

**METAHEURÍSTICAS SIMULATED ANNEALING E PESQUISA EM VIZINHANÇA
VARIÁVEL APLICADAS AO PROBLEMA DE ROTEAMENTO PERIÓDICO DE
VEÍCULOS PARA COLETA DE LIXO****William Higino****Herlisson Maciel Bezerra****Eliseu Junio Araújo****Kelly Cristina Poldi****Antônio Augusto Chaves**

Universidade Federal de São Paulo – UNIFESP, Instituto de Ciência e Tecnologia
Rua Talin, 330, CEP 12231-280, São José dos Campos, SP
william.higino; herlisson.maciel; araujo.elisei; kelly.poldi; antonio.chaves@unifesp.br

RESUMO

A classe de problemas de roteamento de veículos comumente possui o objetivo de minimizar o custo associado ao atendimento de um grupo de consumidores. Desta classe advém o Problema de Roteamento Periódico de Veículos (PRPV) que deve gerar um conjunto de rotas para cada dia tendo os custos globais minimizados e de modo a respeitar todas as restrições envolvidas. Neste trabalho é estudado o problema de coleta de lixo modelado com um PRPV. Para resolver este problema propõe-se a aplicação das metaheurísticas *Simulated Annealing* e Pesquisa em Vizinhança Variável, que possuem o objetivo de gerar boas soluções para o problema a partir da busca em vizinhanças da solução corrente. Para os testes computacionais é utilizado os dados obtidos de um caso real do município de Ponte de Lima localizado em Portugal.

PALAVRAS CHAVE. Problemas de Roteamento de Veículos, Problema de Roteamento periódico de Veículos, Metaheurísticas.

ÁREA PRINCIPAL. Metaheurísticas.

ABSTRACT

The class of vehicle routing problems (VRP) involves designing the optimal set of routes for fleets of vehicles for the purpose of serving a given set of customers. A variant of the VRP is the Vehicle Periodic Routing Problem (VPRP) that consist of generating a set of routes for each day with minimal global costs and respecting all restrictions of the problem. This paper present a model for the VPRP applied to solid waste collection. To solve this problem is proposed two classical metaheuristics: *Simulated Annealing* and *Variable Neighborhood Search*. In the computational tests is used a real-world case of Ponte de Lima, a Portuguese municipality.

KEYWORDS. Vehicle Routing Problem, Vehicle Periodic Routing Problem, Metaheuristics.

MAIN AREA. Metaheuristics.

1. Introdução

A velocidade do crescimento populacional acarreta verdadeiros problemas no ambiente urbano, afetando a qualidade de vida de milhões e milhões de pessoas em todo o mundo. Um dos problemas advindos do crescimento populacional é a acelerada produção de lixo urbano sólido, principalmente nas últimas décadas.

O destino do lixo urbano sólido é um problema cuja solução se torna cada vez mais difícil. Em Nova York, cidade do mundo que mais produz lixo, a média diária de lixo produzido é de 13 mil toneladas (Coelho e Soares, 2001). É natural então perguntar-se como gerenciar da forma mais eficiente possível, como no caso de Nova York, 1,5 quilo de lixo *per capita* por dia.

Um mau gerenciamento desse lixo pode comprometer o meio ambiente, a saúde da população no entorno, além dos custos significativos. Segundo Cunha e Caixeta (2002) temos três fatores decisivos pra o estudo do gerenciamento desse lixo:

- A grande quantidade de lixo gerada, como o exemplo de Nova York, sendo que o índice brasileiro *per capita* está em torno de 0,5 a 1 kg/habitante/dia;
- Os impactos ambientais e à população no entorno onde os resíduos são depositados, principalmente causados pelo chorume. Um líquido de odor forte e alto potencial de contaminação resultante do processo de putrefação dos resíduos, que podem contaminar o solo, rios e lençóis freáticos;
- Altos gastos financeiros do gerenciamento do lixo urbano sólido. No Brasil, os serviços de limpeza respondem, em média, por 7% a 15% do orçamento dos municípios. Sendo que, aproximadamente 50% dos custos totais de gerenciamento são de coleta e transporte.

