Um Estudo de Modelos de Problemas de Roteamento em Arcos

Diego Venâncio Thomaz
Departamento de Matemática e Estatística
Universidade Tecnológica Federal do Paraná
Medianeira, Brasil
dvthomaz@utfpr.edu.br

Gustavo Valentim Loch
Departamento de Administração Geral e Aplicada
Universidade Federal do Paraná
Curitiba, Brasil
gustavo.gvalentim@gmail.com

Tatiane Tambarussi Thomaz
Coordenação de Ciência da Computação
Universidade Tecnológica Federal do Paraná
Santa Helena, Brasil
tambarussi@utfpr.edu.br

***Resumo*—O estudo dos problemas de roteamento tem grande importância na área da Pesquisa Operacional, pois sua aplicabilidade abrange diversos problemas que buscam otimização de rotas e economias financeiras. Este trabalho tem por objetivo trazer um estudo de modelos de problemas de roteamento em arcos, especificamente, problemas de coleta de lixo. Além dos modelos matemáticos são também apresentados os códigos para implementação no software LINGO.**

Palavras-chave—coleta de lixo; modelo exato; roteamento em arcos

I. INTRODUÇÃO

Os problemas de roteamento estão presentes em diversos contextos práticos onde há a necessidade de otimização de rotas, como na coleta de lixo, medidores e entregadores de faturas, varredores de ruas, entre outros.

Com a evolução dos modelos para problemas da Pesquisa Operacional evoluíram também as ferramentas que podem ser utilizadas na resolução desses problemas, como o LINGO, que foi escolhido por se tratar de um software de linguagem simples de implementação e por oferecer atrativos tempos computacionais.

Deve-se buscar em um grafo um circuito que percorra todos os arcos/arestas pelo menos uma vez, dentro de um período estabelecido, que minimize essa rota, solucionando o problema de roteamento. Quando um grafo representa uma malha rodoviária, este tem características do caso Misto do Problema do Carteiro Chinês, por apresentar arcos (direcionados) e arestas (não direcionadas), representando as ruas de mão única e mão dupla, no mesmo grafo [3]. Existem vários modelos que visam solucionar os problemas de roteamento, cada um dependendo das características trazidas por eles, podendo citar os trabalhos [1], [6], [7], [8] e [9].

O Problema do Carteiro Chinês (CPP), do inglês, Chinese Postman Problem, consiste em encontrar uma rota com menor custo possível em que o carteiro a partir de um depósito entregue suas cartas e retorne ao depósito. Já no contexto da coleta de lixo, um veículo deve sair de um depósito e passar sobre um determinado conjunto de ruas coletando a demanda de lixo necessária com o menor custo (distância) possível e retornar ao depósito ao fim desta tarefa. Com o objetivo de se representar problemas de coleta de lixo ainda mais reais foram propostos outros modelos levando em consideração a quantidade de lixo a ser coletada, consequentemente a capacidade de lixo que o veículo suporta carregar e um horizonte de planejamento maior que um dia. Foram eles:

O Problema de Roteamento em Arcos Capacitados (CARP), do inglês, Capacitated Arc Routing Problem, consiste em coletar em um único dia uma demanda de lixo utilizando um conjunto de veículos que partem e retornam ao depósito sem exceder suas capacidades [1].

A modelagem do planejamento da coleta de lixo urbano sobre um período maior que um dia pode ser feita como um Problema de Roteamento Periódico em Arcos Capacitados (PCARP), do inglês, Periodic Capacitated Arc Routing Problem [6].

Estes modelos serão apresentados nas seções seguintes.

II. O PROBLEMA DO CARTEIRO CHINÊS

Quando uma malha urbana, ou uma rede rodoviária, é representada por um grafo, os arcos representam os trechos de ruas de mão única e as arestas, os de mão dupla. Os nós são os cruzamentos entre as ruas. Todavia, um grafo é uma estrutura mais genérica do que uma malha urbana.

Em [4] tem-se que um grafo completo $G = (V, A)$ é uma estrutura, onde V é o conjunto dos vértices (pontos ou nós) e A o conjunto das arestas. Um grafo é dito orientado quando

para quaisquer nós $x_i \in V$ o arco $(x_i, x_j) \in A$ e não orientado quando para os nós $x_i \in V$ a aresta $(x_i, x_j) \in A$.

O CPP é um tipo de problema que envolve roteamento em arcos e tem diversas aplicações, como por exemplo, na coleta de lixo, limpeza de ruas, remoção de neve das vias públicas, inspeções periódicas em linhas elétricas, redes de gasodutos, leitura de medidores de consumo de água, energia, entre outros [7].

