Universidade Tecnológica Federal do Paraná
Campus Medianeira – Paraná – Brasil
ISSN 2175-1846 / v.01, n01, 2010

INOVAÇÃO E TECNOLOGIA

Um estudo sobre roteirização

Patricia Fernandes Mendonça[1]; Franciele Buss Frescki Kestring[2]
Fausto Pinheiro da Silva[3]1,2,3 Universidade Tecnológica Federal do Paraná – UTFPR –Medianeira – Brasil
patymendonca_17@hotmail.com
francieleb@utfpr.edu.br
faustosilva@utfpr.edu.br

Resumo: *O presente trabalho tem por objetivo apresentar e implementar alguns métodos que resolvem o problema de minimização de rotas e os respectivos instrumentos computacionais nos quais é possível a implementação de tais métodos. Para contextualizar tal problema, utilizou-se o percurso de um carteiro em um bairro do município de São Miguel do Iguaçu, oeste do Paraná. Realizou-se previamente um estudo sobre Teoria dos Grafos, o Problema do Carteiro Chinês (PCC) e conceitos básicos de Programação Linear. Neste trabalho foram analisados os algoritmos de Dijkstra e Floyd, bem como algoritmos de rota utilizando Programação Linear e o algoritmo de Hierholzer. Foram utilizados os softwares Excel, Lingo e TORA para implementar tais algoritmos.Verificou-se que o uso das tecnologias facilitou a implementação dos algoritmos e foi possível otimizar a rota praticada pelo carteiro.*

Palavras - chave: Teoria dos Grafos; Algoritmo de Floyd; Problema do Carteiro Chinês.

1. Introdução

Os problemas de transporte e roteirização vêm recebendo grande destaque em pesquisas científicas devido à sua complexidade e importância nos sistemas de distribuição de mercadorias e atendimento de serviços. Autores como Raduan (2009), destacam que, no aspecto operacional, para que determinada atividade de transporte de bens e serviços seja realizada de forma eficiente, há que se planejar adequadamente a sequência de locais a serem atendidos e em que instante no tempo.

A Teoria dos Grafos é uma das maneiras de se abordar tais problemas de transporte e roteirização. Essa teoria teve seus primeiros estudos no século XVIII, quando o matemático Leonard Euler propôs o problema conhecido como “o problema das sete pontes de Königsberg”, Figura 1.

Com a evolução da tecnologia da informação, a utilização de algoritmos para resolver problemas envolvendo grafos vieram no auxílio de várias situações onde a implementação computacional se fez necessária, como problemas para encontrar o caminho mínimo, ou com o custo mínimo. Alguns algoritmos para encontrar distâncias entre vértices de um grafo, por exemplo, os algoritmos de Floyd e de Dijkstra, necessitam de implementação computacional para resolver problemas de grande porte. Outra situação na qual é interessante utilizar *softwares* é o Problema do Carteiro Chinês, PCC,que é

um exemplo de otimização combinatória que proporciona a solução de vários problemas do mundo real, como análise de recursos para favorecer a coleta de lixo urbano trazendo redução dos custos, rota para entrega de cartas, rota de caminhões para empresas de transporte dentre outros.

O estudo sobre a Teoria dos Grafos cria um viés para solução de vários problemas nos quais a matemática aplicada é fundamental, justificando assim o seu estudo, embasado em referenciais teóricos e em sua aplicação, por meio da análise dos dados apresentados como resultado pelos *softwares*. Neste artigo estudou-se o Problema do Carteiro Chinês Não Direcionado, (PCCND), que consiste em encontrar o melhor caminho sendo este o mais curto, num circuito fechado, que visite assim todas as arestas de um grafo (conexo) não direcionado.