O presente trabalho foca-se no estudo de uma parte do problema de gerenciamento de lixo, que é a coleta e o transporte. O problema foi abordado como um problema de roteamento periódico de veículos para coleta de lixo.

No Problema de Roteamento de Veículos (PRV) (Christofides *et al.*, 1979), que tem inúmeras aplicações práticas e é um dos problemas mais importantes de otimização, temos uma frota de veículos, com capacidade conhecida, que devem visitar clientes que têm também uma demanda conhecida. Tem-se como objetivo de encontrar o melhor conjunto de rotas, para que haja o menor custo de utilização dos veículos, sem exceder suas capacidades (Toth e Vigo, 2002).

Já o Problema de Roteamento Periódico de Veículos (PRPV) (Christofides e Beasley, 1984) é uma generalização do PRV que tem o mesmo objetivo podendo ser visualizado da seguinte forma: a geração de um conjunto de rotas para cada dia tendo os custos globais minimizados e de modo a respeitar todas as restrições envolvidas. Neste trabalho, de acordo com Bianchi-Aguiar *et al.* (2011), foram incorporadas condições adicionais ao PRPV para tratar o problema de coleta de lixo: a frota utilizada é heterogênea, ou seja, a capacidade de cada veículo e/ou custos são distintos; existem dois locais a serem utilizados como depósitos; a demanda diária de um cliente pode ser variável; cada rota diária de um veículo é limitada em função da sua distância para que não exceda o tempo máximo de trabalho do motorista.

O PRPV pertence à classe de problemas NP-difícil (Golden *et al.*, 1995). Sendo assim, os métodos heurísticos são as técnicas propostas mais utilizadas para a sua resolução. Dentre os métodos utilizados destacam-se Algoritmos Genéticos (Holland, 1975), *Greedy Randomized Adaptive Search Procedure* (GRASP) (Feo e Resende, 1995), Busca Tabu (Glover, 1986), entre outros.

Na literatura Carvalho *et al.* (2003) apresentam seu problema tratado como um PRV e propõe um método, de fácil entendimento e poucos parâmetros, que cria uma solução inicial e a refina para conseguir seus resultados computacionais. Já Gonçalves *et al.* (2005) apresentam seu problema modelado como Problema de Roteamento Periódico de Unidades Móveis de Pistoneio, que difere do problema clássico principalmente por não ter uma definição prévia do número de visitas de cada cliente. Os autores também fazem uso de heurísticas para a obtenção de seus resultados.

Neste trabalho, para buscar boas soluções para o problema foram utilizadas duas metaheurísticas: *Simulated Annealing* (Kirkpatrick *et al.*, 1983) e Pesquisa em Vizinhança

Variável (Mladenovic e Hansen, 1997). Para avaliar a eficiência dos métodos utilizou-se o caso de teste proposto por Bianchi-Aguiar *et al.* (2011).

Este trabalho está organizado como segue. Na seção 2 apresenta-se uma formulação matemática para o problema. Na seção 3 são apresentados os principais procedimentos heurísticos tomados como base no desenvolvimento deste trabalho. A seção 4 descreve, em detalhes, o procedimento heurístico proposto para a solução do problema. Os resultados computacionais obtidos pela aplicação das metaheurísticas propostas são apresentados e discutidos na seção 5. A última seção conclui o trabalho.

2. Formulação Matemática

Neste trabalho é tratado o problema de coleta de lixo do município de Ponte de Lima, localizado em Portugal, que opera com 5 veículos de diferentes capacidades, há 994 contêineres de lixo distribuídos de forma não uniforme pelo município para serem atendidos e as coletas são feitas em 6 dias da semana. O objetivo central é traçar rotas para cada veículo com diferentes frequências de coleta nos contêineres e distribuição nos dias da semana de forma a realizar tal tarefa com custos mínimos.

Bianchi-Aguiar *et al* (2011) apresenta uma formulação matemática para o problema de coleta de lixo, modelando-o como um problema de roteamento periódico de veículos. A Tabela 1 apresenta os índices, parâmetros e conjuntos do modelo. Assume-se que y_{ir} é a variável que realiza a atribuição do cliente i para a agenda r (recebendo valor 1 se o cliente i é visitado pela agenda r) e x_{ijkl} é a variável que indica se o veículo k visita o cliente j imediatamente após visitar o cliente i no dia l (recebendo valor 1 se a aresta ij estiver na rota). A formulação completa é apresentada pelas equações 2.1 a 2.10.