De maneira geral o CPP, somente pode ser resolvido em tempo não polinomial. A modelagem matemática para o CPP não direcionado, partindo de um grafo $G = (V, A)$ é dada em [9]:

$$\min z = \sum_{(i,j)\in A} c_{ij} x_{ij} \quad (1)$$

sujeito à:

$$\sum_{(i,j)\in A} x_{ij} = \sum_{(j,i)\in A} x_{ji} \quad \forall i \in V \quad (2)$$

$$x_{ij} + x_{ji} \geq 1, \forall (i,j) \in A \quad (3)$$

$$x_{ij} \in Z^+ \quad (4)$$

A variável x_{ij} representa a quantidade de vezes que ocorre o movimento do nó i para o nó j, fato garantido por (4). A expressão (1) é a função objetivo do modelo que consiste em minimizar o custo total de utilização das arestas. O fluxo contínuo é garantido em (2), e (3) garante que todo arco será atravessado.

III. TESTE NUMÉRICO PARA O CPP

Para contextualizar o CPP vamos considerar o grafo apresentado na Fig.1.

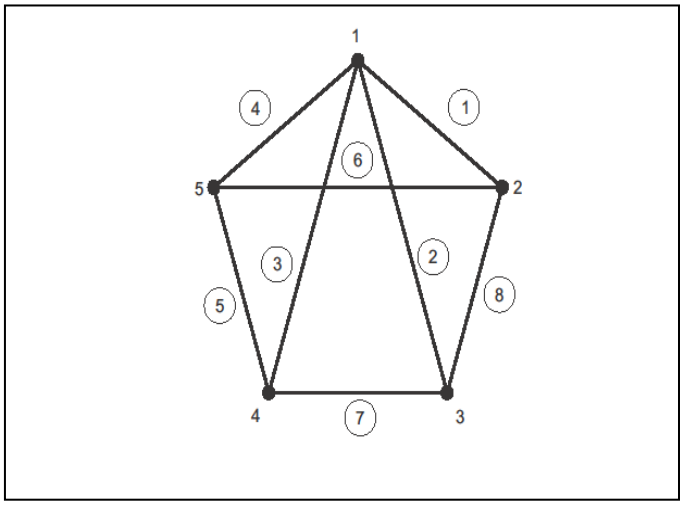

Fig. 1. Grafo representando um conjunto de ruas.

Este é um grafo não orientado possuindo 5 nós e 8 arestas cujos custos c_{ij} de atravessar cada arco do nó i para o nó j são os valores circulados sobre cada uma das arestas. Por exemplo: o custo de se atravessar o arco (1,5) é 4.

Neste caso vamos considerar a possibilidade de se utilizar o depósito como parte da rota durante o percurso. O modelo matemático para o CPP proposto em [9] foi implementado no software LINGO, cujo código é apresentado na Fig. 2.

```
SETS:
PONTOS/1..5/:;
ROTAS(PONTOS,PONTOS):d,x;
ENDSETS

DATA:
!Matriz de custo;
d =
    0 1 2 3 4
    1 0 8 0 6
    2 8 0 7 0
    3 0 7 0 5
    4 6 0 5 0;

ENDDATA

!Função objetivo
MIN = FO;
FO = @SUM(ROTAS(i,j):d(i,j)*x(i,j));
! Restrições;
@FOR(PONTOS(n):@SUM(ROTAS(i,n)|d(i,n)#N
E#0:x(i,n))=
@SUM(ROTAS(n,j)|d(n,j)#NE#0:x(n,j)));
@FOR(ROTAS(i,j)|d(i,j)#NE#0:x(i,j)+x(j,
i)>=1);
@FOR(ROTAS(i,j):@GIN(x(i,j)));
END
```

Fig. 2. PCC implementado no LINGO.

Neste código temos que d é a matriz de custos onde em cada uma das posições (i, j) temos o custo de se atravessar o arco (i, j). Por exemplo: para $d(2,3) = 8$, tem-se o custo de atravessar o arco $(2,3)$. Vale ressaltar que na diagonal principal da matriz d temos que todos os custos são nulos, pois não há um arco neste caso. O comando `d(n,j)#NE#0` utilizado na implementação do modelo faz com que sejam considerados apenas os arcos que possuam custo diferente de zero na matriz d. São exibidas ainda a função objetivo e o conjunto de restrições. Este código foi compilado no software LINGO e chegamos até o resultado ótimo cuja rota ótima é dada por:

$$1 \to 2 \to 3 \to 1 \to 2 \to 5 \to 1 \to 4 \to 5 \to 4 \to 3 \to 1$$

Em que o custo da mesma é $z = 44$. Se for considerada uma tarefa (coleta ou entrega) em cada arco é necessária a

introdução de outro modelo, o CARP, que é uma extensão natural do CPP e será definido a seguir.

