

PROBLEMA DE ROTEAMENTO DE VEÍCULOS USANDO OTIMIZAÇÃO DE COLÔNIA DE FORMIGAS PARA COLETA SELETIVA – ESTUDO DE CASO**Pierluigi Mario Michelino de Palma**Universidade Estadual Paulista – UNESP, campus de Sorocaba – SP
Av. Três de Março, 511, Sorocaba – SP, 18087-180
pierluigi.mpalma@gmail.com**Luiza Amalia Pinto Cantão**Universidade Estadual Paulista – UNESP, campus de Sorocaba – SP
Av. Três de Março, 511, Sorocaba – SP, 18087-180
luiza@sorocaba.unesp.br**RESUMO**

O Problema de Roteamento de Veículos (PRV), em inglês *Vehicle Routing Problem – VRP*, é um problema de otimização combinatória, onde temos por objetivo determinar a melhor rota possível que os veículos possam cumprir entre um depósito e um conjunto de consumidores. Os PRV foram estudados pela primeira vez por Dantzig e Ramser em 1959 e, apesar da fácil compreensão do problema em si, apresentam elevada complexidade computacional. Para a resolução do PRV podem ser usadas várias heurísticas diferentes, métodos matemáticos que buscam aproximar, a cada iteração, a melhor solução possível. Nesse caso, utilizamos a heurística baseada no comportamento probabilístico de formigas, Otimização por Colônia de Formigas (*Ant Colony Optimization – ACO*), afim de encontrarmos uma rota otimizada para um estudo de caso de coleta seletiva na cidade de Sorocaba – SP.

PALAVRAS CHAVE: Problema de Roteamento de Veículos, Otimização por Colônia de Formigas, Coleta Seletiva.

Área Principal: L&T – Logística e Transporte

ABSTRACT

The Vehicle Routing Problem (VRP) is a combinatorial optimization problem, where we aim to determine the best possible route that vehicles can meet between a deposit and a set of consumers. The VRP were first studied by Dantzig and Ramser in 1959 and, despite the easy understanding of the problem itself, have high computational complexity. For the resolution of the VRP could use several different heuristics, mathematical methods seeking approach, at each iteration, the best solution possible. In this case, it will be used a heuristic based on probabilistic behavior of ants, Ants Colony Optimization *ACO*, in order to find a VRP solution. Thus, this study presents the possible results through a routing on it's initial.

KEYWORDS: Vehicle Routing Problem, Ant Colony Optimization, solid waste collection.

Main Area: L&T – Logistics and Transportation

1 Introdução

O Problema de Roteamento de Veículos (PRV), é definido por Toth (2001) como um problema de logística onde existe um conjunto de consumidores (ou cidades, nós) cada qual necessita de uma quantidade $q_i, i = 1, \dots, n$ de um produto no nó i , e um depósito (nó $i = 0$) com veículos de capacidade Q_j . O propósito é encontrar a melhor rota possível entre os consumidores, minimizando os custos de transporte. Algumas restrições básicas do problema são:

- Toda rota começa e termina no depósito;
- Todo nó é visitado somente uma vez.

O PRV vem sendo utilizado nas últimas décadas em função da tentativa de diminuir os gastos em toda a cadeia de produção, até o consumidor. Essencialmente essa tipologia de problema se resume ao atendimento de uma demanda, sob forma de coleta ou entrega, ao longo de uma determinada região geográfica. Considerado um problema de otimização combinatória, foi estudado pela primeira vez por Dantzig (1959), onde abordava a distribuição e preço de combustível. Devido a sua complexidade combinatória, descrita por Goldberg (2000), poucos PRV podem ser resolvidos de maneira exata. Precisa-se recorrer a heurísticas, métodos matemáticos que, iteração após iteração, sempre retornam o melhor caminho encontrado. Existem diferentes heurísticas, como aquelas apresentadas por Toth (2001), mas aplicaremos a (*Ant Colony Optimization* – ACO), Otimização por Colônia de Formigas, amplamente discutida por Dorigo (2004).

Otimização por colônia de formigas é um algoritmo introduzido por Dorigo (1992) para o Problema do Caxeiro Viajante (*Traveling Salesman Problem* – TSP). Esse algoritmo é baseado no comportamento probabilístico das formigas para encontrar a melhor rota possível para o TSP, posteriormente este algoritmo foi adaptado para o PRV, como apresenta Dorigo (2004).