Tabela 1 – Notação

i, j, h	Cliente
k	Veículo
l	Dia
r	Agenda
L	Horizonte de planejamento (dias)
N	Número de clientes
P	Número de depósitos
K_l	Número de veículos disponíveis no dia l
d_{ij}	Distância entre os clientes i e j (km)
q_i	Quantidade total de lixo do cliente i (Kg)
s_i	Duração do serviço no cliente i (minutos)
t_{ij}	Duração da rota de i a j (minutos)
Q_k	Capacidade do veículo k (Kg)
T_l	Duração máxima da rota no dia l (minutos)
C_i	Número de agendas possíveis para o cliente i
a_{rl}	Constante que indica se o dia l pertence à agenda r
\mathcal{L}	Dias do horizonte de planejamento, $\mathcal{L} = \{1; \dots; L\}$
\mathcal{U}	Locais, $\mathcal{U} = \{v_0; v_1; \dots; v_{N+P}\}$, onde v_0 corresponde à garagem
\mathcal{U}_c	Clientes, $\mathcal{U}_c = \{v_1; v_2; \dots; v_N\}$
\mathcal{U}_p	Locais de descarga (depósitos) $ \mathcal{U}_p = P$, $\mathcal{U}_p = \{v_{N+1}; v_{N+2}; \dots; v_{N+P}\}$
K_l	Veículos disponíveis no dia l , $ K_l = K_l$
C_i	Conjunto de agendas possíveis para o cliente i , $ C_i = c_i$

$$\text{minimizar } \sum_{i \in \mathcal{U}} \sum_{j \in \mathcal{U}, j \neq i} \sum_{l \in \mathcal{L}} \sum_{k \in \mathcal{K}_l} d_{ij} x_{ijkl} \quad (2.1)$$

Sujeito a:

$$\sum_{r \in \mathcal{C}_i} y_{ir} = 1; i \in \mathcal{U}_c \quad (2.2)$$

$$\sum_{j \in \mathcal{U}, j \neq i} \sum_{k \in \mathcal{K}_l} x_{ijkl} - \sum_{r \in \mathcal{C}_i} a_{rl} y_{ir} = 0; i \in \mathcal{U}_c; l \in \mathcal{L} \quad (2.3)$$

$$\sum_{i \in \mathcal{U}, i \neq h} x_{ihkl} - \sum_{j \in \mathcal{U}, j \neq h} x_{hjkl} = 0; h \in \mathcal{U} \setminus \{v_0\}; l \in \mathcal{L}; k \in \mathcal{K}_l \quad (2.4)$$

$$u_{ikl} - u_{jkl} + Q_k x_{ijkl} \leq Q_k - q_j; i, j \in \mathcal{U} \setminus \{v_0\}; j \neq i; q_i + q_j \leq Q_k; l \in \mathcal{L}; k \in \mathcal{K}_l \quad (2.5)$$

$$q_i \leq u_{ikl} \leq Q_k; i \in \mathcal{U} \setminus \{v_0\}; l \in \mathcal{L}; k \in \mathcal{K}_l \quad (2.6)$$

$$\sum_{j \in \mathcal{U} \setminus \{v_0\}} x_{0jkl} \leq 1; l \in \mathcal{L}; k \in \mathcal{K}_l \quad (2.7)$$

$$\sum_{i \in \mathcal{U}} \sum_{j \in \mathcal{U}, j \neq i} (t_{ij} + s_i) x_{ijkl} \leq T_l; l \in \mathcal{L}; k \in \mathcal{K}_l \quad (2.8)$$

$$\sum_{i \in \mathcal{U}_p} x_{i0kl} - \sum_{j \in \mathcal{U} \setminus \{v_0\}} x_{0jkl} = 0; l \in \mathcal{L}; k \in \mathcal{K}_l \quad (2.9)$$

$$x_{ijkl}, y_{ir} \in \{0,1\}; i, j \in \mathcal{U}_c; j \neq i; l \in \mathcal{L}; k \in \mathcal{K}_l; r \in \mathcal{C}_i \quad (2.10)$$