IV. O PROBLEMA DE ROTEAMENTO EM ARCOS CAPACITADOS

O CARP é caracterizado por ter uma demanda associada a cada arco do grafo, em geral, esta demanda está associada à coleta de resíduos ou a entrega de objetos. Os veículos de uma frota de capacidade W devem atravessar os arcos coletando ou entregando determinadas demandas sem excederem suas capacidades, este problema foi proposto por Golden e Wong em 1981 [5].

O modelo matemático proposto em [2] parte de um grafo $G = (V, A)$ e utiliza as seguintes parâmetros:

- R é o conjunto de arcos requeridos em A.
- q_{ij} é a demanda do arco $(i, j) \in R \subset A$.
- W é a capacidade do veículo k.
- c_{ij} é o custo de atravessar o arco $(i, j) \in A$.
- K é a quantidade de veículos.

E as seguintes variáveis de decisão:

- x_{ij} é o número de vezes que um veículo k atravessa o arco $(i, j) \in A$.
- y_{ij} assume o valor 1 se o veículo k descarrega ou coleta ao longo do arco $(i, j) \in R$ e 0 caso contrário.

É modelado da seguinte forma:

$$\min z = \sum_{(i,j)\in A} \sum_{k=1}^{K} c_{ij} x_{ijk} \tag{5}$$

sujeito à:

$$\sum_{p\in A} x_{pik} = \sum_{p\in A} x_{ipk} \forall i \in V, \forall k = 1,2,...,K \tag{6}$$

$$\sum_{k=1}^{K} y_{ijk} = 1 \forall (i, j) \in R \tag{7}$$

$$x_{ijk} \geq y_{ijk} \forall (i, j) \in R, k = 1,2,...,K \tag{8}$$

$$\sum_{(i,j)\in R} q_{ij} y_{ijk} \leq W \forall = 1,2,...,K \tag{9}$$

$$M \sum_{i \notin S,\, j \in S} x_{ijk} \geq \sum_{(j,p) \in A[S] \cap R} x_{jpk} \quad \forall S \subseteq N, \quad 1 \notin S, A[S] \cap R \neq 0, k = 1,2,..., K \tag{10}$$

$$x_{ijk} \in Z^{+}, \forall (i, j) \in A, k = 1,2,...,K \tag{11}$$

$$y_{ijk} \in \{0,1\}, \forall (i, j) \in R, k = 1,2,...,K \tag{12}$$

A Equação (5) é a função objetivo que minimiza o custo representado pela distância total percorrida pelos veículos. Vale ressaltar que os arcos podem ser atravessados mais de uma vez. A Equação (6) garante a conservação do fluxo dos veículos ao longo do trajeto. A Equação (7) garante que cada arco seja atendido uma única vez durante todo o trajeto por um único veículo k. Um carro só fará a entrega ou coleta em um arco $(i, j) \in R$ se for atravessar este arco, garantido por (8). A Equação (9) se refere a capacidade do veículo. A Equação (10) evita a formação de subciclos. As Equações (11) e (12) garantem que as variáveis x_{ijk} e y_{ijk} são inteiras e binárias, respectivamente.

V. TESTE NUMÉRICO PARA O CARP

Para ilustrar o CARP consideramos o grafo dado na Fig. 1 e a Tabela I que apresenta as demandas sobre cada um dos arcos. Por exemplo: a demanda de coleta de lixo sobre o arco $(2,3)$ é de 700 quilos com um custo 8.

TABELA I. DEMANDA SOBRE CADA ARCO

Nós	1	2	3	4	5
1	0	500	700	800	1000
2	500	0	700	0	900
3	700	700	0	300	0
4	800	0	300	0	200
5	1000	900	0	200	0

A formulação matemática proposta em [2] foi implementada no software LINGO, cujo código está na Fig. 3.

