



## Um Algoritmo Híbrido Baseado em Colônia de Formigas para o Problema de Roteamento de Veículos com Restrições de Cobertura\*

Model magazine Abakos ICEI the PUC Minas

Vitor A A Souza<sup>1</sup>  
Ramon Lopes<sup>2</sup>  
Tiago Januario<sup>3</sup>

### Resumo

O Problema de Roteamento de Veículos (PRV) é um problema clássico de Otimização Combinatória, e a principal motivação para o seu estudo e de suas variações é o alto gasto com o transporte de cargas, impactando nos preços da matéria prima e do produto final. Neste trabalho, estuda-se uma variação do PRV denominada Problema de Cobertura Multi-Veículo (PCMV) que, diferentemente das demais variações existentes na literatura, possui restrições de cobertura e natureza seletiva, além das restrições de carga nos veículos. Este trabalho propõe um algoritmo híbrido baseado em Colônia de Formigas para o PCMV que utiliza um algoritmo *Variable Neighborhood Search* e fundamenta-se na transformação do PCMV em um Problema de Roteamento de Veículos Capacitado, para o qual o método apresentado foi projetado. O algoritmo proposto obteve 16 novos limites primais para as 30 instâncias consideradas em um tempo computacional inferior a 21 segundos, enquanto os resultados apresentados na literatura foram encontrados em até quatro horas de execução.

**Palavras-chave:** Colônia de Formigas, Roteamento de Veículos, Metaheurística Híbrida, Metaheurísticas.

\* Artigo apresentado à Revista Abakos

<sup>1</sup> Universidade Federal de Minas Gerais, Brasil– vitor.andrade@dcc.ufmg.br

<sup>2</sup> Universidade Federal do Recôncavo da Bahia, Brasil– rlopes@ufrb.edu.br

<sup>3</sup> Universidade Federal da Bahia, Brasil– januario@dcc.ufba.br

### Abstract

The Vehicle Routing Problem (VRP) is a classical Combinatorial Optimization problem that has been drawing the scientific community attention due to its applicability in supply chain management. In this work, we study an unusual VRP variation called Multi-vehicle Covering Tour Problem (MCTP) that combines covering and capacity constraints, and selectivity property. To this end, we transform a MCTP instance into a Capacitated Vehicle Routing Problem and propose a hybrid algorithm that embeds a Variable Neighborhood Search algorithm in an Ant Colony Optimization procedure. Our method obtained 16 new upper bounds for the 30 instances considered within 21 seconds, in turn the results presented in the literature were obtained within four hours.

**Keywords:** Ant Colony Optimization, Vehicle Routing Problem, Hybrid Metaheuristic, Metaheuristic.

## 1 INTRODUÇÃO

O Problema de Roteamento de Veículos (PRV) (Dantzig; Ramser, 1959) é um problema  $\mathcal{NP}$ -Difícil<sup>4</sup> clássico de Otimização Combinatória. Nesse problema, deve-se determinar um conjunto de rotas, geralmente compartilhando a mesma origem (depósito), para atender a um conjunto de consumidores de modo a minimizar o custo de deslocamento da frota de veículos. A principal motivação para o estudo do PRV e de suas variações é o alto gasto com o transporte de cargas, impactando nos preços da matéria prima e do produto final. Estima-se que os gastos com transporte de cargas são responsáveis por aproximadamente metade dos custos de logística, podendo chegar a 70% do custos no caso de indústrias de alimentos e de bebidas (Bräysy; Gendreau, 2005). Aplicações reais mostram que o uso de procedimentos computacionais para o planejamento do processo de distribuição resulta em uma significativa redução dos custos com transporte (Toth; Vigo, 2001).

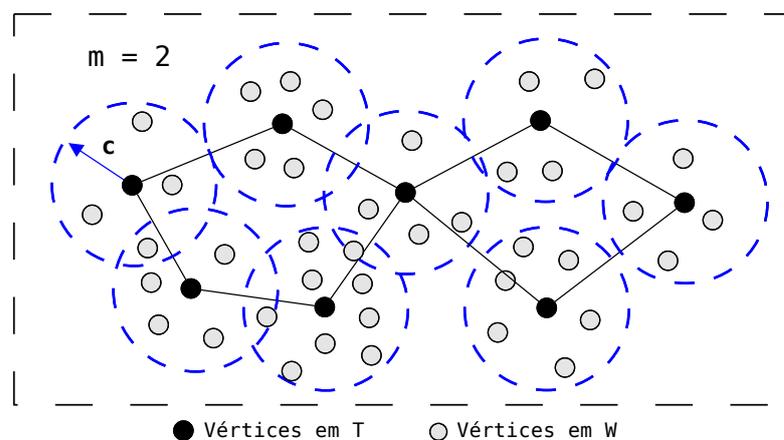
Devido à sua relevância tanto prática quanto teórica, são encontradas diversas variações do PRV na literatura. Dentre as principais, destacam-se: (i) PRV com restrição de distância (Laporte; Desrochers; Nobert, 1984), em que cada rota não deve exceder um determinado limite de distância; (ii) PRV com janela de tempo (Bräysy; Gendreau, 2005), em que cada cliente deve ser visitado dentro de um intervalo de tempo específico, (iii) PRV com coleta e entrega (Savelsbergh; Sol, 1995), em que cada veículo deve realizar coletas e entregas de mercadorias em pontos específicos. Neste trabalho, é estudada uma variação do PRV, denominada Problema de Cobertura Multi-Veículo (PCMV), que possui restrições de cobertura, em que um vértice não visitado deve estar a, no máximo, uma distância pré-definida de algum vértice que seja visitado por um veículo, e natureza seletiva, de forma que nem todos os vértices precisam ser visitados. Vale ressaltar que essas duas características não são muito comuns dentre as variações do PRV existentes na literatura.

