permitirem a exploração de diversas soluções, entretanto, tem um maior custo computacional. Maravilha et al. [13] também apresenta um algoritmo baseado em colônia de formigas, além de um algoritmo baseado no sistema imunológico. É feita a comparação entre as duas abordagens, sendo que o algoritmo imunológico apresenta melhores resultados.

Apesar das inúmeras aplicações práticas do VRPSPD, existem situações onde a realização da coleta não é obrigatória, permitindo a escolha de quais coletas realizar. Um exemplo desta situação pode ser observado nas indústrias de agrotóxicos. No Brasil, os usuários tem um prazo de um ano para devolver as embalagens vazias para os estabelecimentos comerciais. As empresas produtoras e comercializadoras de agrotóxicos tem um prazo de um ano para coleta e destinação das embalagens. Com isso, não é necessário atender a todas demandas imediatamente, já que tem-se o prazo de um ano para efetuar a coleta, podendo selecionar quais delas realizar.

Devido a esta característica, pode-se reduzir ainda mais o custo da rota, pois um veículo ao não realizar uma determinada coleta, possuirá mais espaço livre para a carga de produtos de entrega, permitindo assim, atender mais clientes. Apesar da realização das demandas de coleta não ser obrigatória, ela é desejada, pois assim evita uma futura visita ao consumidor para recolher o produto que ainda não foi coletado. Portanto, deve-se maximizar o número de coletas realizadas.

Um problema de otimização combinatória com estas características é o Problema de Roteamento de Veículos com Entregas e Coletas Seletivas (SVRPDSP, do inglês *Single Vehicle Routing Problem with Deliveries and Selective Pickups*) [14]. Nele, cada consumidor possui demandas de entrega e coleta, e para cada demanda de coleta existe um lucro associado pela sua realização. O objetivo é definir um percurso que atenda a todas as demandas de entrega e, também, realize as coletas que forem consideradas lucrativas, minimizando uma função objetivo denominada custo líquido (custo total da rota subtraído do lucro total obtido através das coletas realizadas). Gribkovskaia et al. [14] apresentam a formulação matemática do SVRPDSP e propõem um algoritmo baseado na metaheurística *Tabu Search*, além de algumas heurísticas construtivas e de melhoria.

Sural e Bookbinder [15] tratam de um outro problema semelhante ao SVRPDSP, denominado Problema de Roteamento de Veículos com *Backhauls* Irrestrito (SVRPUB, do inglês *Single Vehicle Routing Problem with Unrestricted Backhauls*). A diferença entre eles é que no SVRPUB nem todos os consumidores apresentam demandas de coleta. Para resolvê-lo, os autores propõem algumas técnicas de melhoria para modelagem matemática e, em seguida, aplicam o software CPLEX para resolver de maneira exata o modelo proposto, permitindo encontrar a solução ótima, para problemas de tamanho médio, em tempo razoável.

Outro trabalho que também aborda o roteamento de veículos com entregas obrigatórias e coletas opcionais é [16]. Neste os autores tratam do Problema de Coleta Seletiva e Entrega (SPDP, *Selective Pickup and Delivery Problem*, onde é considerado um conjunto de pontos para aluguel de bicicletas e estas bicicletas devem ser distribuídas de acordo com a demanda de cada ponto. Neste problema a distribuição dos produtos é feita entre os pontos de consumo, não existindo um depósito, mas apenas um ponto de partida. Para solução do problema, os autores utilizam um algoritmo genético.

Os trabalhos encontrados na literatura, que abordam problemas de roteamento de veículos com coletas não obrigatórias, fazem uso de formulações mono-objetivo, onde a função objetivo é dada, geralmente, pelo custo total das rotas menos o lucro obtido pelas coletas realizadas. Mas, as características desta classe de problema possibilitam a formulação de um modelo multiobjetivo, onde podem ser consideradas duas funções objetivo: (i) a minimização do custo total da rota e (ii) a maximização do número total de coletas realizadas.

O estudo de problemas de roteamento de veículos com múltiplos objetivos vem apresentando maior destaque nos últimos anos, como pode ser observado pelo trabalho de [17] em que faz uma revisão da literatura para problemas de roteamento de veículos entrega multiobjetivo. A maior parte dos trabalhos apresentados neste estudo foram publicados a partir do ano 2000.

Em algumas situações reais é necessário visualizar diversos cenários, dados os diversos objetivos do problema, para se tomar uma decisão. Assim sendo, ao modelar problemas de maneira multiobjetivo, estes se tornam mais interessantes para aplicações práticas [18]. Apesar de serem mais atraentes para aplicações reais, o grau de complexidade é maior pois envolvem objetivos conflitantes.