A função objetivo (2.1) minimiza a distância total percorrida pelos veículos em todos os dias da agenda planejada. As restrições (2.2) asseguram que uma agenda adequada é atribuída para cada cliente. As restrições (2.3) garantem que os clientes são visitados somente nos dias correspondentes em suas agendas. As restrições (2.4) garantem que a rota seja gerada adequadamente, assegurando que toda vez que o veículo chega a um determinado cliente, ele em seguida deixa tal cliente. As restrições (2.5) e (2.6) garantem que não haja repetições na rota. As restrições (2.7) asseguram que no máximo, cada veículo seja utilizado uma vez por dia. O limite de tempo para cada dia é assegurado nas restrições (2.8). As restrições (2.9) asseguram que todos os veículos passem por um depósito antes de, finalmente, se dirigirem à garagem. A restrição (2.10) impõe que as variáveis x e y sejam binárias.

3. Metaheurísticas

3.1. Pesquisa em Vizinhança Variável

A metaheurística de Pesquisa em Vizinhança Variável (*Variable Neighborhood Search* ou VNS) é um método de busca local, proposto por Mladenovic e Hansen (1997), que consiste em explorar o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança. Diferentemente de outras metaheurísticas baseadas em métodos de busca local, o método VNS não segue uma trajetória, mas sim explora vizinhanças gradativamente mais “distantes” da solução corrente e focaliza a busca em torno de uma nova solução se, e somente se, um movimento de melhora é realizado. A Figura 2 apresenta o pseudocódigo do VNS.

No VNS, parte-se de uma solução inicial aleatória e a cada iteração seleciona-se aleatoriamente, na vizinhança $N^{(k)}(s)$, um vizinho s' da solução corrente s . O vizinho s' é submetido então a um procedimento de busca local, obtendo-se assim \hat{s} . Se o ótimo local encontrado for melhor que s , então este passa a ser a solução corrente e recomeça-se a busca a partir da primeira estrutura de vizinhança $N^{(1)}(s)$. Se não ocorrer de \hat{s} ser melhor que s a busca continua a partir da vizinhança $N^{(k+1)}(s)$. O VNS será executado até a condição de parada ser atingida.

algoritmo VNS
gere uma solução inicial s
seja k_{max} o número de estruturas diferentes de vizinhança
enquanto (critério de parada não for satisfeito) **faça**
 $k \leftarrow 1$
 enquanto ($k \leq k_{max}$) **faça**
 gere um vizinho s' aleatoriamente da k -ésima vizinhança de s ($s' \in N^k(s)$)
 aplique uma busca local em s' obtendo um ótimo local (\hat{s})
 se ($f(\hat{s}) < f(s)$) **então**
 $s \leftarrow \hat{s}$
 $k \leftarrow 1$
 senão
 $k \leftarrow k + 1$
 fim-enquanto
fim-enquanto
fim-algoritmo

Figura 2 – Pseudocódigo da metaheurística VNS

3.2. Simulated Annealing

A metaheurística *Simulated Annealing* (SA) é um método probabilístico proposto em Kirkpatrick *et al.* (1983), que, assim como o VNS, explora o espaço de soluções através de soluções vizinhas. As soluções vizinhas são escolhidas utilizando estruturas de vizinhança, que são escolhidas aleatoriamente e, sempre que há melhora, o método continua a busca a partir da nova solução. Caso não haja melhora, existe certa probabilidade de que a nova solução (mesmo que pior) seja escolhida como solução corrente, que é menor quanto menor for a temperatura corrente. A Figura 3 apresenta o pseudocódigo do SA.