---

<sup>4</sup>Do inglês *non-deterministic polynomial*

O PCMV difere das demais variações encontradas na literatura porque (i) cada vértice de um conjunto específico não pode ser visitado, mas deve ser coberto e (ii) cada vértice de um outro conjunto específico deve ser obrigatoriamente visitado por alguma rota, enquanto os demais vértices podem ou não ser visitados. Uma de suas aplicações está relacionada com a distribuição de unidades móveis de saúde em zonas de desastre para fornecer alimentos, água e remédios para a população afetada, de modo que uma região não visitada deve distar no máximo uma distância predefinida de alguma região que tenha sido visitada (Torre; Dolinskaya; Smilowitz, 2012).

O PCMV é definido sobre um grafo  $G = (V \cup W, E)$  simples, completo, finito e não-direcionado, onde  $V$  define o conjunto de vértices percorráveis,  $W$ , o conjunto de vértices que devem ser cobertos e  $E = \{(i, j) \mid i, j \in V \cup W\}$ , o conjunto de arestas. O conjunto  $V$  é composto por um subconjunto de vértices que devem ser visitados, definido por  $T \subseteq V$ , e um subconjunto  $V \setminus T$  de vértices que podem ser visitados ou não. O depósito pertence ao conjunto  $T$ . Para cada aresta  $(i, j) \in E$  é associada uma distância  $d_{ij} \geq 0$ , de tal forma que a propriedade de desigualdade triangular seja respeitada. Um vértice visitado  $v \in V$  cobre um vértice  $w \in W$  se  $d_{vw} \leq c$ , tal que  $c \geq 0$  é um parâmetro. Além disso, um total de  $m$  veículos idênticos são disponibilizados no depósito. Com isso, o objetivo do PCMV é definir um conjunto de, no máximo,  $m$  rotas, que iniciam e terminam o trajeto no depósito, de modo que a distância total percorrida nas rotas seja mínima, satisfazendo as seguintes restrições: (i) cada vértice  $v \in V$  pode ser visitado por, no máximo, uma única rota, (ii) cada vértice  $v \in T$  deve ser visitado por exatamente uma rota, (iii) cada vértice  $v \in W$  deve ser coberto por algum vértice visitado, (iv) o número de vértices em uma rota não pode ultrapassar o valor de um parâmetro  $p$  e (v) a distância percorrida em uma rota não pode ultrapassar o valor de um parâmetro  $q$ . O PCMV é  $\mathcal{NP}$ -Difícil, já que o Problema do Caixeiro Viajante (Dantzig; Fulkerson; Johnson, 1954) é um de seus casos particulares quando  $m = 1$ ,  $T = V$ ,  $W = \emptyset$  e não são impostas as restrições de capacidade ( $p = |V|$  e  $q = \infty$ ).



**Figura 1 – Exemplo de instância do PCMV.**

A Figura 1 ilustra uma instância do PCMV. Nesse cenário, há dois veículos ( $m = 2$ ), portanto até duas rotas devem ser definidas. Os vértices em  $T$ , isto é, aqueles que devem ser obrigatoriamente visitados são representados por vértices em preto; os vértices em  $W$ , ou seja,

aqueles que devem ser cobertos no sentido de estar a uma distância máxima  $c$  de algum vértice que tenha sido visitado são representados por vértices em tonalidade clara. A partir de cada vértice visitado, um círculo de raio  $c$  é definido, de modo a ilustrar os vértices cobertos por aquele vértice. Por fim, pode-se notar que todo vértice não visitado é coberto, podendo ser coberto por mais de um vértice visitado.

O PCMV foi originalmente proposto em (Hachicha et al., 2000). Nesse trabalho, foram propostas uma formulação de Programação Inteira com um número exponencial de restrições do tipo *Generalized Subtour Elimination Constraints* (GSEC) e três heurísticas, as quais são baseadas em heurísticas e vizinhanças clássicas da literatura do PRV, a saber: *savings* (Clarke; Wright, 1964), *sweep* (Gillett; Miller, 1974) e *route-first, cluster-second* (Beasley, 1983). Ademais, experimentos foram realizados em instâncias artificiais e reais e seus resultados foram reportados. Vale ressaltar que, apesar de ter sido proposta uma formulação, nenhum algoritmo foi implementado para resolver o PCMV de forma exata.

Uma abordagem para o PCMV é encontrada em (Jozefowicz, 2012), em que os autores propuseram uma reformulação por Recobrimento de Conjuntos e um algoritmo de Geração de Colunas para o PCMV. Contudo, a reformulação proposta não está de acordo com a definição original do problema, já que em uma solução encontrada pelo algoritmo apresentado: (i) cada vértice em  $T$  pode não ser visitado exatamente uma única vez, e (ii) uma rota pode não satisfazer as restrições acerca do número máximo de vértices visitados e de comprimento máximo. Além disso, o conjunto  $T$  não foi considerado na definição do problema nem foi explicado o processo pelo qual as instâncias foram geradas. Por fim, fica claro que esses autores abordaram uma variação do PCMV, divergindo da especificação original.