Para o nosso estudo de caso, Simões (2011) afirma que na cidade de Sorocaba – SP, juntaram-se três cooperativas de reciclagem de lixo (Reviver, Coeso e Catares) em um único depósito, para aumentar a abrangência e usufruir melhor dos recursos da prefeitura. Localizada na zona Leste da cidade, a Reviver possui uma frota de 6 caminhões, com capacidade de até 3 toneladas cada, coletam semanalmente em todas as regiões de Sorocaba seguindo uma rota específica. Essas rotas utilizadas pelos caminhões não possuem nenhuma estratégia de planejamento. Ela é fruto da experiência dos operadores e dos testes feitos por eles. O intuito de aplicar a modelagem de PRV nessa situação é a de melhorar a etapa da coleta para que todo o resto do sistema de reciclagem possa ser beneficiado. Esse estudo tem por objetivo proporcionar uma rota otimizada em relação a minimização da distância percorrida, aumentando a eficiência desta etapa do processo.

2 O Problema

Para estudarmos o PRV para a coleta seletiva da cidade de Sorocaba – SP, tomamos por base o seguinte problema de otimização:

$$(PRVJT) \min \sum_{k \in K} \sum_{(i,j) \in E} c_{ij} x_{ijk}, \quad (1a)$$

$$\sum_{k \in K} \sum_{j \in N} x_{ijk} = 1 \quad \forall i \in C, \quad (1b)$$

$$\sum_{j \in N} x_{0jk} = 1 \quad \forall k \in K \quad (1c)$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0 \quad \forall k \in K, \forall h \in C, \quad (1d)$$

$$\sum_{i \in N} x_{i,n+1,k} = 1 \quad \forall k \in K, \quad (1e)$$

$$x_{ijk}(s_{ik} + t_{ij} - s_{jk}) \leq 0 \quad \forall k \in K, \forall (i, j) \in E, \quad (1f)$$

$$\sum_{i \in C} d_i \sum_{j \in N} x_{i,j,k} \leq Q \quad \forall k \in K, \quad (1g)$$

$$x_{ijk} \geq 0 \quad \forall k \in K, (i, j) \in A, \quad (1h)$$

$$x_{ijk} \in \{0, 1\} \quad \forall k \in K, (i, j) \in A, \quad (1i)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \quad S \subset C, 2 \leq |S| \leq \left\lfloor \frac{n}{2} \right\rfloor, \forall k \in K, \quad (1j)$$

Onde a função (1a) expressa a minimização do custo total das rotas. A restrição (1b) assegura que cada cliente i seja designado para um único veículo. Sucessivamente, das restrição (1c) à (1e) determina que cada caminhão saia sempre do depósito (nó 0) somente uma vez, que se deixe um nó h somente se tenhamos entrado nele e que se retorne ao depósito (nó $n + 1$) somente uma vez. A restrição (1f) indica que um veículo k , após ter deixado o nó i viajando para o nó j , não chegue em j antes de $s_{ik} + t_{ij}$, onde s_{ik} é o instante em que o veículo k começa a servir o nó i e t_{ij} é o tempo de coleta estimado em cada nó. A restrição (1g) impõem que a demanda total de cada rota do veículo k não seja maior que a capacidade Q do veículo. A restrição (1h) delimita que a variável x_{ijk} seja sempre um valor positivo. E por fim, a condição (1i) e a condição (1j) impõe que o problema seja articulado com variáveis binárias e que não existam sub-rotas, respectivamente.

3 Otimização por Colônia de Formigas

As metaheurísticas são algoritmos matemáticos aplicados em diferentes problemas e instâncias, entre eles o PRV para atingir uma solução mais próxima da ótima. Vale ressaltar que dificilmente a aplicação de um desses algoritmos obterá a melhor solução para o problema. Eles aproximam para o melhor valor possível, mas dificilmente encontram a melhor solução.

Nesse estudo de caso usamos a metaheurística por Colônia de Formigas, introduzida por Dorigo (1992) e descrita amplamente por Dorigo (2004), utilizando a comparação entre formigas–formigueiro–comida para resolver o PRV. O formigueiro é onde todas as formigas iniciam a rota em busca de alimento. Todas as formigas saem do formigueiro e começam a procurar comida seguindo um caminho aleatoriamente, sem ter uma meta preestabelecida, deixando atrás de si uma trilha de feromônio, hormônio animal. O feromônio evapora com o tempo, com isso, a menor rota até encontrar o alimento, ou seja aquela que pode ser percorrida com o menor tempo possui maior taxa de feromônio, pois sendo menor, o feromônio não evaporou totalmente. Após as formigas terem tomados caminhos aleatórios até conseguir chegar na fonte de alimento, a trilha mais curta entre as que foram percorridas, será seguida por outras formigas, pois é a que possui maior taxa de feromônio não evaporado, e será marcada por uma quantia maior de feromônio, fazendo dela o melhor percurso conhecido até aquele momento até a comida. Sucessivamente outras formigas podem tomar novos percursos aleatórios, achando ou não um melhor percurso até o alimento. Por isso, essa metaheurística é