algoritmo SA (T_0, SA_{max}, α)
gere uma solução inicial s
 $IterT \leftarrow 0$
 $T \leftarrow T_0$
enquanto (critério de parada não for satisfeito) **faça**
 enquanto ($IterT < SA_{max}$) **faça**
 $IterT \leftarrow IterT + 1$
 gere um vizinho s' aleatoriamente ($s' \in N(s)$)
 calcule $\Delta = f(s') - f(s)$
 se ($\Delta \leq 0$) **então**
 $s \leftarrow s'$
 senão
 $s \leftarrow s'$ com probabilidade $e^{-\Delta/T}$
 fim-enquanto
 $T \leftarrow \alpha \times T$
 $IterT \leftarrow 0$
fim-enquanto
fim-algoritmo

Figura 3: Pseudocódigo da meta-heurística *Simulated Annealing*

No SA, são definidas inicialmente uma temperatura inicial e uma solução de partida. A cada iteração, é gerada uma nova solução vizinha, por meio de uma estrutura de vizinhança

qualquer. A nova solução é então avaliada e, caso for melhor que a solução corrente, continua a busca a partir dela. Caso a nova solução seja inferior à anterior, a nova solução pode ou não substituir a anterior, isso ocorrendo com probabilidade $e^{-\Delta T}$. A temperatura é então decrementada (de acordo com o parâmetro $\alpha < 1$), até que se atinja uma temperatura muito baixa (por exemplo $T < 0,0001$).

4. Implementação

Uma solução para o PRPV aplicado à coleta de lixo é representada através de uma matriz. Cada linha da matriz representa um dia da semana. Em cada dia é determinado a rota para cada veículo, sendo que o veículo é representado por um número com sinal negativo e sua sequência de visitas vem logo em seguida até encontrar outro veículo ou chegar ao fim da linha. A representação inclui também um vetor, indicando a agenda com que cada cliente é atendido, sendo as agendas 1-6 para frequências de um dia, 7-11 para dois dias, 12-15 para três dias, 16-18 para 4 dias, e a 19 para 6 dias. A Figura 3 apresenta um exemplo de solução para o PRPV com dois veículos, cinco clientes e um período de programação de seis dias. Um exemplo de rota pode ser visto no dia 2 para o veículo (-1) com a visita dos 3, 5 e 1 na sequência, sendo que, o veículo se desloca da garagem até o cliente 3 e após o cliente 1 ser visitado o veículo se desloca para o depósito mais próximo e em seguida retorna para a garagem.

DIA	ROTA					
1	-1	2	4	-2	1	3
2	-1	3	5	1	-2	
3	-1	1	4	-2		
4	-1	-2	1	4		
5	-1	2	3	-2	1	
6	-1	2	3	1	-2	4

Cientes	c1	c2	c3	c4	c5
Frequência	6	3	4	4	1
Agenda	19	13	17	18	2

Figura 3: Exemplo de representação de uma solução

A função objetiva do PRPV (equação 4.1) é a soma total das distâncias de todas as rotas, ou seja, as distâncias que cada veículo em cada dia percorre ao sair da garagem, visitar os clientes na sequência determinada, ir até um dos depósitos e retornar novamente à garagem. No algoritmo são determinadas penalidades (δ e β) que são somadas às soluções que não satisfazem alguma das restrições impostas pelo problema (peso além da capacidade de cada veículo ($\delta * Ep$) ou tempo total da rota acima do permitido ($\beta * Et$)). Representa-se a distância inicial por D_i , a distância do cliente i ao cliente seguinte j por D_{ij} e a distância de retorno por D_r .

$$\sum_i (D_i + D_{ij} + D_r + (\delta * Ep) + (\beta * Et)) \tag{4.1}$$

Seja s uma solução corrente para o problema, para gerar s' , uma solução vizinha de s , usa-se quatro movimentos para definir diferentes estruturas de vizinhança, as quais serão nomeadas de N^1 , N^2 , N^3 e N^4 .

O primeiro movimento (N^1) consiste em gerar uma solução vizinha trocando a frequência de um cliente, de modo aleatório. A Figura 4 exibe um exemplo de tal estrutura, trocando a frequência do cliente 5 do dia 2 para o dia 4.