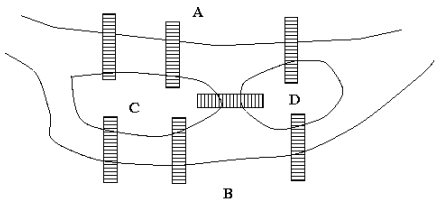
Este trabalho teve por objetivo apresentar alguns algoritmos que resolvem problemas de roteirização. Foram identificados modelos matemáticos que são aplicados na resolução de algoritmos de minimização. Para exemplificar os algoritmos, foram utilizados dados de um bairro real, no qual atua um carteiro. No grafo que representa esse bairro, foram aplicados os algoritmos de Dijkstra e Floyd para encontrar uma rota mínima entre dois vértices predeterminados. Esses dois algoritmos foram executados no *software* TORA. Através de programação linear no *software LINGO®*, foi implementado o modelo linear para PCCND, para duplicar arestas, tornando o grafo Euleriano, a partir do qual foi possível extrair a rota ótima, utilizando o algoritmo de Hierholzer. Na ferramenta Solver do *Excel* foi executado o modelo linear para o problema do caminho mínimo para um subconjunto de vértices, no intuito de encontrar a melhor rota para ser percorrida.

2. Fundamentação teórica

Situações cotidianas, como por exemplo, viagens e passeios, sugerem a utilização de mapas como um auxílio. Esses mapas, por sua vez, carregam informações de rotas, caminhos e distâncias disponíveis para chegar ao destino que se deseja, podendo fornecer várias opções para a mesma situação. O estudo da Teoria dos Grafos pode ser um auxiliar a entender e resolver questões como as comentadas acima.Dentre as aplicações para a Teoria dos Grafos,estão seus problemas clássicos de otimização combinatória como, por exemplo,o PCV, Problema do Caixeiro Viajante e o PCC, Problema do Carteiro Chinês. O PVC consiste em encontrar um circuito que possua a menor distância, começando por uma cidade qualquer e a ela retornando visitando cada cidade apenas uma vez. Já o PCC encontra a menor distância a ser percorrida tendo como restrição percorrer todas as arestas do grafo.

O percurso de arcos é o um problema antigo relacionado a grafos, sendo que alguns autores como Bonchev e Rouvray (1991), relatam que o primeiro registro conhecido da Teoria dos Grafos foi feito

pelo matemático suíço Leonhard Euler, com o problema das sete pontes de Königsberg, ilustrado na Figura 1. Neste problema, procurava-se saber se havia um caminho fechado[1] que atravessasse exatamente uma vez cada uma das sete pontes sobre o rio Pregel em Königsberg, hoje Kaliningrado, Rússia. O problema foi solucionado pelo próprio Euler, que encontrou as condições para a existência de um percurso fechado (grafo Euleriano), e que demonstrou que não havia solução que satisfizesse aquele caso particular, nascendo assim a Teoria
dos Grafos.



Fonte: GUEDES (2004)

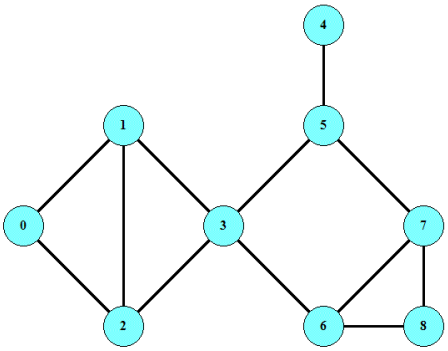
2.1Grafos

2.1.1 Conceitos preliminares

Um grafo $G(V, A)$ não orientado é definido pelo par de conjuntos V e A, não vazios, sendo $V(G)$ o conjunto dos vértices, nodos ou nós do grafo e $A(G)$o conjunto das arestas ou arcos do grafo. Cada aresta está associada a dois vértices, ou seja, uma aresta é um conjunto de pares não ordenados $a = (v, w)$, onde $v, w \in V$. Denota-se por $n(V)$ o número de vértices de Ge por $n(A)$ o número de arestas de G. A Figura 2 é um exemplo de um grafo, no qual tem-se que $n(V) = 9$ e $n(A) = 12$.

Figura 2. Um exemplo de grafo

[1] Percurso ou caminho fechado: Trajeto no qual o vértice inicial e o final são iguais.



Fonte: Próprio autor

Seja um grafo $G=(V,A)$ não orientado. Um vértice v é vizinho de um vértice w se existir em G uma aresta (v,w). A vizinhança, ou vizinhança aberta de v, será denotada por $N(v)$, é o conjunto de vértices de G que são vizinhos de v, ou seja, $N(v) = \{w \mid (v,w)$ é aresta em $G\}$.