No código temos que d é a mesma matriz de custos apresentada no código do CPP e t a matriz das demandas sobre cada um dos arcos apresentada na Tabela I. Nesta implementação foram considerados 2 veículos de capacidade individual de 2600 quilos. São apresentadas ainda a função objetivo e o conjunto de restrições. Este código foi compilado no software LINGO e chegamos até o resultado ótimo que traz uma rota a ser percorrida por cada um dos veículos dada por:

$$Veículo1: 1 \to 3 \xrightarrow{S} 2 \xrightarrow{S} 5 \to 4 \xrightarrow{S} 3 \xrightarrow{S} 1$$

$$Veículo2: 1 \xrightarrow{S} 4 \xrightarrow{S} 5 \xrightarrow{S} 1 \xrightarrow{S} 2 \to 1$$

Assim, o veículo 1 atravessa o arco $(1,3)$, atravessa e realiza o serviço (coleta de lixo) sobre o arcos $(3,2)$, $(2,5)$, atravessa o arco $(5,4)$, atravessa e realiza o serviço sobre os arcos $(4,3)$ e $(3,1)$. Da mesma forma podemos fazer uma

leitura da rota do veículo 2 onde a letra S sobre cada uma das setas indicando o caminho a ser percorrido significam a passagem e realização do serviço sobre aquele arco. O custo desta rota é $z = 44$.

```
SETS:
PONTOS/1..5/:;
VEICULOS/1..2/:;
ROTAS(PONTOS,PONTOS):d,t;
VETOR(PONTOS,PONTOS,VEICULOS):x,y;
ENDSETS
DATA:
!Matriz de custo;
d =
    0 1 2 3 4
    1 0 8 0 6
    2 8 0 7 0
    3 0 7 0 5
    4 6 0 5 0;

!Matriz de demanda de cada arco;
t=
   0 500 700 800 1000
   500 0 700 0 900
   700 700 0 300 0
   800 0 300 0 200
   1000 900 0 200 0;
ENDDATA
!Função Objetivo;
MIN = FO;
FO =
@SUM(VETOR(i,j,k):d(i,j)*x(i,j,k));
! Restrições;
!Conservação do Fluxo;
@FOR(VEICULOS(k):@FOR(PONTOS(n):@SUM(P
ONTOS(i)|d(i,n)#NE#0:x(i,n,k))=
@SUM(PONTOS(j)|d(n,j)#NE#0:x(n,j,k))))
;
!Garante o atendimento do arco por um
único veículo;
@FOR(ROTAS(i,j)|d(i,j)#NE#0:@SUM(VEICU
LOS(k):y(i,j,k)+y(j,i,k))=1);
!O veículo só fará a tarefa se for
atravessado;
@FOR(VETOR(i,j,k):x(i,j,k)>=y(i,j,k));
!Capacidade do veículo;
@FOR(VEICULOS(k):@SUM(ROTAS(i,j)|d(i,j
)#NE#0:t(i,j)*y(i,j,k))<=2600);
!x(i,j,k) são variáveis inteiras;
@FOR(VETOR(i,j,k)|d(i,j)#NE#0:@GIN(x(i
,j,k)));
!y(i,j,k) são variáveis binárias;
@FOR(VETOR(i,j,k)|d(i,j)#NE#0:@BIN(y(i
,j,k)));
END
```

Fig. 3. CARP implementado no LINGO.

Quando o objetivo é fazer o planejamento da realização do serviço para um horizonte de tempo maior que um dia há a necessidade da inserção de outro modelo, o PCARP uma extensão natural do CARP e será definido a seguir.

VI. O PROBLEMA DE ROTEAMENTO PERIÓDICO EM ARCOS CAPACITADOS

O PCARP é uma extensão do CARP, onde o planejamento a ser realizado é feito sobre um horizonte de tempo maior com múltiplos períodos. Em [1] os autores Chu, Labadi e Prins propõem um modelo que utiliza um conjunto de combinações de dias permitidos $comb(i, j)$ em que o serviço pode ser executado e nc como o número total de distintas combinações de dias. Estas combinações são enumeradas de 1 até nc e definidas por uma matriz $A(nc \times P)$, onde os elementos $A_{bp} = 1$ se a combinação b contém o dia p. O nó 1 é o depósito que contém K veículos idênticos. Os parâmetros para definir o modelo são dados da seguinte forma:

- W é capacidade dos veículos.
- Q_{ijbp} é a demanda a ser atendida por qualquer combinação b de dias e qualquer dias p é conhecido.
- c_{ij} é o custo de se atravessar o arco $[i, j]$ do nó i para o nó j.
- l_{ijkbp} assume o valor 1 se o arco $[i, j]$ é servido de i para j pelo veículo k no dia p da combinação b.
- Z_{ijb} recebe o valor 1 se o serviço $[i, j]$ utiliza a combinação b.