Uma outra abordagem para o PCMV é apresentada em (Hà et al., 2013). Nesses trabalhos, os autores propuseram uma formulação baseada em fluxo, um algoritmo *Branch-and-Cut* (BC) e uma metaheurística baseada em *Iterated Local Search* (ILS) (Lourenço; Martin; Stützle, 2002). Contudo, os autores consideraram um caso particular do PCMV, uma vez que a restrição do comprimento máximo da rota foi relaxada. A metaheurística implementada utiliza-se de vizinhanças clássicas da literatura do PRV. Nos experimentos computacionais reportados, em 2 horas de execução, o algoritmo BC foi capaz de atestar a otimalidade de 82 dentre as 95 instâncias consideradas; a metaheurística proposta, por sua vez, foi capaz de encontrar soluções ótimas para 79 dentre as 82 instâncias em que um atestado de otimalidade foi obtido.

Um algoritmo exato para o PCMV conforme definição original é apresentado em (Lopes, 2012; Lopes; Souza; Cunha, 2013). Nesse trabalho, foram propostos um algoritmo *Branch-and-Price* (BP), baseado em uma reformulação por Recobrimento de Conjuntos, e uma heurística denominada Heurística de Geração de Colunas (HGC) que, em conjunto, foram capazes de atestar a otimalidade de 9 das 30 instâncias consideradas dentro de quatro horas de execução. Em termos de limites primais, a heurística HGC obteve soluções muito próximas da solução ótima, quando essa foi encontrada. Contudo, a execução da heurística HGC está atrelada ao algoritmo BP, demandando um tempo computacional considerável. Ainda, para 16 das 30 instâncias, nem sequer foi possível avaliar o limite de relaxação linear.

Com base nos trabalhos relacionados, pode-se concluir que o PCMV permanece mal resolvido na prática, isto é, os métodos existentes são computacionalmente custosos e poucos eficazes. Desse modo, há uma carência de abordagens menos computacionalmente custosas e mais eficazes para esse problema. Na literatura, algoritmos baseados em Colônia de Formigas (CF) (Dorigo; Maniezzo; Colomi, 1996) têm apresentado soluções de boa qualidade com baixo custo computacional na resolução de variações do PRV (Bell; McMullen, 2004; Tan; Zhuo; Zhang, 2006; Yu; Yang; Yao, 2009). Portanto, um algoritmo baseado em Colônia de Formigas (CF) pode suprir a carência supracitada para o PCMV. Com base na revisão bibliográfica realizada, não foi encontrado trabalho algum que investigue algoritmos baseado em CF no contexto do PCMV.

Neste trabalho, propõe-se um algoritmo híbrido baseado em CF para o PCMV, até então inexistente na literatura. O método proposto utiliza-se de um algoritmo *Variable Neighborhood Search* (VNS) (Mladenovic; Hansen, 1997) para alcançar um ótimo local a partir da melhor solução encontrada ao final do algoritmo baseado em CF. Experimentos foram realizados, e a qualidade das soluções obtidas foram comparadas com as melhores soluções reportadas na literatura e com aquelas encontradas pela versão do algoritmo baseado em CF em que o algoritmo *Variable Neighborhood Search* (VNS) não foi utilizado. O método proposto encontrou 16 novos melhores limites primais em tempo computacional inferior a 21 segundos, ao passo que as soluções comparadas são encontradas em até quatro horas de execução.

## 2 ALGORITMO HÍBRIDO BASEADO EM COLÔNIA DE FORMIGAS

A abordagem utilizada neste trabalho para resolver o PCMV consiste em duas etapas. Na primeira etapa, resolve-se um Problema de Cobertura de Conjuntos Ponderado (PCCP) para definir um conjunto de vértices  $V' \subset V$  que seja capaz de cobrir todos aqueles em  $W$  e define-se o subgrafo induzido de  $G$  por  $V' \cup T$  como  $G' = (V' \cup T, E')$ . Uma vez que todos os vértices em  $W$  são, por construção, cobertos por  $V'$ , uma solução para o PCMV pode ser encontrada resolvendo um Problema de Roteamento de Veículos Capacitado (PRVC) sobre  $G'$ , em que cada veículo deve atender às restrições de capacidade definidas no PCMV. Portanto, as restrições de cobertura presentes no PCMV são automaticamente satisfeitas, tornando o novo problema a ser resolvido mais tratável. Como resultado, pode-se utilizar de algoritmos menos custosos do ponto de vista computacional. Por fim, na última etapa, um algoritmo CF é aplicado para resolver o Problema de Roteamento de Veículos Capacitado (PRVC) formulado, e a melhor solução encontrada é utilizada como solução inicial para uma metaheurística VNS. Com a utilização da metaheurística VNS, pretende-se explorar um espaço de soluções ainda maior em relação àquele explorado pelo algoritmo CF, fugindo de ótimos locais, e, portanto, potencialmente encontrar soluções melhores. As etapas são explicadas detalhadamente a seguir.