O presente trabalho aborda o Problema de Roteamento de Veículos com Entrega e Coleta Seletiva de forma multiobjetivo, apresentando uma adaptação da metaheurística ILS para geração de soluções do mesmo. As demais seções estão organizadas como segue: Seção 2 são apresentadas a definição do problema e sua modelagem matemática multiobjetivo. Em seguida (Seção 3), é proposto o algoritmo ILS multiobjetivo. Na Seção 4 são apresentados os resultados computacionais obtidos pelo algoritmo, e por fim, na Seção 5 são feitas as conclusões deste trabalho.

## 2 DEFINIÇÃO DO PROBLEMA

O Problema de Roteamento de Veículos com Entrega e Coleta Seletiva Multiobjetivo (MOVRPDSP, do inglês *Multiobjective Vehicle Routing Problem with Delivery and Selective Pickup*) pode ser definido como: dado um grafo completo  $G = (V, A)$ , onde  $V = \{0, 1, 2, \dots, n\}$  é o conjunto de vértices e  $A = \{(i, j) : i, j \in V, i \neq j\}$  o conjunto de arestas. O vértice 0 representa o depósito e os demais vértices representam os consumidores. Cada aresta  $(i, j)$  tem um valor  $c_{ij} \geq 0$  associado que representa

o custo de se alcançar o vértice  $j$  a partir do vértice  $i$ . Cada consumidor  $i$  tem uma demanda  $d_i$  de entrega e uma demanda  $p_i$  de coleta. Tem-se disponível uma frota de  $\bar{k}$  veículos homogêneos de capacidade  $Q$  para realizar as demandas.

Aqui, todas as demandas de entrega devem ser realizadas, porém, as demandas de coleta são opcionais, mas desejadas. Então, o objetivo é construir rotas que minimizem o custo total de se realizar o trajeto, atendendo todas as demandas de entrega, e ao mesmo tempo, minimizar o número de coletas não realizadas.

Os veículos não podem ter sua capacidade extrapolada em nenhum momento do trajeto. Outra restrição é que um consumidor deve ser visitado uma única vez e por um único veículo, ou seja, se este consumidor tiver a demanda de coleta atendida, esta deverá ser realizada simultaneamente com a demanda de entrega. Não é permitido a realização parcial das demandas.

Os dois objetivos a serem otimizados são conflitantes, ou seja, a melhora em um provoca a piora do outro, não existindo uma solução que seja ótima para ambos os objetivos simultaneamente. Assim, não é uma tarefa trivial identificar se uma determinada solução é de melhor qualidade em relação a outra. Isso leva ao conceito de dominância de Pareto (Definição 1), sendo a solução do problema um conjunto de soluções não dominadas (Definições 2, 3).

**Definição 1 (Dominância de Pareto)** Dado um vetor objetivo  $Z(x) = (f_1(x), \dots, f_m(x))$ , com  $m$  funções objetivo para minimização. Uma solução  $x'$  domina uma outra solução  $x''$  se, e somente se, para todos os objetivos  $f_i(x') \leq f_i(x'')$  com  $i = 1, \dots, m$  e, existe pelo menos um objetivo  $f_i(x') < f_i(x'')$ . Esta relação de dominância é representada pela notação  $x' \prec x''$ .

**Definição 2 (Solução Pareto-Ótima)**  $x^*$  é uma solução Pareto-Ótima se não existe qualquer outra solução  $x$  tal que  $x \prec x^*$ , ou seja,  $x^*$  não é dominada por nenhuma outra solução.

**Definição 3 (Conjunto Pareto-Ótimo)** O conjunto  $\mathcal{X}^*$  é um conjunto Pareto-Ótimo se é composto por todas as soluções Pareto-Ótimas. O conjunto-imagem  $\mathcal{Y}^*$  associado ao conjunto Pareto-Ótimo é denominado fronteira Pareto-Ótima.

Encontrar o conjunto Pareto-Ótimo pode ser computacionalmente intratável [18]. Desta forma, a abordagem a problemas de otimização combinatória multiobjetivo é caracterizada pelo desenvolvimento de técnicas para tentar encontrar um conjunto de soluções viáveis que representam pontos próximos à fronteira Pareto-Ótima.

## 2.1 MODELAGEM MATEMÁTICA

A modelagem matemática apresentada nesta seção é uma adaptação da modelagem proposta em [4] para o Problema de Roteamento de Veículos com Coleta e Entrega Simultâneas.