considerada iterativa, pois a cada iteração, lançando novas formigas para a busca de alimento, pode-se encontrar novos caminhos mais curtos, fazendo com que o antigo caminho mais curto seja descartado pelas outras formigas, e fazendo do novo o caminho o mais utilizado. Antes de descrever o algoritmo usado, devemos citar as equações utilizadas, extraídas de Castro (2012).

A equação (2) gers a evaporação do feromônio, como segue.

$$\tau_{ij}(t+1) \leftarrow (1-\rho)\tau_{ij}(t) + \Delta\tau(t) \quad (2)$$

onde a variável τ_{ij} denominada de *trilha de feromônio artificial*, associada a conexão entre os nós i e j . A variável ρ é um número ($\rho \in [0, 1)$) que determina a taxa de decaimento do feromônio e o $\Delta\tau(t)$ é a variação de feromônio adicionada pela formiga naquele percurso.

Outra função, relacionada dessa vez à distância entre cidades, denominada *visibilidade*, é apresentada pela equação (3).

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (3)$$

A equação (3), onde d_{ij} é a distância de um nó i até um no j , será útil para a composição da próxima equação (4), onde é calculada a probabilidade de uma formiga k ir de um nó i para um nó j na iteração t do nosso algoritmo.

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}, & \text{se } j \in J_i^k \\ 0 & \text{caso contrário} \end{cases} \quad (4)$$

onde τ_{ij} é o nível de feromônio no percurso entre i e j , η_{ij} representa a visibilidade entre os nós i e j . Os expoentes α e β , inseridos pelo usuário, definem a relevância de cada uma das partes da equação. Podemos notar que o α está relacionado com a influência do feromônio, enquanto que β influi sobre a importância da visibilidade. Por exemplo, se $\alpha = 0$, cidades mais próximas tenderão a ser escolhidas, enquanto com $\beta = 0$, será considerada apenas a amplificação na quantidade do feromônio. Determinar de maneira correta esses parâmetros é fundamental, pois se errados, podem fazer o algoritmo estagnar e se viciar em uma rota que não seja a melhor possível.

Como última equação, temos aquela que determina a quantidade de feromônio liberada, explicitada por (5).

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)}, & \text{se } (i, j) \in T^k(t) \\ 0 & \text{caso contrário} \end{cases} \quad (5)$$

sendo $L^k(t)$ o comprimento da rota T^k seguida por uma formiga k na iteração t , e Q representa uma constante a ser depositada na trilha para melhorar o algoritmo, definida pelo usuário.

Assim sendo, o termo $\Delta\tau_{ij}(t)$ da equação (2) é substituído por $\Delta\tau_{ij}(t) = \sum_k \Delta\tau_{ij}^k(t)$, onde $\Delta\tau_{ij}^k(t)$ é especificado pela equação (5) e $k = 1, \dots, N$ é o índice das formigas.

Finalmente, de Castro (2012), aconselha usar os seguintes valores para os parâmetros das equações acima: $\alpha = 1$, $\beta = 5$, $\rho = 0.5$, $N = e$, $Q = 100$ e $\tau_0 = 10^{-6}$.

Após a definição das funções e parâmetros, apresentamos o pseudo-algoritmo (1), que descreve o processo da Otimização por Colônia de Formigas.

No nosso estudo de caso, o algoritmo de ACO será utilizado para encontrar a melhor rota possível para a coleta seletiva da cooperativa Reviver.