O PCCP é uma variação do problema clássico de Cobertura de Conjuntos em que pesos são associados aos conjuntos e sua função objetivo consiste em minimizar a soma de tais pesos. No contexto do PCMV, o PCCP foi adaptado de forma que há um conjunto para cada vértice

$i \in V$  que representa o conjunto de vértices em  $W$  cobertos pelo vértice  $i$ . Formalmente, o seguinte modelo de Programação Linear Inteira define o problema:

$$\min \sum_{v \in V} \mu_v x_v \quad (1)$$

sujeito a

$$\sum_{v \in V} \omega_{vw} x_v \geq 1 \quad \forall w \in W \quad (2)$$

$$x_v \in \{0, 1\} \quad \forall v \in V \quad (3)$$

em que  $x_v$  é uma variável de decisão binária assumindo valor 1 se o conjunto referente ao vértice  $v$  for selecionado para compor a solução ou valor 0 caso contrário;  $\mu_v$  representa o peso associado ao conjunto do vértice  $v$  e foi definido como a média aritmética dos custos das arestas incidentes ao vértice  $v$ ; por fim,  $\omega_{vw}$  é um parâmetro que assume valor 1 se o vértice  $v$  cobre o vértice  $w$ , ou seja,  $d_{vw} \leq c$ , ou valor 0 caso contrário. A função objetivo (1) consiste em minimizar a soma dos pesos associados aos conjuntos. As restrições (2) garantem que todo vértice  $w \in W$  é coberto, enquanto as restrições (3) impõem que as variáveis de decisão são binárias. O conjunto de vértices que compõem a solução ótima desse modelo é denotado por  $V'$  ao longo desta seção. Com base em  $V'$ , modela-se um PRVC, cuja solução é, por construção, válida para o PCMV.

O PRVC é definido sobre o grafo  $G' = (V' \cup T, E')$  a partir do PCMV, em que há  $m$  veículos idênticos e capacitados, no sentido de que cada veículo pode visitar no máximo  $p$  clientes e a distância percorrida não pode ultrapassar o valor definido por  $q$ . No PRVC, deve-se encontrar um conjunto de até  $m$  rotas, satisfazendo as restrições de capacidade de cada veículo, em que cada vértice de  $G'$  seja visitado apenas uma vez por um único veículo. A função objetivo consiste em minimizar a soma do comprimento de cada uma das rotas. Vale ressaltar que uma solução viável do PRVC é viável para o PCMV, pois, por construção (i) a visitação de todos os vértices de  $V' \cup T$  pelo conjunto de rotas garante a cobertura de todos os vértices do conjunto  $W$ , (ii) as rotas satisfazem as restrições de capacidade do PCMV e (iii) todos os vértices em  $T$  são visitados. Desse modo, pode-se obter uma solução viável para o PCMV resolvendo-se uma instância do PRVC, em que as restrições de cobertura e a natureza seletiva não são diretamente modeladas, facilitando a sua resolução.

Com base no PRVC modelado, sua resolução fornece uma solução viável para o PCMV. Para esse fim, propomos um algoritmo baseado em CF. CF é uma metaheurística populacional que se baseia no comportamento natural das formigas em uma colônia quando estão em busca de alimento. Na natureza, inicialmente, as formigas andam aleatoriamente até encontrar alimento e retornam à colônia. Durante o deslocamento, cada um desses insetos deposita feromônio por todo o trajeto percorrido. As formigas tendem a seguir os rastros em que há uma maior concentração de feromônio, ou seja, aqueles pelos quais mais formigas passaram. Uma vez que, no decorrer do tempo, o feromônio depositado se evapora e caminhos mais curtos tendem a ser percorridos mais rapidamente, a concentração de feromônio em tais caminhos tenderá a

ser maior. Portanto, em algum momento, as formigas percorrerão o menor caminho existente entre o alimento e a colônia. No contexto computacional de CF, uma formiga artificial constrói probabilisticamente uma solução para o problema com base no feromônio já depositado no ambiente e, em seguida, deposita feromônio com base na solução construída.

A fase de construção da solução, iniciada do depósito, consiste em escolher probabilisticamente um vértice viável e não visitado para compor a rota a partir do último vértice visitado. Um vértice é considerado viável para uma rota em construção quando sua inclusão não viola as restrições de capacidade do veículo. Caso não haja vértice viável, retorna-se ao depósito e inicia-se a construção de uma nova rota a partir desse. Esse processo é repetido até que todos os vértices sejam visitados, de modo a respeitar o número máximo de  $m$  rotas. Na fase de construção, uma formiga  $k$  no vértice  $i$  seleciona probabilisticamente um vértice  $j$  (viável e não visitado) para compor a solução conforme:

$$p_{ij}^k = \frac{[\Delta]^{-\alpha} [\tau_{ij}]^\beta}{\sum_{l \in N_i} [\Delta]^{-\alpha} [\tau_{il}]^\beta} \quad (4)$$

em que  $\Delta = d_{ij} + d_{j1} - d_{i1}$  denota o incremento na rota devido a inserção do vértice  $j$  e  $\tau_{ij}$  denota a concentração de feromônio na aresta que conecta os vértices  $i$  e  $j$ . Os parâmetros  $\alpha$  e  $\beta$  são utilizados como fatores de decaimento para  $\Delta$  e  $\tau_{ij}$ , respectivamente. Considerando a Equação (4), a seleção de um vértice  $j$  depende dos seguintes fatores: (i) concentração de feromônio  $\tau_{ij}$  que indica o quão boa é a escolha do próximo vértice  $j$  a partir do vértice atual  $i$  com base no feromônio depositado pelas formigas e (ii)  $\Delta$  que indica o quão promissora é a escolha de  $j$  como próximo vértice. Finalizada a construção das soluções no algoritmo CF, inicia-se a fase de atualização do feromônio.