O problema é então formulado da seguinte forma:

$$\min z = \sum_{(i,j)\in A} \sum_{k=1}^{K} \sum_{p=1}^{P} c_{ij}(x_{ijkp} + x_{jikp}) \qquad (13)$$

sujeito à:

$$\sum_{b \in comb(i,j)} Z_{ijb} = 1 \forall [i, j] \in R \qquad (14)$$

$$\sum_{[i,j]\in R} x_{ijkp} = \sum_{[i,j]\in R} x_{jikp} \forall k = 1,2,...,K, \quad \forall p = 1,2,...,P, \forall i \in X \qquad (15)$$

$$x_{ijkp} \geq \sum_{b \in comb(i,j)} l_{ijkbp} \forall [i, j] \in R, \quad \forall k = 1,2,...,K, \forall p = 1,2,...,P \qquad (16)$$

$$x_{ijkp} \geq \sum_{b \in comb(i,j)} l_{jikbp} \forall [i,j] \in R, \forall k = 1,2,...,K, \quad (17)$$
$$\forall p = 1,2,...,P$$

$$\sum_{k=1}^{K} (l_{ijkbp} + l_{jikbp}) = A_{bp} Z_{ijb} \forall [i,j] \in R, \quad (18)$$
$$\forall k \in comb(i,j), \forall p = 1,2,...,P$$

$$\sum_{[i,j] \in R} \sum_{b \in comb(i,j)} Q_{ijbp} (l_{ijkbp} + l_{jikpb}) \leq W \quad (19)$$
$$\forall p = 1,2,...,P, \forall k = 1,2,...,K$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ijkp} \geq l_{rskbp} + l_{srkbp} \forall S \subseteq \{s,...,n\}, \quad (20)$$
$$\forall [r,s] \in E_r(S), \forall p = 1,2,...,P,$$
$$\forall k \in comb(r,s), \forall k = 1,2,...,K$$

$$x_{ijkp}, x_{jikp}, Z_{ijb}, l_{ijkbp}, l_{jikbp} \in \{0,1\}, \forall [i,j] \in R, \quad (21)$$
$$\forall p = 1,2,...,P, \forall k = 1,2,...,K, \forall b \in comb(i,j)$$

A Equação (13) traz a função objetivo que minimiza o custo total das viagens. A Equação (14) garante que apenas uma das combinações de dias pode ser designada para cada tarefa. A Equação (15) garante a conservação do fluxo dos veículos ao longo do trajeto. As Equações (16) e (17) garantem que um arco só será servido se o mesmo for atravessado. Cada tarefa só pode ser atendida por um carro e em apenas uma direção, fato garantido em (18). A Equação (19) se refere a capacidade do veículo. A Equação (20) evita a formação de subciclos. A Equação (21) garante que todas as variáveis são binárias.

VII. TESTE NUMÉRICO PARA O PCARP

Para exemplificar o PCARP vamos utilizar o grafo apresentado na Fig. 4. Este grafo representa um conjunto de ruas, onde as arestas contínuas são as ruas que necessitam de visitas diárias e as arestas pontilhadas as ruas que necessitam de visitas uma vez a cada dois dias sem repetição.

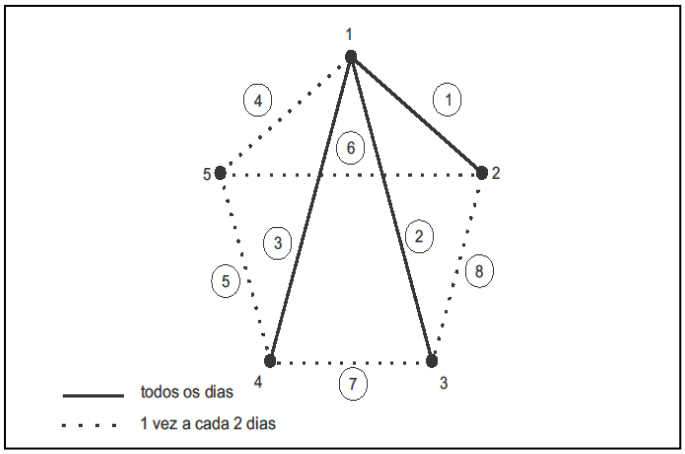

Fig. 4. Grafo representando um conjunto de ruas.

As demandas diárias de cada um dos arcos são apresentadas na Tabela 2:

TABELA II. DEMANDA DIÁRIA SOBRE CADA ARCO

Arestas/Dias	1	2	3	4	5
(1,2)	800	500	400	300	300
(1,3)	700	650	550	900	400
(1,4)	800	650	650	1000	600
(1,5)	200	0	300	0	500
(2,3)	700	0	400	0	500
(2,5)	700	0	400	0	500
(3,4)	400	0	300	0	200
(4,5)	200	0	150	0	200