DIA	ROTA					
1	-1	2	4	-2	1	3
2	-1	3	5	1	-2	
3	-1	1	4	-2		
4	-1	-2	1	4		
5	-1	2	3	-2	1	
6	-1	2	3	1	-2	4

⇒

DIA	ROTA					
1	-1	2	4	-2	1	3
2	-1	3	1	-2		
3	-1	1	4	-2		
4	-1	-2	1	5	4	
5	-1	2	3	-2	1	
6	-1	2	3	1	-2	4

Figura 4: Estrutura de vizinhança N^1

No segundo movimento (N^2) escolhe-se aleatoriamente um dia, um caminhão e dois clientes, e realiza-se a troca dos dois clientes. Tal estrutura é exemplificada na Figura 5.

DIA	ROTA					
1	-1	2	4	-2	1	3
2	-1	3	5	1	-2	
3	-1	1	4	-2		
4	-1	-2	1	4		
5	-1	2	3	-2	1	
6	-1	2	3	1	-2	4

⇒

DIA	ROTA					
1	-1	4	2	-2	1	3
2	-1	3	5	1	-2	
3	-1	1	4	-2		
4	-1	-2	1	4		
5	-1	2	3	-2	1	
6	-1	2	3	1	-2	4

Figura 5: Estrutura de vizinhança N^2

No terceiro movimento (N^3) escolhe-se dois veículos distintos e realiza-se a permutação de dois clientes entre os veículos, como é mostrado na Figura 6.

DIA	ROTA					
1	-1	2	4	-2	1	3
2	-1	3	5	1	-2	
3	-1	1	4	-2		
4	-1	-2	1	4		
5	-1	2	3	-2	1	
6	-1	2	3	1	-2	4

⇒

DIA	ROTA					
1	-1	2	4	-2	1	3
2	-1	3	5	1	-2	
3	-1	1	4	-2		
4	-1	-2	1	4		
5	-1	2	3	-2	1	
6	-1	4	3	1	-2	2

Figura 6: Estrutura de vizinhança N^3

A última estrutura de vizinhança (N^4) consiste em escolher um cliente em determinado dia, e trocá-lo de veículo, como mostrado na Figura 7.

DIA	ROTA					
1	-1	2	4	-2	1	3
2	-1	3	5	1	-2	
3	-1	1	4	-2		
4	-1	-2	1	4		
5	-1	2	3	-2	1	
6	-1	2	3	1	-2	4

⇒

DIA	ROTA					
1	-1	2	4	-2	1	3
2	-1	3	5	1	-2	
3	-1	1	4	-2		
4	-1	-2	1	4		
5	-1	3	-2	2	1	
6	-1	2	3	1	-2	4

Figura 7: Estrutura de vizinhança N^4

4.1. SA aplicado ao problema de Roteamento Periódico de Veículos para coleta de lixo

O algoritmo *Simulated Annealing* (SA) implementado para o problema de Roteamento Periódico de Veículos para coleta de lixo inicia-se de uma solução inicial aleatória e segue de acordo com o esquema do SA original. Dada uma temperatura T , o algoritmo seleciona aleatoriamente um dos movimentos de vizinhança e então calcula a variação na função objetivo. Se a solução calculada for melhor do que a anterior, ela é aceita como nova solução. Caso

contrário, há certa probabilidade, que decresce juntamente com T , de que a nova solução seja aceita.

As quatro estruturas de vizinhanças aleatórias definidas na seção anterior são utilizadas para gerar soluções vizinhas à solução s . A cada iteração seleciona-se uma estrutura aleatoriamente com probabilidade uniforme.

Os parâmetros de controle do SA são a taxa de resfriamento α , o número de iterações a cada temperatura (SA_{max}) e a temperatura inicial T_0 . Neste trabalho utilizou-se $\alpha = 0.95$, $SA_{max} = 1000$ e $T_0 = 1000000$. As penalidades utilizadas foram $\delta = 100$ e $\beta = 100$.

4.2. VNS aplicado ao problema de Roteamento Periódico de Veículos para coleta de lixo

O VNS utiliza os movimentos N^1 , N^2 , N^3 e N^4 e cria-se 12 estruturas de vizinhança aninhadas. No primeiro grupo (1-4) realiza-se um movimento aleatoriamente, no segundo grupo (5-8) três movimentos e no terceiro grupo (9-12) realiza-se seis movimentos.