A fase de atualização consiste na evaporação e no depósito do feromônio. Em primeiro lugar, a concentração do feromônio em todas as arestas é reduzida por um fator constante determinado pela seguinte equação:

$$\tau_{ij} := \left( \rho + \frac{\theta}{\bar{L}} \right) \tau_{ij}, \forall (i, j) \in E' \quad (5)$$

em que,  $0 \leq (\rho + \theta/\bar{L}) \tau_{ij} < 1$  é o fator de evaporação,  $0 \leq \rho < 1$  é o fator de persistência do feromônio,  $\theta$  é uma constante, e  $\bar{L} = \sum_{k=1}^n L^k / m$  é o custo médio das soluções, em que  $L^k$  é o custo da solução construída pela formiga  $k$  e  $n$  é o número total de formigas consideradas em cada iteração do algoritmo. No processo de evaporação do feromônio, considera-se o valor de  $\bar{L}$ , de modo que quanto maior o seu valor menor quantidade de feromônio evapora.

Após o processo de evaporação do feromônio, um total de  $\sigma$  formigas (formigas *elite*), associadas com as melhores soluções encontradas, e a formiga associada com a melhor solução encontrada até o momento depositam feromônio nas arestas utilizadas das soluções construídas de acordo com a seguinte equação:

$$\tau_{ij} := \tau_{ij} + \sum_{\lambda=0}^{\sigma-1} \Delta \tau_{ij}^\lambda + \Delta \tau_{ij}^* \quad (6)$$

em que:

$$\Delta\tau_{ij}^{\lambda} = \begin{cases} \frac{(\sigma-\lambda)}{L^{\lambda}} & \text{se a } \lambda\text{-ésima formiga elite passa pela aresta } (i, j) \\ 0 & \text{caso contrário} \end{cases} \quad (7)$$

$$\Delta\tau_{ij}^{*} = \begin{cases} \frac{\sigma}{L^{*}} & \text{se a aresta } (i, j) \text{ faz parte da melhor solução encontrada } L^{*} \\ 0 & \text{caso contrário} \end{cases} \quad (8)$$

Ao término do algoritmo CF, a melhor solução encontrada é utilizada como solução inicial para o algoritmo VNS. A ideia básica da metaheurística VNS reside na mudança sistemática de vizinhanças dentro de um algoritmo de busca local. Para esse fim, as vizinhanças são ordenadas de acordo com sua complexidade, e há uma mudança de uma vizinhança para outra imediatamente mais custosa se não houver melhoria no custo da solução corrente ao explorar aquela vizinhança. Sempre que a exploração de uma vizinhança resultar em uma melhoria da solução corrente, retorna-se para a vizinhança de menor complexidade. No contexto do algoritmo VNS implementado neste trabalho, quatro vizinhanças foram utilizadas com base na política melhor-aprimorante, a qual consiste em avaliar todas as soluções da vizinhança e implementar o melhor movimento. Essas vizinhanças são discutidas a seguir.

A vizinhança clássica 2-OPT (Croes, 1958) foi utilizada como primeira vizinhança. Esta vizinhança é composta por soluções obtidas a partir da remoção de 2 arestas do caminho seguida da reconexão do caminho de modo a formar uma nova rota. Experimentos preliminares mostraram que a utilização dessa vizinhança obteve resultados comparáveis com aqueles obtidos utilizando-se vizinhanças maiores, por exemplo, 3-OPT, com menor complexidade computacional.

A segunda vizinhança, denominada *SWAP*, é composta por soluções obtidas pela permutação de dois vértices visitados por uma rota. Sejam  $i, j \in S$  dois vértices quaisquer visitados pela rota, uma solução da vizinhança em questão é obtida pela substituição de  $i$  por  $j$  e de  $j$  por  $i$ , resultando na mudança da ordem de visitação dos vértices em questão.

A terceira vizinhança, denominada *EXCHANGE*, é composta por soluções obtidas pela troca entre dois vértices visitados em rotas distintas. Uma solução da vizinhança em questão é obtida pela troca de um vértice  $i$  visitado em uma rota por um vértice  $j$  visitado em outra rota. Vale ressaltar que essa vizinhança é formada por todas as combinações possíveis de pares de vértices em rotas distintas.

A quarta vizinhança, denominada *REMOVE-INSERTION*, é composta por soluções obtidas pela substituição de um dos vértices visitados por outro que ainda não o tenha sido. Para esse fim, cada vértice em  $V \setminus T$  visitado pela rota é considerado para remoção e um vértice dentre aqueles não visitados é considerado para inserção, de modo a cobrir aqueles vértices que deixaram de ser cobertos com a remoção e a garantir a viabilidade da rota.