A aresta $(2,3)$ deve ser visitada a cada dois dias com uma demanda de lixo estimada em 700, 400 e 500 quilos nos dias 1, 3 e 5, respectivamente. A aresta $(1,2)$ deve ser visitada diariamente com demanda estimada em 800, 500, 400, 300 e 300 quilos de lixo nos dias 1, 2, 3, 4 e 5, respectivamente. A formulação matemática proposta em [1] foi implementada no software LINGO, cujo código á trazido na Fig. 5.

```
SETS:
PONTOS/1..5/:;
VEICULOS/1..2/:;
DIAS/1..5/:;
COMB/1..1/:;
ROTAS(PONTOS,PONTOS):d,t;
VETOR1(PONTOS,PONTOS,COMB):z;
VETOR(PONTOS,PONTOS,VEICULOS,DIAS):x;
VETOR2(PONTOS,PONTOS,VEICULOS,COMB,DI
AS):l;
VETOR3(COMB,DIAS):A1_2,A2_1,A1_3,A3_1
,A1_4,A4_1,A1_5,A5_1,A2_5,A5_2,A2_3,A
3_2,A3_4,A4_3,A4_5,A5_4;
VETOR4(COMB,DIAS):Q1_2,Q2_1,Q1_3,Q3_1
,Q1_4,Q4_1,Q1_5,Q5_1,Q2_5,Q5_2,Q2_3,Q
3_2,Q3_4,Q4_3,Q4_5,Q5_4;
VETOR5(COMB,COMB):q;

ENDSETS
DATA:
!matriz de custo;
d =
    0 1 2 3 4
    1 0 8 0 6
    2 8 0 7 0
    3 0 7 0 5
    4 6 0 5 0;
```

```
!matriz das combinações;

A1_2 = 1 1 1 1 1;
A2_1 = 1 1 1 1 1;
A1_3 = 1 1 1 1 1;
A3_1 = 1 1 1 1 1;
A1_4 = 1 1 1 1 1;
A4_1 = 1 1 1 1 1;
A1_5 = 1 0 1 0 1;
A5_1 = 1 0 1 0 1;
A2_5 = 1 0 1 0 1;
A5_2 = 1 0 1 0 1;
A2_3 = 1 0 1 0 1;
A3_2 = 1 0 1 0 1;
A4_5 = 1 0 1 0 1;
A5_4 = 1 0 1 0 1;
A4_3 = 1 0 1 0 1;
A3_4 = 1 0 1 0 1;

!Matriz das demandas;

Q1_2 = 800 500 400 300 300;
Q2_1 = 800 500 400 300 300;
Q1_3 = 700 650 550 900 400;
Q3_1 = 700 650 550 900 400;
Q1_4 = 800 650 650 1000 600;
Q4_1 = 800 650 650 1000 600;
Q1_5 = 200 0 300 0 500;
Q5_1 = 200 0 300 0 500;
Q2_5 = 700 0 400 0 500;
Q5_2 = 700 0 400 0 500;
Q2_3 = 700 0 400 0 500;
Q3_2 = 700 0 400 0 500;
Q4_5= 200 0 150 0 200;
Q5_4= 200 0 150 0 200;
Q4_3 = 400 0 300 0 200;
Q3_4 = 400 0 300 0 200;
ENDDATA
!Função Objetivo;
MIN = FO;
FO =
@SUM(VETOR(i,j,k,p):d(i,j)*(x(i,j,k,p)
));
! restrições;

!Apenas uma combinação de dias pode
ser designado para cada tarefa;
@FOR(ROTAS(i,j)|d(i,j)#NE#0:@SUM(COMB(
c):z(i,j,c))=1);

!Manutenção do fluxo;
@FOR(PONTOS(i):@FOR(DIAS(p):@FOR(VEICU
LOS(k):@SUM(PONTOS(j)|d(i,j)#NE#0:x(i,
j,k,p))=@SUM(PONTOS(j)|d(j,i)#NE#0:x(j
,i,k,p)))));
```

```
!O veículo só fará a tarefa se for
atravessado;
@FOR(VETOR(i,j,k,p)|d(i,j)#NE#0:x(i,
j,k,p)>=@SUM(COMB(c):l(i,j,k,c,p)));
@FOR(VETOR(j,i,k,p)|d(i,j)#NE#0:x(j,
i,k,p)>=@SUM(COMB(c):l(j,i,k,c,p)));

!Cada tarefa só pode ser atendida
por um único véículo em uma única
direção;
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS
(i,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(
l(1,2,k,c,p)+l(2,1,k,c,p)))=A1_2(c,p
)*z(1,2,c))));
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS
(i,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(
l(2,1,k,c,p)+l(1,2,k,c,p)))=A2_1(c,p
)*z(2,1,c))));
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS
(i,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(
l(1,3,k,c,p)+l(3,1,k,c,p)))=A1_3(c,p
)*z(1,3,c))));
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS
(i,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(
l(3,1,k,c,p)+l(1,3,k,c,p)))=A3_1(c,p