O depósito da cooperativa será nosso formigueiro, onde todos os caminhões devem voltar para descarregar o material coletado, uma vez que estiverem com a carga completa, atingido a janela de tempo de serviço dos cooperados ou simplesmente tiverem cumprido toda a rota pre-estabelecida. Os caminhões sairão do depósito cumprindo uma rota determinada entre os pon-

Algorithm 1: Otimização por Colônia de Formigas.

```

1 Inicialização dos Parâmetros
2 %%O primeiro valor para o feromônio ( $\tau_0$ ) é determinado pelo usuário, para podermos
  comparar com os novos e manter a melhor solução%%
3  $t \leftarrow 1$  %% Inicializa a variável que determina o número de iterações%%
4 while não terminaram-se as iterações do
5   for todas as formigas  $k$  utilizadas do
6     Construa um tour  $T^k(t)$  entre todos os nós, partindo do nó inicial, escolhendo a
     próxima cidade baseado na equação (4);
7   end for
8   if o tour encontrado por uma formiga for menor que  $L_{best}$  then
9     Atualize o melhor tour  $L_{best}$  já encontrado;
10  end if
11  for todos os nós do
12    Atualize o feromônio em todos os caminhos, usando a equação (2);
13    Atualize a equação (5) com o novo parâmetro  $L^k(t)$  encontrado;
14  end for
15  Passe para a próxima iteração( $t \leftarrow t + 1$ );
16 end while

```

tos de coleta, que serão agrupados em nós para facilitar a abordagem matemática. Com isso, na primeira iteração do algoritmo, cada um dos seis caminhões executará uma rota aleatória, cumprindo todas as restrições indicadas anteriormente, deixando ao longo do percurso que cada caminhão executará, uma trilha de feromônio que ao longo do tempo evaporará, assim como acontece com as formigas. Na segunda iteração, os caminhões não percorrerão uma trilha aleatória novamente, mas usarão as trilhas de feromônio deixadas pelos caminhões na iteração precedente, como indicação para escolher os nós a ser seguidos, um após o outro. Repetindo esse processo por um número relativamente alto de vezes, sempre seguindo as trilhas de feromônio deixadas pelos caminhões nas iterações precedentes, os mesmos terão evidenciado, com uma alta taxa de feromônio, a melhor rota encontrada, pois ela foi seguida iteração após iteração.

No algoritmo existem índices que determinam a relevância dada à evaporação do feromônio e a distância até o próximo nó, evitando que o algoritmo caia em casos de estagnação da solução, fazendo com que a solução encontrada não seja tão próxima da melhor solução possível real.

Além disso, no algoritmo é preciso adicionar rotinas de tratamento da capacidade de carga dos caminhões, fazendo com que o caminhão não vá para nós que possam ultrapassar a sua capacidade máxima de carga, definida em (1g).

A seguir apresentamos o Algoritmo 2 com as modificações apropriadas.

4 Dados do Problema

Nesta seção apresentaremos as informações utilizadas para a organização dos dados e os métodos com que os dados fornecidos foram trabalhados. Em relação aos dados, temos a maioria deles fornecidos pela cooperativa Reviver (planilhas, bairros abrangidos, horários) e a elaboração dos nós foi obtida através do software *Google Earth (Versão 7.0.3.8542)*. A escolha do *Google Earth* basea-se no uso deste software em trabalhos já publicados, como o de Rodrigues (2008) e de Tunjongsirigul (2012).

Em relação aos métodos utilizados para elaborar os dados, como software secundário para

Algorithm 2: Otimização por Colônia de Formigas para PRV com Janelas de Tempo – PRVCJT

```

1 Inicialização dos Parâmetros
2 %%O primeiro valor para o feromônio ( $\tau_0$ ) é determinado pelo usuário, para podermos
   comparar com os novos e manter a melhor solução%%
3  $t \leftarrow 1$  %% Inicializar a variável que determina o número de iterações%%
4 while não terminarem as iterações do
5     for todas as formigas  $k$  utilizadas do
6         Construa um tour  $T^k(t)$  entre todos os nós, partindo do nó inicial, escolhendo a
           próxima cidade baseado na equação (4);
7     end for
8     if (o tour encontrado por uma formiga for menor que  $L_{best}$ ) e (respeitarem os
           limites de tempo) e (a capacidade não for excedida) then
9     end if
10    Atualize o melhor tour  $L_{best}$  já encontrado; for todos os nós do
11        Atualize o feromônio em todos os caminhos, usando a equação (2);
12        Atualize a equação (5) com o novo parâmetro  $L^k(t)$  encontrado;
13    end for
14    Passe para a próxima iteração( $t \leftarrow t + 1$ );
15 end while

```

o processamento das imagens geradas no *Google Earth* usou-se o processador *ImageJ*, programado em java. Escolheu-se usar esse processador e analisador de imagens seguindo as publicações de Abramoff (2004) e de Tajimaa (2011). Como software principal para encontrar a melhor rota usamos *ACOTSP, Versão 1.02* elaborado pela equipe e usado pela primeira vez em Dorigo(1992), sucessivamente utilizado também por Manfrin (2006), por Lopez-Ibáñez (2011) e por Twomey (2010), entre outros.