O Algoritmo 1 apresenta o pseudocódigo do método apresentado, denominado por Colônia de Formigas com VNS (CFVNS). Na linha 1, por meio do PCCP, cria-se a instância do PRVC. Na linha 2, a variável  $\pi$ , que representa o número de iterações sem melhoria, é inicializada em 0. Nas linhas 3–12, o laço referente ao algoritmo CF é repetido até atingir o número

---

**Algorithm 1:** Colônia de Formigas com VNS.

---

**Entrada:** Instância do PCMV.

**Saída** : Melhor solução encontrada.

```
1 Cria instância do PRVC
2  $\pi \leftarrow 0$ 
3 enquanto  $\pi < \pi_{max}$  faça
4   para cada formiga  $k \in \{1, \dots, n\}$  faça
5      $k \leftarrow$  Constrói solução
6      $\pi \leftarrow \pi + 1$ 
7     para cada formiga  $k \in \{1, \dots, n\}$  faça
8       se  $L^k < L^*$  então
9          $L^* \leftarrow L^k$ 
10         $k^* \leftarrow k$ 
11         $\pi \leftarrow 0$ 
12   Atualiza feromônio
13  $k^* \leftarrow$  VNS( $k^*$ )
14 retorna  $k^*$ 
```

---

máximo de iterações sem melhoria, denotado por  $\pi_{max}$ . Nas linhas 4–5, para cada formiga constrói-se uma solução. Na linha 6, o valor da variável  $\pi$  é incrementado. Nas linhas 7–11, o valor da melhor solução  $L^*$  é atualizado. Na linha 12, o feromônio é atualizado. Na linha 13, o algoritmo VNS é aplicado na melhor solução encontrada  $k^*$ . Na linha 14, a melhor solução obtida é retornada.

### 3 RESULTADOS

Nos experimentos realizados, as instâncias consideradas foram geradas de acordo com a metodologia proposta em (Hachicha et al., 2000). Essa metodologia propõe a geração de  $|V|+|W|$  pontos em uma região quadrada, definida por  $[0, 100] \times [0, 100]$ , sendo que o depósito deve ser gerado em uma região quadrada, definida por  $[25, 75] \times [25, 75]$ . Os pontos devem ser gerados com base em uma distribuição contínua uniforme de números aleatórios. O conjunto  $V$  é composto pelos  $|V|$  primeiros pontos gerados, dos quais os  $|T|$  primeiros pontos são atribuídos ao conjunto  $T$ ; os demais são atribuídos ao conjunto  $W$ . Por fim, o valor de  $c$  foi definido como  $c = \max\{\max_{h \in V \setminus T} \min_{l \in W} \{d_{lh}\}, \max_{l \in W} \{d_{lh(l)}\}\}$ , em que  $h(l)$  é o segundo vértice em  $V \setminus T$  mais próximo do vértice  $l$ . Desse modo, cada vértice em  $V \setminus T$  cobre ao menos um vértice em  $W$  e cada vértice de  $W$  é coberto por no mínimo dois vértices de  $V \setminus T$ . Apesar de os autores terem proposto que  $d_{ij}$  é definido como a distância euclidiana entre os vértices em questão, neste trabalho,  $d_{ij}$  foi definido como o teto da distância euclidiana entre  $i$  e  $j$ . Foram geradas instâncias com as seguintes configurações  $|V| = \{50, 100, 200\}$ ,  $|W| = \{50, 100, 200\}$ ,  $|T| = \{1, \lceil |V|/4 \rceil + 1, \lceil |V|/2 \rceil + 1\}$ ,  $p = \{4, \infty\}$  e  $q = \{200, \infty\}$ . Por fim, o parâmetro  $m$  foi configurado para  $\lceil |V|/p \rceil$  quando  $p = 4$  ou para  $\infty$  quando  $p = \infty$ .

Para realizar o ajuste de parâmetros do algoritmo apresentado neste texto, foram amostrados 30 valores distintos, obtidos a partir da sequência de Hammersley (Wong; Luk; Heng, 1997), para cada um dos seguintes parâmetros dentro dos seguintes intervalos: (i)  $\alpha$ , variando no intervalo [1, 5]; (ii)  $\beta$ , variando no intervalo [2, 8]; (iii)  $\rho$ , variando no intervalo [0.4, 0.99]; (iv)  $\theta$ , variando no intervalo [40, 99]. Com base nos resultados computacionais obtidos pela execução da CFVNS para todas as combinações de parâmetros apresentados, totalizando  $30^4$  experimentos para cada instância, extraímos a média dos parâmetros com os quais foram obtidas as melhores soluções. Os parâmetros foram definidos para os seguintes valores:  $n = 100$ ,  $\sigma = 3$ ,  $\pi_{max} = 50$ ,  $\tau_0 = 10$ , como a concentração inicial de feromônio;  $\alpha = 2,03$ ,  $\beta = 5,10$ ,  $\rho = 0,58$ ,  $\theta = 71$ .

Os experimentos foram executados em uma máquina com processador Intel Core 2 Duo E8400 3.00 GHz e 5 GB de memória RAM, executando sobre o sistema operacional Ubuntu 12.04. Os algoritmos propostos foram implementados em C++ e compilados com GCC 4.7.3. O ILOG CPLEX versão 12.5 foi utilizado apenas como resolvidor de Programação Linear Inteira para o PCCP apresentado. Os resultados a serem apresentados foram obtidos de um total de 50 execuções considerando os valores dos parâmetros supracitados.