)*z(3,1,c))));
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS
(i,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(
l(1,4,k,c,p)+l(4,1,k,c,p)))=A1_4(c,p
)*z(1,4,c))));
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS
(i,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(
l(4,1,k,c,p)+l(1,4,k,c,p)))=A4_1(c,p
)*z(4,1,c))));
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS
(i,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(
l(1,5,k,c,p)+l(5,1,k,c,p)))=A1_5(c,p
)*z(1,5,c))));
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS
(i,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(
l(5,1,k,c,p)+l(1,5,k,c,p)))=A5_1(c,p
)*z(5,1,c))));
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS
(i,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(
l(2,5,k,c,p)+l(5,2,k,c,p)))=A2_5(c,p
)*z(2,5,c))));
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS
(i,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(
l(5,2,k,c,p)+l(2,5,k,c,p)))=A5_2(c,p
)*z(5,2,c))));
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS
(i,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(
l(4,5,k,c,p)+l(5,4,k,c,p)))=A4_5(c,p
)*z(4,5,c))));
```

```
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS(i
,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(l(5,
4,k,c,p)+l(4,5,k,c,p)))=A5_4(c,p)*z(5,
4,c))));
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS(i
,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(l(2,
3,k,c,p)+l(3,2,k,c,p)))=A2_3(c,p)*z(2,
3,c))));
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS(i
,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(l(3,
2,k,c,p)+l(2,3,k,c,p)))=A3_2(c,p)*z(3,
2,c))));
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS(i
,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(l(3,
4,k,c,p)+l(4,3,k,c,p)))=A3_4(c,p)*z(3,
4,c))));
@FOR(COMB(c):@FOR(DIAS(p):@FOR(ROTAS(i
,j)|d(i,j)#NE#0:@SUM(VEICULOS(k):(l(4,
3,k,c,p)+l(3,4,k,c,p)))=A4_3(c,p)*z(4,
3,c))));
!Capacidade do veículo;
@FOR(DIAS(p):@FOR(VEICULOS(k):@SUM(COM
B(c):(Q1_2(c,p)*(l(1,2,k,c,p)+l(2,1,k,
c,p))+Q2_1(c,p)*(l(2,1,k,c,p)+l(1,2,k,
c,p))+Q1_3(c,p)*(l(1,3,k,c,p)+l(3,1,k,
c,p))+Q3_1(c,p)*(l(3,1,k,c,p)+l(1,3,k,
c,p))+Q1_4(c,p)*(l(1,4,k,c,p)+l(4,1,k,
c,p))+Q4_1(c,p)*(l(4,1,k,c,p)+l(1,4,k,
c,p))+Q1_5(c,p)*(l(1,5,k,c,p)+l(5,1,k,
c,p))+Q5_1(c,p)*(l(5,1,k,c,p)+l(1,5,k,
c,p))+Q2_5(c,p)*(l(2,5,k,c,p)+l(5,2,k,
c,p))+Q5_2(c,p)*(l(5,2,k,c,p)+l(2,5,k,
c,p))+Q2_3(c,p)*(l(2,3,k,c,p)+l(3,2,k,
c,p))+Q3_2(c,p)*(l(3,2,k,c,p)+l(2,3,k,
c,p))+Q3_4(c,p)*(l(3,4,k,c,p)+l(4,3,k,
c,p))+Q4_3(c,p)*(l(4,3,k,c,p)+l(3,4,k,
c,p))+Q4_5(c,p)*(l(4,5,k,c,p)+l(5,4,k,
c,p))+Q5_4(c,p)*(l(5,4,k,c,p)+l(4,5,k,
c,p)))<=5000));

!Todas as variáveis são binárias;

@FOR(VETOR(i,j,k,p)|d(i,j)#NE#0:@BIN(x
(i,j,k,p)));
@FOR(VETOR(j,i,k,p)|d(j,i)#NE#0:@BIN(x
(j,i,k,p)));
@FOR(VETOR2(i,j,k,c,p)|d(i,j)#NE#0:@BI
N(l(i,j,k,c,p)));
@FOR(VETOR2(j,i,k,c,p)|d(j,i)#NE#0:@BI
N(l(j,i,k,c,p)));
@FOR(VETOR1(i,j,c)|d(i,j)#NE#0:@BIN(z(
i,j,c)));
END
```