Seja  $V$  o conjunto de consumidores, num total de  $n$  consumidores e  $V_0$  o conjunto de consumidores mais o depósito ( $V_0 = V \cup \{0\}$ ). Para alcançar um consumidor  $j$  a partir do consumidor  $i$ , existe um custo  $c_{ij} \geq 0$ . As demandas de coleta e entrega do consumidor  $j$  são representadas por  $p_j$  e  $d_j$ , respectivamente, com  $j = 1, 2, 3, \dots, n$ . O número de veículos disponíveis é dado por  $\bar{k}$ , sendo  $Q$  a capacidade destes. O parâmetro  $y_{ij}$  é o somatório das cargas coletadas entre o depósito e o nó  $i$  (inclusive) dirigida ao nó  $j \in V_0$ . Já  $z_{ij}$  é o somatório das demandas de entrega dos consumidores visitados após o nó  $i$  (inclusive) e dirigida ao nó  $j \in V_0$ .

Considerando ainda as seguintes variáveis:

$$x_{ij}^k = \begin{cases} 1 & , \text{ se arco } (i, j) \text{ faz parte da rota trafegada pelo veículo } k, \\ 0 & , \text{ caso contrário.} \end{cases}$$

$$pc_j = \begin{cases} 1 & , \text{ se a demanda de coleta do consumidor } j \text{ é satisfeita,} \\ 0 & , \text{ caso contrário.} \end{cases}$$

O modelo matemático pode ser expresso por:

$$\min \sum_{k=1}^{\bar{k}} \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij}^k \quad (1)$$

$$\min \sum_{j=1}^n p_j - \sum_{j=1}^n p_j pc_j \quad (2)$$

sujeito a

$$\sum_{i=0}^n \sum_{k=1}^{\bar{k}} x_{ij}^k = 1, \quad j = 1, \dots, n \quad (3)$$

$$\sum_{i=0}^n x_{ij}^k - \sum_{i=0}^n x_{ji}^k = 0, \quad j = 0, \dots, n \text{ e } k = 0, \dots, \bar{k} \quad (4)$$

$$\sum_{j=1}^n x_{0j}^k \leq 1, \quad k = 1, \dots, \bar{k} \quad (5)$$

$$\sum_{i=0}^n z_{ij} - \sum_{i=0}^n z_{ji} = d_j, \quad \forall j \neq 0 \quad (6)$$

$$y_{ij} + z_{ij} \leq Q \sum_{k=1}^{\bar{k}} x_{ij}^k, \quad i, j = 0, \dots, n \quad (7)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 0, \dots, n \quad (8)$$

$$y_{ij} \geq 0, \quad i, j = 0, \dots, n \quad (9)$$

$$z_{ij} \geq 0, \quad i, j = 0, \dots, n \quad (10)$$

A função objetivo, dada na Equação 1, visa a minimização do custo total da rota, já a Equação 2, apresenta a segunda função objetivo, que visa a minimização das coletas não realizadas.

Estas funções objetivo devem satisfazer algumas restrições. Cada ponto de demanda deve ser visitado por um único veículo (Equação 3). A Equação 4 representa a restrição de conservação do fluxo. A Equação 5 indica que  $\bar{k}$  veículos, no máximo, podem ser utilizados. A Equação 6 obriga a satisfação de todas as demandas de entrega. A Equação 7 define que as demandas devem ser transportadas nos arcos incluídos na solução e ainda impõem um limite para a carga total transportada pelo veículo. A Equação 8 representa a restrição de integralidade e as Equações 9 e 10 representam as restrições de não-negatividade para demandas de coleta e entrega, respectivamente.

### 3 O ALGORITMO ILS MULTIOBJETIVO

Diversas técnicas heurísticas podem ser encontradas na literatura para solução de problemas multiobjetivo, dentre elas, os algoritmos evolucionários, propostos em [19–22], vem se destacando.

Ao se tratar de problemas de roteamento de veículos mono-objetivo com serviço de coleta e entrega, heurísticas baseadas no ILS [9, 10] apresentam bons resultados e não necessita de ajuste de muitos parâmetros. Além disso, adaptações do ILS foram utilizadas para resolução de outros problemas de otimização combinatória multiobjetivo, apresentando bons resultados [23]. Assim, o presente trabalho propõe uma adaptação desta metaheurística, possibilitando a resolução de problemas de otimização combinatória multiobjetivo, em especial o MOVRPDS.

O algoritmo ILS é baseado na ideia de que um procedimento de busca local pode ser melhorado a partir da geração de novas soluções de partida, sendo estas novas soluções obtidas por meio de perturbações em uma solução ótima local [24]. Ele possui 4 etapas: (i) geração de uma solução inicial; (ii) busca local, que retorna uma solução melhorada; (iii) perturbação, que modifica uma solução e; (iv) critério de aceitação, que decide qual solução será utilizada na próxima iteração do algoritmo.