T	V	W	CF		CFVNS		mls	DPR	t(s)
			Mínimo	Médio	Mínimo	Médio			
1	50	50	338	339,3	330	330,3	*328	-0,61%	0,2
1	50	100	355	355,0	353	353,0	*310	-12,18%	0,2
1	100	100	500	521,7	500	513,3	501	0,20%	0,5
1	100	200	620	652,3	611	629,8	*604	-1,15%	0,6
1	200	200	710	735,9	695	712,9	723	4,03%	0,9
13	50	50	595	613,4	595	605,7	*571	-4,03%	0,6
13	50	100	561	584,8	555	558,5	*543	-2,16%	0,7
26	100	100	1062	1092,0	1037	1056,7	1064	2,60%	2,5
26	100	200	1348	1393,7	1320	1356,0	1289	-2,35%	2,6
51	200	200	1782	1836,4	1702	1733,4	1726	1,41%	6,2
26	50	50	914	942,8	902	917,2	905	0,33%	1,5
26	50	100	870	897,5	833	860,8	846	1,56%	1,6
51	100	100	1541	1611,0	1502	1535,3	1485	-1,13%	5,9
51	100	200	1878	1953,3	1822	1864,3	1818	-0,22%	6,4
101	200	200	2945	3067,9	2732	2796,5	2766	1,24%	20,5

**Tabela 1 – Resultados computacionais com  $p = 4$  e  $q = 200$ . O símbolo \* destaca os valores correspondentes à solução ótima da instância respectiva.**

As Tabelas 1 e 2 apresentam os resultados computacionais para as instâncias e configurações consideradas. As três primeiras colunas apresentam a quantidade de vértices nos conjuntos  $T$ ,  $V$  e  $W$ , respectivamente. As duas colunas seguintes apresentam respectivamente o custo mínimo e o custo médio obtidos pelo algoritmo CF sem considerar a etapa de VNS. As duas próximas colunas apresentam respectivamente o custo mínimo e o custo médio obtidos pela CFVNS. As três últimas colunas apresentam o melhor limite superior (mls) reportado em (Lopes; Souza; Cunha, 2013), o desvio percentual relativo definido por  $DPR = 100 \times \frac{mls - CFVNS_{best}}{CFVNS_{best}}$ , em que

T	V	W	CF		CFVNS		mls	DPR	t(s)
			Mínimo	Médio	Mínimo	Médio			
1	50	50	285	288,4	272	276,6	*272	0,00%	0,2
1	50	100	302	302,9	272	272,2	*271	-0,37%	0,1
1	100	100	368	388,6	357	360,6	354	-0,84%	0,4
1	100	200	415	447,9	384	404,5	393	2,34%	0,5
1	200	200	457	493,2	432	437,9	457	5,79%	0,8
13	50	50	374	394,2	349	364,1	*349	0,00%	0,6
13	50	100	398	411,1	367	378,6	*363	-1,09%	0,6
26	100	100	527	561,5	488	509,8	507	3,89%	2,1
26	100	200	570	603,5	520	547,5	528	1,54%	2,0
51	200	200	742	815,9	625	668,1	653	4,48%	5,2
26	50	50	468	507,8	445	465,7	475	6,74%	1,3
26	50	100	489	511,8	450	478,0	456	1,33%	1,3
51	100	100	641	691,0	568	614,4	562	-1,06%	4,0
51	100	200	719	798,4	649	686,1	689	6,16%	4,8
101	200	200	1053	1237,5	850	902,2	889	4,59%	13,4

**Tabela 2 – Resultados computacionais com  $p = \infty$  e  $q = \infty$ . O símbolo \* destaca os valores correspondentes à solução ótima da instância respectiva.**

$CFVNS_{best}$  representa o melhor resultado obtido pela CFVNS, e, por fim, o tempo gasto em segundos para a execução da CFVNS. Por fim, o símbolo \*, na coluna mls, indica que o valor apresentado é o custo da solução ótima reportado na literatura para aquela instância.

A Tabela 1 apresenta os resultados computacionais obtidos no cenário em que  $p = 4$  e  $q = 200$ . Dentre as 15 instâncias consideradas, a CFVNS foi capaz de obter novos limites primais em 7 delas. Por sua vez, a Tabela 2 apresenta os resultados computacionais obtidos quando  $p = \infty$  e  $q = \infty$ . Nesse cenário, a CFVNS foi capaz de obter 9 novos limites primais dentre as 15 instâncias consideradas. Além disso, a metaheurística apresentada obteve a solução ótima para 2 instâncias avaliadas e o tempo máximo de execução para esse conjunto de instâncias foi de 13,4 segundos. É importante ressaltar que os resultados reportados em (Lopes; Souza; Cunha, 2013) foram obtidos dentro de 4 horas de execução do algoritmo BP e de uma heurística em conjunto, ao passo que a CFVNS utilizou no máximo 20,5 segundos no mesmo conjunto de instâncias. Com base nos resultados apresentados em ambos os cenários, nota-se que, como resultado da aplicação do algoritmo VNS, o algoritmo CFVNS encontrou soluções melhores em média quando comparado ao algoritmo CF.

## 4 CONCLUSÕES

Neste trabalho, foi abordado o Problema de Cobertura Multi-Veículo, uma variação do PRV que inclui restrições de cobertura, em que os vértices não visitados devem estar a, no máximo, uma distância pré-definida de algum vértice que seja visitado, e natureza seletiva, de forma que nem todos os vértices precisam ser visitados. Com base na revisão bibliográfica

realizada, os métodos propostos na literatura não se mostraram eficazes, já que poucas instâncias foram resolvidas na otimalidade e os métodos demandaram um elevado tempo computacional.