Assim, os dados dos problemas e ferramentas usadas foram:

- Locais de coleta (logradouros e numero de casas), fornecidos pela cooperativa numa planilha;
- Localização geográfica dos locais de coleta (coordenadas LAT/LONG);
- Características dos caminhões: volume máximo de transporte e velocidade de coleta;
- Características da cooperativa: horários de trabalho dos cooperados;
- Ferramenta cartográfica: *Google Earth*;
- Distância entre os nós: *Google Earth*;
- Processamento das imagens geradas: *ImageJ*;
- Elaboração e processamentos dos dados: *AcoTsp* desenvolvido por Dorigo (1992).

A partir das planilhas fornecidas pela cooperativa Reviver, usando o *Google Earth*, encontramos todos os logradouros possíveis onde os caminhões da cooperativa coletam materiais recicláveis. Dividimos em cores as ruas abrangidas por cada caminhão, para ter uma ideia melhor de onde realmente os caminhões fazem as coletas, para determinarmos onde cada caminhão trafega, e conseqüentemente ter informação sobre quanto a rota da cooperativa será otimizada. Caminhões na mesma região indicam pouca otimização, pois diferentes caminhões trafegam pela mesma área. A Figura 1 mostra as áreas atendidas pelos diferentes caminhões.

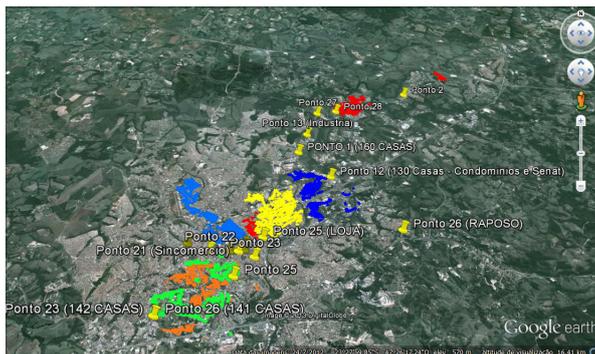


Figura 1: Marcação das rotas varridas pelos caminhões.

Cor	Caminhão	Cor	Caminhão	Cor	Caminhão
Amarelo	Caminhão 1	Vermelho	Caminhão 3	Verde	Caminhão 5
Azul	Caminhão 2	Laranja	Caminhão 4	Anil	Caminhão 6

Tabela 1: Legenda de cores para a Figura 1.

As rotas foram determinadas com o primeiro e o último número de casa habitada presente na rua. Indústrias, condomínio ou lojas foram estipulados como 20 pontos de coleta, devido ao volume de material reciclável gerado. Algumas vezes precisou-se usar o “Street View”, ferramenta do *Google Maps* para determinar ruas que não encontramos ou para esclarecer divergências em relação ao número das habitações.

4.1 Detalhamento dos nós

Após ter repassado todas as ruas varridas pelos caminhões para o *Google Earth*, começamos, de maneira empírica, a agrupar os pontos de coletas próximos para a determinação dos nós. Dessa maneira determinamos 74 nós, incluído o nó 0 (cooperativa). Sempre usando a ferramenta *Google Earth*, desenhamos polígonos para determinar as áreas dos nós.

Os critérios utilizados para a determinação desses nós foram:

- Quantidade de casas à serem visitadas no nó;
- Proximidade à outros pontos de coleta;
- Atenção especial para: condomínios e indústria devido ao grande volume de material reciclável produzido;
- Estimativa de tempo para percorrer o nó inteiro.

A Figura 2 ilustra este processo.

A determinação das coordenadas pelo *Google Earth* permite continuarmos de duas maneiras o roteamento. Ou linearizando as coordenadas terrestres encontradas transformando-as em UTM (*Universal Transverse Mercator coordinate system*), ou através de um software de processamento de imagem. Com os conhecimentos de Cálculo Diferencial e Integral, e baseando-se em Palma (2009), determinou-se mais vantajoso utilizar a linearização. As imagens geradas pelo *Google Earth* possuem resolução elevada (3192 x 4264 pixels) e com isso, a precisão do processamento da imagem resultante também é elevada. Medimos usando a ferramenta de régua do *Google Earth*, a imagem resultante tem 20 x 20 km de comprimento, portanto uma



Figura 2: Nós do PRV.

precisão de:

$$\text{Precisão} = \frac{20000 \text{ metros}}{3192 \text{ pixels}} = 6.3 \text{ metros por pixel}$$

Com isso, através de uma ferramenta própria do *ImageJ*, conseguimos determinar todos os pontos cartesianos definidos na Figura 2. Com estes pontos definidos podemos usar o *ACOTSP*, pois este software trabalha com pontos em coordenadas cartesianas.