À adaptação proposta é dado o nome de *Multiobjective Iterated Local Search* (MOILS). Seu funcionamento é ilustrado no Algoritmo 1. O MOILS inicia seu funcionamento com a geração de soluções iniciais (linha 1). Neste trabalho, esta etapa foi realizada com a geração de duas soluções através do ILS. A primeira atendendo todas as demandas de coleta e a segunda, não atendendo a nenhuma demanda de coleta. As soluções geradas são inseridas no conjunto *Front* que armazenará as soluções não dominadas.

Em seguida, são executados *maxIter* iterações, onde uma solução em *Front* é selecionada e o bloco de repetição principal do algoritmo (linhas 6–15) é executado. Neste bloco, as etapas de perturbação (linha 7) e busca local (linha 8), semelhantes ao ILS, são aplicadas à solução escolhida. O critério de aceitação é substituído por uma função de atualização, onde tenta-se inserir a nova solução gerada no conjunto *Front* (linha 9).

A variável *cont* representa o número de iterações realizadas sem a geração de uma solução não dominada. A sua atualização é feita da seguinte forma: se a solução gerada após a perturbação e busca local for inserida em *Front* (indicada pela variável lógica *inserido*) então *cont* é reiniciado (linha 11), senão *cont* é incrementado de uma unidade (linha 13).

O parâmetro *maxCont* representa o número máximo de vezes que uma solução será explorada sem a geração de uma solução não dominada.

As subseções a seguir descrevem o funcionamento das etapas de seleção, perturbação, busca local e atualização do MOILS, empregadas neste trabalho.

#### 3.1 SELEÇÃO

O procedimento de seleção utilizado é baseado na *crowding-distance* do algoritmo NSGA-II (*Nondominated Sorting Genetic Algorithm II*) [19]. No algoritmo proposto, é atribuído, a cada solução, uma probabilidade proporcional à sua *crowding-distance*. Em seguida, é feito o sorteio de uma solução baseado nesta probabilidade.

Quanto mais distante das outras soluções, uma determinada solução estiver, maior a probabilidade desta ser escolhida. Assim é possível realizar a exploração do espaço de soluções de forma mais inteligente, onde é dado prioridade a regiões pouco exploradas, evitando a concentração da busca em determinados nichos.

O cálculo da *crowding-distance* utilizada neste trabalho é um pouco diferente do realizado no NSGA-II. As soluções presentes nos extremos da fronteira tem a *crowding-distance* igual a duas vezes a distância entre elas e a solução mais próxima. Enquanto no NSGA-II, as soluções presentes nas extremidades da fronteira recebem um valor infinito.

---

**Algoritmo 1** Pseudo-código do algoritmo MOILS

---

```
1:  $Front \leftarrow \text{gerarSoluçõesIniciais}()$ ;  
2:  $iter \leftarrow 0$ ;  
3: enquanto  $iter < \text{maxIter}$  fazer  
4:    $s' \leftarrow \text{selecionar}(Front)$ ;  
5:    $cont \leftarrow 0$ ;  
6:   enquanto  $cont < \text{maxCont}$  fazer  
7:      $s'' \leftarrow \text{perturbação}(s')$ ;  
8:      $s'' \leftarrow \text{buscaLocal}(s'')$ ;  
9:      $inserido \leftarrow \text{atualizar}(Front, s'')$ ;  
10:    se  $inserido$  então  
11:       $cont \leftarrow 0$ ;  
12:    se não  
13:       $cont \leftarrow cont + 1$ ;  
14:    fim se  
15:  fim enquanto  
16:   $iter \leftarrow iter + 1$ ;  
17: fim enquanto  
18: retornar  $Front$ ;
```

---

### 3.2 PERTURBAÇÃO

Na etapa de perturbação são utilizados mecanismos que realizam modificações em uma solução, escapando de ótimos locais, permitindo a exploração de outras áreas do espaço de busca. As modificações realizadas não podem ser muito pequenas, pois não permitiriam a exploração de novas soluções, nem muito grandes, pois assim, as buscas locais seriam realizadas a partir de pontos completamente aleatórios.

Neste trabalho, a perturbação foi realizada em duas etapas. Primeiro, é feita a alteração do status de realização da coleta de alguns consumidores selecionados aleatoriamente. Se o consumidor tiver a sua coleta realizada, então ela não será mais realizada, e vice-versa. Em seguida, um dos seguintes mecanismos de perturbação é escolhido aleatoriamente e aplicado à solução: Multiple Swap, Multiple Shift, Ejection Chain.