O VNS utiliza uma heurística de busca local que consiste em analisar as trocas de pares de clientes atendidos em um mesmo dia, modificando as rotas dos veículos. Foram estudadas as estratégias de busca *firstimprovement* e *bestimprovement*, que se diferem pela forma como conduzem a busca. Na *firstimprovement* a busca continua a partir da nova solução sempre que esta for melhor que a solução corrente. Enquanto que na *bestimprovement*, a solução corrente é atualizada somente após a busca ter realizado todos os movimentos, caso a melhor solução encontrada seja melhor que a solução corrente. Para o VNS proposto, a busca local com a estratégia *firstimprovement* encontrou resultados melhores.

5. Resultados Computacionais

As metaheurísticas VNS e SA foram codificadas na linguagem C/C++, utilizando a plataforma CodeBlocks. Os testes computacionais foram executados em um PC Intel Core i5 de 2.67 GHz com 2 GB de memória RAM.

Nos testes utiliza-se a instância proposta por Bianchi-Aguiar *et al.* (2011), sendo a representação real da coleta de lixo do município de Ponte de Lima em Portugal. Esta instância contém: cinco veículos, cada qual com sua capacidade e tempo máximo previsto para utilização; uma garagem, de onde partem os veículos para seguirem suas rotas e depois depositarem o lixo no depósito que estiver mais próximo; dois depósitos, a serem visitados no final da rota e antes dos veículos retornarem à garagem; e 51 clientes, cada um com sua exigência de quantidade de lixo a ser recolhida e frequência específicas, os quais serão visitados no máximo seis dias na semana.

Os resultados computacionais obtidos pelas metaheurísticas são apresentados na Tabela 2, que possui as seguintes colunas: nome da instância testada, valor em quilômetros da solução utilizada na prática atualmente, a melhor solução encontrada (coluna Melhor), solução média (coluna Média), o desvio percentual (coluna Dev) e o tempo computacional (coluna Tempo) em segundos. O desvio percentual é calculado como $Dev = 100\% (Média - Melhor) / Melhor$. O VNS e o SA foram executados 20 vezes cada.

Tabela 2 – Resultados computacionais das metaheurística SA e VNS.

Instância	Solução Atual	SA				VNS			
		Melhor	Média	Dev (%)	Tempo (s)	Melhor	Média	Dev (%)	Tempo (s)
Ponte de Lima	2389	1633	1728,3	5,84	1000,00	1705	1808,5	6,07	918,00

Observa-se na Tabela 2 que a metaheurística SA obteve a melhor solução para o problema de coleta de lixo, reduzindo a distância percorrida em 31% em relação à solução atual adotada no município. Além disso, cabe ressaltar que o número de rotas também foi reduzido, na

solução atual são necessárias 26 rotas enquanto que na solução apresentada pelo SA existem 25 rotas. A solução do VNS também apresenta uma redução considerável na distância percorrida (28%) em relação à solução atual. Porém a melhor solução obtida pelo VNS foi 4,4% pior que a melhor solução encontrada pelo SA. Ambos os métodos se mostraram robustos, obtendo soluções médias próximas das melhores soluções de acordo com o desvio percentual (5,84% para o SA e 6,07% para o VNS). Os tempos computacionais dos algoritmos foram aceitáveis, obtendo boas soluções em poucos minutos de execução.

6. Conclusão

Neste trabalho foram propostos algoritmos baseados nas metaheurísticas *Simulated Annealing* (SA) e Pesquisa em Vizinhança Variável (VNS) para resolver o problema de roteamento periódico de veículos (PRPV) para a coleta de lixo. Os algoritmos apresentados iniciam a busca utilizando uma solução inicial aleatória. As duas abordagens se diferem no fato de que no VNS são exploradas vizinhanças gradativamente mais “distantes” da solução corrente e é focalizada a busca em torno de uma nova solução se, e somente se, um movimento de melhora é realizado. Enquanto no SA, o algoritmo seleciona aleatoriamente um dos movimentos de vizinhança e então calcula a variação na função objetivo. Se a solução vizinha for melhor do que a solução corrente, ela é aceita como nova solução corrente. Caso contrário, há uma probabilidade de que a nova solução seja aceita.