5 Resultados

Segundo dados fornecidos pela cooperativa Reviver, semanalmente os caminhões atendem 837 ruas, com 11979 pontos de coleta. O centro de reciclagem situado no bairro Zulmira, Zona Leste da cidade, nos mostra a necessidade de uma logística de coleta e transporte, devido à grande área abrangida e ao constante aumento do número de pontos de coleta. No total, foram estipulados 73 nós, além do nó 0 (Cooperativa, situada na rua Salvador Stefanelli).

No total, não foram encontradas 290 casas. Isto possivelmente aconteceu pois algumas ruas ainda não foram adicionadas ao database do *Google Earth* ou porque algumas das ruas elencadas nas planilhas são ruas internas de condomínios, onde o *Google Earth* não possui acesso. Contudo, tomando as devidas proporções, foram adicionados pontos de coletas nos condomínios onde não puderam determinar as ruas com exatidão, para o resultado ser o mais correto possível. Portanto estimou-se que 130 casas não foram realmente encontradas com a ferramenta Google Earth.

Para a elaboração dos dados, o programa *ACOTSP* requer um arquivo de leitura no formato *.tsp. Portanto foi criado um arquivo desse tipo com as coordenadas cartesianas recuperadas pelo processamento no *ImageJ*. Na Tabela 2 encontram-se os pontos cartesianos associados a cada nó.

A partir disso, foi possível elaborar o problema com o software *AcoTsp*. Como resultado da elaboração, foram gerados 3 arquivos *.txt, stat.Reviver-Sorocaba, cmp.Reviver-Sorocaba e best.Reviver-Sorocaba, respectivamente. Cada um deles possui uma parte da elaboração do problema.

O arquivo stat.Reviver-Sorocaba é utilizado como saída de informações estatística, sobre o ensaio recém elaborado. Não possui grande função para o usuário, pois reporta somente a elaboração computacional do problema.

O arquivo best.Reviver-Sorocaba contém informações sobre os parâmetros utilizados no ensaio, a melhor solução encontrada e algumas informações adicionais, listadas em Tabela 3 e Tabela 4.

Por último temos o mais importante dos 3 arquivos, o cmp.Reviver-Sorocaba. Ele inicia

Nó	x	y	Nó	x	y	Nó	x	y
0	375	422	24	526	327	49	410	452
1	905	32	25	529	342	50	450	465
2	804	102	26	494	392	51	427	467
3	726	85	27	509	365	52	482	503
4	709	83	28	555	384	53	439	485
5	725	107	29	536	375	54	396	476
6	700	106	30	522	399	55	368	473
7	700	129	31	487	390	56	356	480
8	674	106	32	546	353	57	433	516
9	679	124	33	495	427	58	417	501
10	659	138	34	387	288	59	399	514
11	624	139	35	391	330	60	359	501
12	594	199	36	419	362	61	415	534
13	573	245	37	397	385	62	347	525
14	575	286	38	447	384	63	378	526
15	602	302	39	463	402	64	339	538
16	618	313	40	419	391	65	342	542
17	647	336	41	438	411	66	377	555
18	592	315	42	465	440	67	429	557
19	582	331	43	465	449	68	442	570
20	571	342	44	480	462	69	411	575
21	647	370	45	458	449	70	385	639
22	594	361	46	438	440	71	707	442
23	550	326	47	383	423	72	587	384
			48	392	453	73	463	493

Tabela 2: Coordenadas dos nós.

Parameter-settings:	
max_tries	20
max_tours	0
max_time	10
seed	1,37E+09
n_ants	20
nn_ants	20
alpha	1.00
beta	4.00
rho	0.95
q-0	0.95

Tabela 3: Parâmetros ajustados.

Melhor Média	30325.00
Nº médio de iterações	3.60
Melhor Tentativa	30325.00
Pior Tentativa	30326.00

Tabela 4: Resultados.

com os parâmetros utilizados no problema, apresentados na Tabela 3 e posteriormente apresenta uma sessão com os dados experimentais resultantes do ensaio. Fornece, além de melhor distância e em qual interação ela foi encontrada, o roteamento entre os nós, indicando a rota otimizada a ser finalmente utilizada pela cooperativa. Seguem na Tabela 5 os dados finais apresentado pelo software.