### 3.3 BUSCA LOCAL

A etapa de busca local realiza a exploração da vizinhança de uma solução, a fim de encontrar uma outra solução ainda melhor. Neste trabalho foi utilizado o método de refinamento VND com as estruturas de vizinhança Shift(1,0), Swap(1,1), Crossover, Or-opt, 2-opt e Exchange.

Após a execução do VND, tenta-se realizar a demanda de coleta dos consumidores que não estão sendo atendidos. A demanda de coleta de um consumidor será atendida se mantiver a solução válida. Para todas as estruturas de vizinhança foi utilizado a estratégia melhor-aprimorante.

### 3.4 ATUALIZAÇÃO

A etapa de atualização verifica se uma determinada solução  $s$  não é dominada pelas soluções presentes no conjunto  $Front$ . Se  $s$  não é dominada por nenhuma solução em  $Front$ , então ela é inserida no conjunto e as soluções por ela dominadas são removidas.

## 4 TESTES COMPUTACIONAIS

O algoritmo foi implementado em linguagem C++, com o ambiente de desenvolvimento Eclipse. Os testes foram executados em uma máquina com 2 processadores Intel(R) Xeon(R) E5506 de 2.13 GHz, com 16 GB de memória RAM e sistema operacional GNU/Linux, distribuição Ubuntu Server 10.04 de 64 bits.

Para testar a eficiência do algoritmo proposto, foi utilizado um conjunto de 14 instâncias [25], comumente encontradas na literatura para o VRSPD, e com número de consumidores variando entre 50 e 199. Antes de executar os testes com estas instâncias, foram realizados experimentos preliminares para que os parâmetros pudessem ser ajustados. Através destes experimentos, o número de iterações do algoritmo ( $\text{maxIter}$ ) foi definido como 100 e o número de iterações permitidas sem a ocorrência de soluções não dominadas ( $\text{maxCont}$ ) foi definido como 18.

### 4.1 MÉTRICAS DE DESEMPENHO

A análise da qualidade dos resultados para problemas multiobjetivo é mais complexa que para problemas com um único objetivo, pois é avaliado um conjunto de soluções ao invés de uma única solução. Portanto, é necessário aplicar algumas métricas de desempenho para analisar algumas características do conjunto de soluções.

As principais características avaliadas em otimização multiobjetivo são: (i) a proximidade da fronteira de Pareto-Ótima; (ii) a cardinalidade, que indica o número de soluções obtidas; (iii) a distribuição, isto é, se as soluções estão distribuídas de maneira uniforme ao longo da fronteira; e (iv) a extensão, ou seja, o intervalo de valores que é coberto pela fronteira para cada objetivo.

Como não se tem conhecimento da Fronteira Pareto-Ótima para o problema aqui abordado, então a primeira característica apresentada não será considerada. Para avaliação das características *iii* e *iv* são utilizadas métricas apresentadas em [26].

## 4.2 RESULTADOS OBTIDOS

Na Tabela 1 são apresentados o melhor, a média e o pior resultados para avaliação das características citadas anteriormente. A segunda coluna (Cons.) da tabela informa o número de consumidores envolvidos. Os resultados presentes nesta tabela referem-se a 30 execuções do algoritmo.

Tabela 1: Medidas de eficiência

Instância	Cons.	Cardinalidade			Extensão			Distribuição		
		Melhor	Média	Pior	Melhor	Média	Pior	Melhor	Média	Pior
CMT1X	50	13	6,47	1	104,77	35,99	0,00	10,18	4,76	1,00
CMT1Y	50	29	20,03	12	283,49	188,77	146,70	25,07	16,10	7,83
CMT2X	75	24	12,33	4	284,97	94,16	24,73	22,52	10,78	2,67
CMT2Y	75	48	35,43	27	401,77	314,42	210,97	45,96	33,58	24,54
CMT3X	100	30	19,13	12	379,18	125,98	37,69	28,14	16,84	9,82
CMT3Y	100	50	38,97	27	356,16	255,61	167,04	47,25	36,64	24,08
CMT12X	100	37	25,80	18	293,74	168,59	134,99	34,11	23,89	15,41
CMT12Y	100	55	33,43	23	339,87	207,59	130,47	53,44	31,45	21,18
CMT11X	120	71	55,07	35	454,13	339,07	214,33	69,09	52,76	32,82
CMT11Y	120	69	46,50	25	261,96	219,39	172,89	67,47	44,40	23,17
CMT4X	150	35	26,10	16	742,86	261,28	93,01	33,29	23,91	11,78
CMT4Y	150	68	52,43	39	696,62	500,17	367,03	66,45	50,51	36,58
CMT5X	199	43	29,53	18	653,73	293,05	150,16	40,44	27,60	16,47
CMT5Y	199	73	56,40	42	956,75	740,07	552,39	71,11	54,58	39,80

Para a avaliação de cardinalidade do conjunto de soluções, as instâncias com poucos consumidores apresentaram resultados razoáveis, exceto a instância CMT1X, onde a cardinalidade foi muito baixa. Entretanto, para instâncias envolvendo um maior número de consumidores, os resultados obtidos pelo MOILS se mostraram bons, apresentando uma quantidade significativa de soluções não dominadas nos conjuntos retornados.