Desse modo, este artigo apresenta um algoritmo híbrido baseado em Colônia de Formigas com *Variable Neighborhood Search* (CFVNS) para o PCMV, uma variação do PRV que inclui restrições de cobertura, em que um vértice não visitado deve estar a, no máximo, uma distância pré-definida de algum vértice que seja visitado por um veículo, e natureza seletiva, de forma que nem todos os vértices precisam ser visitados.

Experimentos foram realizados para avaliar o método proposto, e a qualidade das soluções obtidas foram comparadas com as melhores soluções reportadas na literatura e com as soluções encontradas pelo algoritmo proposto sem a utilização do algoritmo VNS. A CFVNS proposta obteve 16 novos limites primais para as instâncias consideradas em um tempo computacional inferior a 21 segundos, enquanto os resultados apresentados na literatura foram encontradas em até quatro horas de execução.

## Referências

- BEASLEY, J. E. Route first-cluster second methods for vehicle routing. **Omega**, v. 11, n. 4, p. 403–408, 1983.
- BELL, J. E.; MCMULLEN, P. R. Ant colony optimization techniques for the vehicle routing problem. **Advanced Engineering Informatics**, v. 18, n. 1, p. 41–48, 2004.
- BRÄYSY, O.; GENDREAU, M. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. **Transportation Science**, INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, EUA, v. 39, p. 104–118, fev. 2005.
- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. **Operations Research**, v. 12, n. 4, p. 568–581, 1964.
- CROES, A. A method for solving traveling salesman problems. **Operations Research**, v. 5, p. 791–812, 1958.
- DANTZIG, G.; FULKERSON, R.; JOHNSON, S. Solution of a large-scale traveling-salesman problem. **Operations Research**, v. 2, p. 393–410, 1954.
- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. **Management Science**, v. 6, n. 1, p. 80–91, 1959.
- DORIGO, M.; MANIEZZO, V.; COLORNI, A. The ant system: Optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 26, n. 1, p. 29–41, 1996.
- GILLET, B. E.; MILLER, L. R. A heuristic algorithm for the vehicle-dispatch problem. **Operations Research**, v. 22, n. 2, p. 340–349, 1974.
- HACHICHA, M. et al. Heuristics for the multi-vehicle covering tour problem. **Computers & Operations Research**, Elsevier Science Ltd., Oxford, UK, v. 27, n. 1, p. 29–42, jan. 2000.
- HÀ, M. H. et al. An exact algorithm and a metaheuristic for the multi-vehicle covering tour problem with a constraint on the number of vertices. **European Journal of Operational Research**, v. 226, n. 2, p. 211–220, 2013.
- JOZEFOWIEZ, N. Column generation for the multiple vehicle covering tour problem. In: **Book of Extended Abstracts of 5th International Workshop on Freight Transportation and Logistics**, 2012. p. 534–537.
- LAPORTE, G.; DESROCHERS, M.; NOBERT, Y. Two exact algorithms for the distance-constrained vehicle routing problem. **Networks**, Wiley Subscription Services, Inc., A Wiley Company, v. 14, n. 1, p. 161–172, 1984.
- LOPES, R. **Algoritmos Exatos e Heurísticos para Problemas Seletivos de Roteamento de Veículos com Restrições de Cobertura**. 2012. Dissertação (Mestrado) — Universidade Federal de Minas Gerais.
- LOPES, R.; SOUZA, V. A. A.; CUNHA, A. S. da. A branch-and-price algorithm for the multi-vehicle covering tour problem. **Electronic Notes in Discrete Mathematics**, v. 44, p. 61–66, 2013.

LOURENÇO, H. R.; MARTIN, O.; STÜTZLE, T. Iterated local search. In: GLOVER, F.; KOCHENBERGER, G. (Ed.). **Handbook of Metaheuristics**. [S.l.]: Kluwer Academic Publishers, Norwell, MA, 2002, (International Series in Operations Research & Management Science, v. 57). p. 321–353.

MLADENOVIC, N.; HANSEN, P. Variable neighborhood search. **Computers & Operations Research**, v. 24, n. 11, p. 1097–1100, 1997.

SAVELSBERGH, M. W. P.; SOL, M. The general pickup and delivery problem. **Transportation Science**, v. 29, p. 17–29, 1995.

TAN, X.; ZHUO, X.; ZHANG, J. Ant colony system for optimizing vehicle routing problem with time windows (VRPTW). In: HUANG, D.-S.; LI, K.; IRWIN, G. (Ed.). **Computational Intelligence and Bioinformatics**. [S.l.]: Springer Berlin Heidelberg, 2006, (Lecture Notes in Computer Science, v. 4115). p. 33–38.

TORRE, L. E. de la; DOLINSKAYA, I. S.; SMILOWITZ, K. R. Disaster relief routing: Integrating research and practice. **Socio-Economic Planning Sciences**, v. 46, n. 1, p. 88–97, mar. 2012.

TOTH, P.; VIGO, D. **The Vehicle Routing Problem**. [S.l.]: Society for Industrial and Applied Mathematics, 2001.

WONG, T.-T.; LUK, W.-S.; HENG, P.-A. Sampling with hammersley and halton points. **Journal of Graphical Tools**, A. K. Peters, Ltd., Natick, MA, USA, v. 2, n. 2, p. 9–24, nov. 1997.

YU, B.; YANG, Z.-Z.; YAO, B. An improved ant colony optimization for vehicle routing problem. **European Journal of Operational Research**, v. 196, n. 1, p. 171–176, 2009.