Rotas	Distância	Sequência de Atendimento
1	50300 m	0-8-6-4-3-1-2-5-7-9-0
2	32600 m	0-11-10-12-13-26-38-40-0
3	23750 m	0-14-15-16-17-71-21-18-0
4	13260 m	0-24-23-19-22-72-20-32-25-0
5	11200 m	0-27-29-28-30-31-0
6	9800 m	0-35-34-39-33-41-36-37-47-0
7	16100 m	0-55-56-60-62-64-65-70-0
8	10600 m	0-46-42-43-45-44-73-52-0
9	7200 m	0-50-53-51-49-0
10	15800 m	0-48-54-58-57-68-67-0
11	16950 m	0-59-61-69-66-63-60-0

Tabela 5: Rota Otimizada pelo Software

Em relação a distância total a ser percorrida entre os nós, o resultado foi 30325. Lembrando que este resultado não é em metros, mas sim na escala de pixels montada, logo a distância estima pelo software será:

$$\text{Real distância} = 30325 \times 6.3 \text{ (relação)} = 191047.5 \text{ metros}$$

Além desse tipo de análise, o software retorna as melhores rotas para que todos os pontos sejam alcançados, determinando assim uma possível solução de roteamento para a cooperativa.

Para propor uma real aplicação da solução à cooperativa, fez-se necessário formar um arranjo de rotas semanal, para os 6 caminhões da cooperativa. Com o intuito de utilizar da melhor forma possível os recursos da cooperativa (frota de caminhões), resolveu-se dividir as rotas 4 e 6 pois possuem maior percurso, e as rotas 10 e 11 pois uma boa parte dos nós abrangidos são condomínios e alguns outros indústrias, gerando um volume maior de material reciclável. Assim, as rotas finais são apresentadas na Tabela 6.

Rotas	Distância	Sequência de Atendimento
1	50300 m	0-8-6-4-3-1-2-5-7-9-0
2	32600 m	0-11-10-12-13-26-38-40-0
3	23750 m	0-14-15-16-17-71-21-18-0
4A	6800 m	0-24-23-19-22-0
4B	6900 m	0-72-20-32-25-0
5	11200 m	0-27-29-28-30-31-0
6A	4500 m	0-35-34-39-33-0
6B	3900 m	0-41-36-37-47-0
7	16100 m	0-55-56-60-62-64-65-70-0
8	10600 m	0-46-42-43-45-44-73-52-0
9	7200 m	0-50-53-51-49-0
10A	8000 m	0-48-54-58-0
10B	8200 m	0-57-68-67-0
11A	8900 m	0-59-61-69-0
11B	9200 m	0-66-63-60-0

Tabela 6: Rota Otimizada.

Após essa análise, formulou-se o arranjo de rotas para os caminhões da cooperativa,

apresentado da Tabela 7. Deixou-se um dia livre para cada caminhão com a intenção que, num futuro, seja possível encaixar nessa formulação uma sucessiva ampliação da coleta seletiva em diferentes zonas da cidade de Sorocaba-SP. Algumas rotas foram repetidas no arranjo pois, como no antigo arranjo utilizado pela cooperativa, algumas rotas produzem mais material que outras, fazendo com que seja necessária mais que uma visita semanal.

Tabela 7: Arranjo de Rotas

	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
Caminhão 1	<i>Livre</i>	Rota 10A	Rota 8	Rota 3	Rota 10A
Caminhão 2	Rota 4A	Rota 10B	<i>Livre</i>	Rota 4A	Rota 10B
Caminhão 3	Rota 4B	Rota 11A	Rota 1	<i>Livre</i>	Rota 11A
Caminhão 4	Rota 6A	Rota 11B	<i>Livre</i>	Rota 6A	Rota 11B
Caminhão 5	Rota 6B	<i>Livre</i>	Rota 5	Rota 7	Rota 6B
Caminhão 6	Rota 2	Rota 7	Rota 9	Rota 2	<i>Livre</i>

6 Conclusão

Como mencionado inicialmente, a resolução de um PRV não é uma tarefa simples. Seja pelo âmbito computacional, em fatores de capacidade computacional, seja pelo âmbito de programação e também pelo âmbito de pleno entendimento. Contudo demonstrou-se que é possível otimizar uma rota, independente da finalidade primária dela.