Em relação à extensão da fronteira retornada pelo algoritmo, assim como na avaliação da cardinalidade, quanto mais complexa a instância, ou seja, quanto maior o número de consumidores envolvidos, maior é o valor retornado. Isso ocorre pois, quanto mais consumidores existirem maior o número de cenários, ou soluções, que podem ser obtidos.

A instância CMT1X, no pior resultado para a extensão, apresentou o valor 0.00. Este resultado é referente à solução que apresentou cardinalidade igual a 1 (pior caso na avaliação da cardinalidade para a instância). A extensão desta fronteira apresenta este valor pois ela é formada por apenas um único ponto no espaço de soluções.

Ao verificar a qualidade da distribuição das soluções ao longo da fronteira, percebe-se que seus resultados estão próximos ao da cardinalidade. Então, fica evidente que o MOILS consegue gerar soluções distribuídas de maneira uniforme pela fronteira, representando bem as configurações possíveis entre custo e coletas.

Esse resultado indica, também, que o MOILS consegue evitar a concentração em determinados nichos do espaço de soluções. Isto deve-se principalmente à técnica de seleção utilizada, baseada na *crowding-distance*.

A Figura 1 apresenta as fronteiras encontradas para as instâncias CMT3X e CMT5Y, respectivamente, em uma execução do algoritmo. Apesar da instância CMT3X (Figura 1(a)) apresentar um bom resultado para a qualidade da distribuição, existe um espaço considerável entre as soluções na fronteira. Isso ocorre devido à sua cardinalidade pequena em relação à extensão. Para a instância CMT5Y (Figura 1(b)), que obteve bons resultados para cardinalidade e distribuição, o gráfico apresenta uma fronteira visualmente melhor.

O tempo para execução do algoritmo também é avaliado. A Tabela 2 apresenta o tempo médio, em minutos, para sua execução.

Tabela 2: Média de tempo para execução em minutos

Instância	CMT1X	CMT1Y	CMT2X	CMT2Y	CMT3X	CMT3Y	CMT12X
Tempo (min.)	2,84	6,54	4,93	12,56	26,32	50,08	29,54
Instância	CMT12Y	CMT11X	CMT11Y	CMT4X	CMT4Y	CMT5X	CMT5Y
Tempo (min.)	36,06	131,97	86,36	66,31	150,52	122,56	299,47

Devido ao não conhecimento da fronteira Pareto-Ótima, é feita a comparação dos melhores resultados obtidos pelo MOILS com abordagens mono-objetivo. A Tabela 3 apresenta os melhores resultados conhecidos [10] quando todas as coletas são realizadas, e compara com os obtidos pelo MOILS. Ela apresenta ainda, o melhor resultado encontrado considerando apenas a minimização do custo ( $f_1$ ) e as coletas não realizadas ( $f_2$ ) associada, e também, o custo associado quando todas as coletas são realizadas.