Fig. 5. PCARP implementado no LINGO

Neste código temos que d é a matriz de custos apresentada no CPP, Ai_j é a matriz das combinações de dias de visitas e Qi_j a matriz das demandas em cada um dos arcos. Foi considerado um horizonte de tempo de 5 dias e 2 veículos com capacidade de 5000 quilos individuais para coletar as demandas sobre cada um dos arcos do conjunto de ruas apresentado na Fig. 4. A Fig. 5 traz ainda a função objetivo e o conjunto de restrições. Este código foi compilado no software LINGO e chegamos até o resultado ótimo que traz uma rota a ser percorrida por cada um dos veículos:

*Veículo*1

$Dia1: 1 \xrightarrow{S} 3 \xrightarrow{S} 2 \xrightarrow{S} 5 \xrightarrow{S} 4 \xrightarrow{S} 3$

$3 \rightarrow 1 \xrightarrow{S} 2 \rightarrow 1$

$Dia3: 1 \rightarrow 2 \xrightarrow{S} 5 \rightarrow 4 \xrightarrow{S} 1$

$Dia5: 1 \rightarrow 3 \xrightarrow{S} 4 \xrightarrow{S} 1 \xrightarrow{S} 2 \rightarrow 1$

*Veículo*2

$Dia1: 1 \xrightarrow{S} 4 \rightarrow 5 \xrightarrow{S} 1$

$Dia2: 1 \xrightarrow{S} 2 \rightarrow 1 \rightarrow 3 \xrightarrow{S} 1 \rightarrow 4 \xrightarrow{S} 1$

$Dia3: 1 \xrightarrow{S} 5 \xrightarrow{S} 4 \xrightarrow{S} 3 \xrightarrow{S} 2 \xrightarrow{S} 1$

$1 \rightarrow 3 \xrightarrow{S} 1$

$Dia4: 1 \xrightarrow{S} 2 \rightarrow 1 \rightarrow 3 \xrightarrow{S} 1 \rightarrow 4 \xrightarrow{S} 1$

$Dia5: 1 \xrightarrow{S} 5 \xrightarrow{S} 4 \rightarrow 5 \xrightarrow{S} 2 \xrightarrow{S} 3 \xrightarrow{S} 1$

As demandas de todas as arestas de fato foram atendidas de acordo com sua necessidade. A demanda da aresta $(1,2)$ foi coletada todos os dias e isto ocorreu nos dias 1 e 5 pelo veículo 1 e nos dias 2, 3 e 4 pelo veículo 2. Já para a aresta $(1,5)$ as coletas foram realizadas nos dias 1, 3 e 5 pelo veículo 2, pois a mesma possuía a necessidade de visita a cada dois dias sem repetição. O custo da rota ótima a ser realizada pelos 2 veículos durante os 5 dias é $z = 156$.

VIII. CONSIDERAÇÕES FINAIS

Através deste trabalho podemos mostrar um estudo além das equações algébricas, função objetivo e conjunto de restrições. Para cada um dos problemas de roteamento em arcos trazidos neste trabalho mostramos os códigos implementados no LINGO para problemas de pequenos porte, com poucos nós e arcos. Vale ressaltar que mesmo para estes problemas de pequeno porte os modelos contém um alto número de variáveis.

REFERÊNCIAS

[1] F. Chu, N. Labadi and C. Prins, “Heurististics for the Periodic Routing Problem”, Journal of Intelligent Manufacturing, vol. 16, 2005.

[2] M. Dror, “Arc Routing: Theory, Solutions, and Applications”, Mass. Kluwer Academic. Boston, 2000.

[3] H. A. Eiselt, M. Gendreau, G. Laporte, “Arc Routing Problems, Part I: The Chinese Postman Problem,” Operations Research, vol. 43, n°2, 1995.

[4] M. C. Goldbarg, “Otimização Combinatória e Programação Linear: Modelos e Algoritmos,” 2ª Edição. Rio de Janeiro, 2005.

[5] B. L. Golden, R. T. Wong, “Capacitated Arc Routing Problem”, Networks, vol. 11, 1981.

[6] P. Lacomme, C. Prins, W. Ramdane-Chérif, “Evolutionary Algorithms for Periodic Arc Routing Problems”, European Journal of Operational Research, vol. 165, 2005.

[7] I. M. Monroy, C. A. Amaya, A. Langevin, “The Periodic Capacitated Arc Routing Problem With Irregular Services”, Discrete Applied Mathematics, vol.161, 2013.

[8] G. V. Batista, “Proposta de Um Modelo Matemático Para o Problema de Roteamento em Arcos Capacitado e Periódico,” Dissertação de Mestrado. UFPR, Curitiba, 2014.

[9] H. F. Wang, Y. P. Wen, “Time-Constrained Chinese Postman Problems”, Computers and Mathematics with Applications, vol. 44, 2002.