As metaheurística SA e VNS se mostraram eficientes, obtendo soluções melhores que a solução atual praticada no município de Ponte de Lima (apresentada em Bianchi-Aguiar (2011)). Os algoritmos também foram robustos, tendo pequenos desvios percentuais entre as melhores soluções e as soluções médias, e sendo a busca realizada em poucos minutos de execução.

Como trabalhos futuros, propõe-se implementar o método *Path-Relinking* (Glover, 1986) e o método híbrido Busca por Agrupamentos (CS, do inglês *Clustering Search*) (Chaves e Lorena, 2010), visando encontrar melhoras soluções. Além disso, a realização de testes com outras instâncias também será objeto de investigação futura. Atualmente algumas cidades brasileiras adotam uma política de coleta de lixo semelhante à apresentada no município de Ponte de Lima, tais como Porto Alegre/RS e Itú/SP.

Agradecimentos

Os autores agradecem ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) pelo suporte financeiro para a realização deste trabalho.

Referências Bibliográficas

Bianchi-Aguiar, T.; Carravilla, M. A.; Oliveira, J. F. (2011), Municipal wastecollection in Ponte de Lima, Portugal – A vehicleroutingapplication. *OR Insight*. doi:10.1057/ori.2011.23

Chaves, A.A.; Lorena, L.A.N. Clustering search algorithm for the capacitated centeredclustering problem. *Computers & Operations Research*, v. 37, p. 552-558, 2010.

Christofides, N.; Beasley, J. E. (1984), The Period Routing Problem, *Networks* 14, 237 – 256.

Christofides, N; Mingozzi, A.; Toth, P. (1979), The vehicle routing problem. In Christofides, Mingozzi, Toth, Sandi, editors. *Combinatorial Optimization*, 315-338, John Wiley & Sons.

Carvalho, D. B.; Oliveira, G. A.; Souza, M. J. F. (2003), Método de pesquisa em vizinhança variável aplicado à resolução do problema de roteamento de veículos, *XXXV SBPO*, 676-684.

Coelho, M. A.; Soares, L. T. (2001), Geografia geral: o espaço natural e socioeconômico. Moderna, 4ª ed., 427.

Cunha, V.; Caixeta F.J.V. (2002), Gerenciamento da Coleta de Resíduos Sólidos Urbanos: Estruturação e Aplicação de Modelo Não-Linear de Programação por Metas, *Gestão & Produção*, 9(2):143–161.

Feo, T. A.; Resende, M. G. C. (1995), Greedy randomized adaptive search procedures. *Journal of Global Optimization*, v. 6, 109-133.

Glover, F. (1986), Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 5, 553-549.

Golden, B. L.; Chao, I. M.; Wasil, E. (1995), An Improved Heuristic for the Period Vehicle Routing Problem. *Networks*, 25-44.

Gonçalves, L. B.; Ochi, L. S.; Martins, S. L. (2005), Heurísticas GRASP para um Problema de Roteamento Periódico de Veículos. Disponível em: www.ic.uff.br/PosGraduacao/Dissertacoes/274.pdf. Acesso: 29/04/2012.

Holland, J. H. (1975), *Adaptation in natural and artificial systems*. Michigan: University of Michigan Press, 211.

Kirkpatrick, S.; Gellat, D. C.; Vecchi, M. P. (1983), *Optimization by simulated annealing*. *Science*, v. 220, n. 4598, 671-680.

Mladenovic, N.; Hansen, P. (1997), *Variable Neighborhood Search*. *Computers and Operations Research*, 1097–1100.

Toth, P.; Vigo, D. (2002), *The vehicle routing problem*. Society for Industrial and Applied Mathematics. 367 pp.

Xavier, R. S.; Lisboa, A. C.; Vieira, D. A. G.; Saldanha, R. R. (2010), Heurística para modelagem e minimização do consumo de combustível para rotas de coleta de lixo. Disponível em: www.sobrapo.org.br/sbpo2010/xliisbpo_pdf/72338.pdf. Acesso: 29/04/2012.