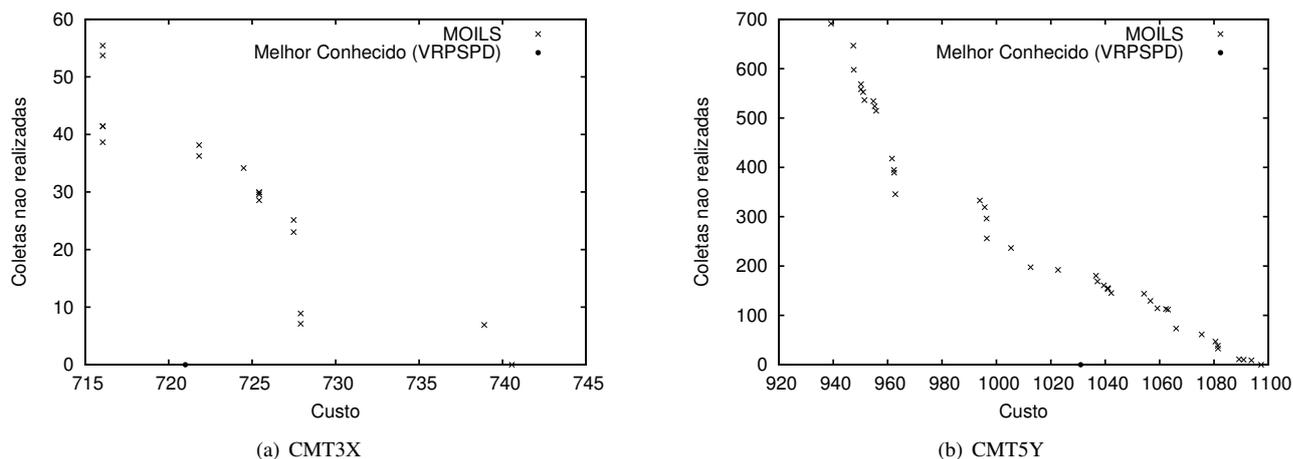


Figura 1: Gráficos das fronteiras para as instâncias CMT3X e CMT5Y

Tabela 3: Melhores resultados obtidos para ambos objetivos

Instância	Considerando $f_1$		Considerando $f_2$ (Coletas 100% realizadas)		
	Melhor $f_1$	$f_2$ associado	$f_1$ associado	Melhor conhecido*	gap** (%)
CMT1Y	448,88	149,50	474,27	466,77	1,61
CMT2X	677,29	112,11	692,26	668,77	3,51
CMT2Y	610,03	295,93	716,58	663,25	8,04
CMT3X	716,05	55,44	726,28	721,27	0,69
CMT3Y	689,13	256,73	739,77	721,27	2,56
CMT12X	630,68	106,43	683,85	644,70	6,07
CMT12Y	615,40	248,04	680,72	659,52	3,21
CMT11X	751,73	347,42	867,78	833,92	4,06
CMT11Y	797,80	183,52	878,19	830,39	5,76
CMT4X	838,26	264,75	882,18	852,46	3,49
CMT4Y	783,42	367,95	886,91	852,35	4,05
CMT5X	1015,01	282,09	1087,97	1030,55	5,57
CMT5Y	939,17	691,26	1097,53	1030,55	6,50

\* Resultados apresentados em [10].

\*\* Diferença percentual entre os resultados obtidos pelo MOILS e os melhores conhecidos.

Pode ser observado, através da Tabela 3, pelas segunda e quarta colunas que, como afirmado anteriormente, a não realização de algumas coletas permite a redução do custo total das rotas. Ainda nesta tabela, percebe-se também que, quando todas as coletas são realizadas o MOILS consegue criar rotas com custos próximos aos melhores resultados conhecidos na literatura.

## 5 CONCLUSÕES

Neste trabalho é proposto uma adaptação do algoritmo ILS para resolução de problemas multiobjetivo, e sua aplicação no Problema de Roteamento de Veículos com Entrega e Coleta Seletiva Multiobjetivo. Este algoritmo mostrou-se como uma técnica eficiente na geração de soluções não dominadas para o problema, principalmente para instâncias que envolvem um número maior de consumidores.

Os resultados encontrados são fortemente dependentes dos mecanismos utilizados nas etapas de perturbação e busca local, sendo que a utilização de outros mecanismos podem melhorar ainda mais os resultados obtidos. Para instâncias pequenas, onde a qualidade foi inferior, comparado às outras instâncias, pode-se executar o algoritmo por um número maior de iterações, visando a obtenção de uma fronteira mais consistente.

Pelo fato de apresentar uma estrutura genérica, assim como o ILS, a alteração das etapas deste algoritmo não interferem no seu comportamento geral, podendo ser utilizado em outros problemas multiobjetivo.

Assim, trabalhos futuros poderão explorar melhor o conjunto de mecanismos de perturbação e busca local como, também, aplicar o algoritmo proposto a outros problemas de otimização combinatória multiobjetivo.

## AGRADECIMENTOS

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro (edital 004/2010/PIBIT-CNPq-UFVJM e projeto PQ:306910/2006-3), à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela concessão de bolsa de estudo através do Programa de Fomento à Pós-Graduação (PROF) e à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) que possibilitaram a realização deste trabalho.