No caso da cooperativa, o sucesso do processo completo de coleta é fruto de diferentes fatores: experiência dos operadores, acessibilidade de algumas ruas, disponibilidade dos moradores que separam o material reciclável, entre outros. Isso nos mostra que o processo depende de inúmeros fatores que não podem ser plenamente controlados. Contudo é possível, ao menos no campo do roteamento, traçar uma rota otimizada para aumentar o rendimento nesse setor

Com o crescimento da cooperativa serão necessárias sucessivas modificações no roteamento, integradas também com um aumento da frota de caminhões e dos funcionários para poder acompanhar a coleta. Contudo esse estudo de caso conseguiu fornecer uma possível solução otimizada para a situação do cenário atual. Uma proposta futura é a de trabalhar este cenário com Problemas de Roteamento de Veículos Dinâmicos.

Agradecimentos

Os meus agradecimentos vão à FUNDUNESP e ao Programa RENOVE (Edital 14/2012), pelo apoio financeiro para que a realização desse trabalho se tornasse possível. Agradeço também à Prof^{ra}. Dr^a. Luiza Amalia Pinto Cantão pela paciência na orientação e incentivo que tornaram possível a conclusão deste trabalho.

Referências

- M. D. Abramoff, P. J. Magalhães, S. J. Ram, Image processing with ImageJ. *Biophotonics international*, Universiteit Utrecht. 2004.
- M. Arenales, V. Armentano, R. Morabito, H. Yanasse, *Pesquisa Operacional*. Elsevier, Rio de Janeiro, 2007.

- L. N. de Castro, F. J. Von Zuben**, *Inteligência Coletiva*. Disponível em:
<ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia006_03/topico4_03.pdf>. Acessado em: Novembro 2012.
- G. B. Dantzig, R. H. Ramser**, *The Truck Dispatching Problem*. Management Science, 1959.
- M. Dorigo**, *Optimization, Learning and Natural Algorithms (in Italian)*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1992.
- M. Dorigo, T. Stützle**, *Ant Colony Optimization*. A Bradford Book, MIT, Cambridge Massachusetts, 2004.
- M. C. Goldberg, H. P. L. Luna**, *Otimização Combinatória e Programação Linear*. Campus, Rio de Janeiro, 2000.
- M. Manfrin, M. Birattari, T. Stützle, M. Dorigo**, *Parallel Ant Colony Optimization for the Traveling Salesman Problem*. Ant Colony Optimization and Swarm Intelligence, 2006.
- M. Lopez-Ibáñez, J. Dubois-Lacoste, T. Stützle, M. Birattari**, *The irace Package: Iterated Race for Automatic Algorithm Configuration*. IRIDIA – Institut de Recherches Interdisciplinaires et de Developpements en Intelligence Artificielle. Université Libre de Bruxelles, 2011.
- D. Palma, M. S. N. Oliveira, V. L. B. de Jesus**, *As representações integrais e o ensino de matemática nas licenciaturas em física: um exemplo simples baseado em observações experimentais*. Brasil, 2009.
- S. B. Rodrigues**, *Methaeurística Colônia de Formigas Aplicada a um Problema de Roteamento de Veículos: Caso da Itaipu Binacional*. XL SBPO, João Pessoa, PB, 2008.
- G. V. B. Simões, J. L. Ferraz, S. D. Mancini, S. H. Bonilla, W. A. Bizzo**, *Coleta Seletiva como Instrumento de Políticas Públicas: Experiência do Município de Sorocaba – SP*. *International Workshop Advances in Cleaner Production “Cleaner Production Initiatives And Challenges For A Sustainable World”*, São Paulo-Brazil, 2011.
- R. Tajimaa, Y. Katob**, *Comparison of threshold algorithms for automatic image processing of rice roots using freeware ImageJ*. Institute for Sustainable Agro-ecosystem Services, University of Tokyo, 2011.
- P. Toth, D. Vigo**, *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 2001.
- B. Tunjongsirigul, P. Pongchairerks**, *An Application of GA Algorithm on Vehicle Routing Problem in a Case Study of a Bakery Company in Thailand*. *The second RMUTP International Conference*, Thailand, 2010.
- C. Twomey, T. Stützle, M. Dorigo, M. Manfrin, M. Birattari**, *An analysis of communication policies for homogeneous multi-colony ACO algorithms*. Université Libre de Bruxelles (ULB), IRIDIA, Belgium, 2010.