## REFERÊNCIAS

- [1] Associação Brasileira de Logística. “Associação Brasileira de Logística”. [www.aslog.org.br](http://www.aslog.org.br), 2008. Acessado em Abril de 2011.
- [2] H. Min. “The multiple vehicle routing problem with simultaneous delivery and pickup points”. *Transportation Research Part A: General*, vol. 23A, no. 5, pp. 377–386, 1989.
- [3] M. Dell’Amico, G. Righini and M. Salani. “A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection”. *Transportation Science*, vol. 40, pp. 235–247, 2005.
- [4] F. A. T. Montané and R. D. Galvão. “A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service”. *Computers & Operations Research*, vol. 33, pp. 595–619, 2006.
- [5] J.-F. Chen. “Approaches for the vehicle routing problem with simultaneous deliveries and pickups”. *Journal of the Chinese Institute of Industrial Engineers*, vol. 23, no. 2, pp. 141–150, 2006.
- [6] N. A. Wassan, A. H. Wassan and G. Nagy. “A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries”. *Journal of Combinatorial Optimization*, vol. 15, no. 4, pp. 368–386, 2007.
- [7] N. Bianchessi and G. Righini. “Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery”. *Computers & Operations Research*, vol. 34, pp. 578–594, 2007.
- [8] L. M. B. Freitas and F. A. T. Montané. “Metaheurísticas VNS-VND e GRASP-VND para problemas de roteamento de veículos com coleta entrega simultâneas”. *Simpósio de Pesquisa Operacional e Logística da Marinha*, 2008.
- [9] A. Subramanian, L. S. Ochi and L. dos Anjos Formiga Cabral. “An efficient heuristic for the vehicle routing problem with simultaneous pickup and delivery”. Technical report, Instituto de Computação, Universidade Federal Fluminense, 2008.
- [10] A. Subramanian, L. Drummond, C. Bentes, L. Ochi and R. Farias. “A parallel heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery”. *Computers & Operations Research*, vol. 37, no. 11, pp. 1899 – 1911, 2010. Metaheuristics for Logistics and Vehicle Routing.
- [11] E. I. Gökçe. “A revised ant colony system approach to vehicle routing problems”. Master’s thesis, School of Engineering and Natural Sciences, Sabanci Univerity, 2004.
- [12] Y. Gajpal and P. Abad. “An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup”. *Computers & Operations Research*, 2009.
- [13] A. L. Maravilha, V. W. C. Morais, L. Assis and A. Vivas. “Comparação entre duas abordagens bio-inspiradas aplicadas ao problema de roteamento de veículos com coleta e entrega simultâneas”. *XLII Simpósio Brasileiro de Pesquisa Operacional*, 2010.
- [14] I. Gribkovskaia, G. Laporte and A. Shyshou. “The single vehicle routing problem with deliveries and selective pickups”. *Computers & Operations Research*, vol. 35, pp. 2908–2924, 2008.
- [15] H. Süral and J. H. Bookbinder. “The single-vehicle routing problem with unrestricted backhauls”. *Networks*, vol. 41, no. 3, pp. 127–136, 2003.
- [16] X.-L. Liao and C.-K. Ting. “An evolutionary approach for the selective pickup and delivery problem”. In *Proc. IEEE Congress Evolutionary Computation (CEC)*, pp. 1–8, 2010.
- [17] N. Jozefowicz, F. Semet and E.-G. Talbi. “Multi-objective vehicle routing problems”. *European Journal of Operational Research*, vol. 189, pp. 293–309, 2008.
- [18] E. Angel, E. Bampis and L. Gourvès. *Approximation in Multiobjective Problems*, chapter 28, pp. 28.1–28.15. Chapman & Hall/CRC computer & information science, 2007.
- [19] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [20] A. Corberán, E. Fernández, M. Laguna and R. Martí. “Heuristic solutions to the problem of routing school buses with multiple objectives”. *Journal of the Operational Research Society*, vol. 53, no. 4, pp. 427–435, 2002.
- [21] A. H. F. Dias and J. A. Vasconcelos. “Multiobjective genetic algorithms applied to solve optimization problems”. *IEEE Transactions on Magnetism*, vol. 38, no. 2, pp. 1133–1136, 2002.
- [22] N. Jozefowicz, F. Semet and E.-G. Talbi. “An evolutionary algorithm for the vehicle routing problem with route balancing”. *European Journal of Operational Research*, vol. 195, pp. 761–769, 2009.
- [23] M. J. Geiger. “Foundations of the Pareto Iterated Local Search Metaheuristic”. In *MCDM*, pp. 19–23, Grécia, Junho 2006.
- [24] H. R. Lourenço, O. C. Martin and T. Stützle. *Iterated Local Search*, chapter 11, pp. 321–353. Kluwer Academic Publishers, 2003.
- [25] S. Salhi and G. Nagy. “A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling”. *Journal of the Operational Research Society*, vol. 50, pp. 1034–1042, 1999.
- [26] E. Zitzler, K. Deb and L. Thiele. “Comparison of multiobjective evolutionary algorithms: empirical results”. *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.