Um grafo G é dito orientado ou direcionado quando o sentido das ligações entre os vértices é importante, ou seja, $(v,w) \neq (w,v)$.

Em um grafo G não orientado, chama-se grau de um vértice, e escreve-se $gr(v)$, ao número de arestas incidentes a v, ou seja, aquelas arestas que têm v como extremo.

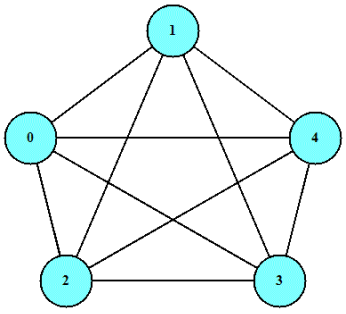
2.1.2 Grafos Eulerianos

Segundo Goldbarg (2000), um dos mais antigos problemas da Teoria dos Grafos é o da determinação de um percurso sobre um grafo **G** que passe por toda aresta de **G** exatamente uma vez. Um grafo pode ser dito Euleriano se existe uma trilha fechada de comprimento m em **G**, ou seja, partindo de um vértice e a ele retornando atravessando cada aresta apenas uma vez. A Figura 3 apresenta um exemplo de grafo Euleriano.

Em 1736, Euler enunciou um teorema, que afirma que um grafo conexo[2] **G** é Euleriano se, e somente se, todos os seus vértices têm grau par.

[2] Um grafo $G(V,A)$ é dito conexo se existe um caminho entre qualquer par de vértices.

Figura 3: Exemplo de grafo Euleriano



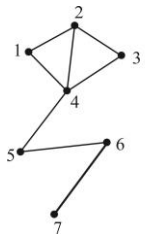
Fonte: Próprio autor

2.1.3 Representação de um grafo

Existem muitas formas de se organizar os dados de um grafo, de modo que eles possam ser introduzidos em um computador. A mais intuitiva delas consiste em dizer, para cada vértice, quais outros vértices estão ligados a ele (ou adjacentes a ele). Isto esta ilustrado nas alíneas (a) e (b) do Quadro 1.

Quadro1 - Representação de grafos orientados e não orientados por diagrama e matriz

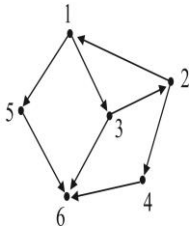
a) Grafo não orientado



Vértice	Vért. adjacentes
1	2, 4
2	1, 3, 4
3	2, 4
4	1, 2, 3, 5
5	4, 6
6	5, 7
7	6

b) Grafo Orientado

Vértice	Vért. adjacentes
1	3, 5
2	1, 4
3	2, 6
4	6
5	6
6	----



c) Matriz de Adjacência

	1	2	3	4	5	6	7
1	0	1	0	1	0	0	0
2	1	0	1	1	0	0	0
3	0	1	0	1	0	0	0
4	1	1	1	0	1	0	0
5	0	0	0	1	0	1	0
6	0	0	0	0	1	0	1
7	0	0	0	0	0	1	0

d) M. A^3 de um grafo orientado

	1	2	3	4	5	6
1			2		1	
2	4			3		
3		1				7
4						6
5						1
6						

Fonte: BOAVENTURA NETTO e JURKIEXICZ (2009)

[3] M.A : Matriz de adjacência.

Matriz de Adjacência consiste em uma matriz que representa os dados de um grafo. Definição: dado um grafo G=(V, E), a matriz de adjacências M é uma matriz de ordem |*v*| x |*v*|, tal que:
|v|= número de aresta.
M[*i*,*j*]= 1, se existir aresta de *i* a *j* .
M[*i*,*j*]=0, se não existir aresta de *i* a *j*.

2.2O Problema do Carteiro Chinês

Em 1962, o matemático chinês Kwan Mei-Ko, com sua experiência e passagem como funcionário dos correios durante a revolução cultural chinesa, modelou algo parecido com o que Euler fez nas pontes de Königsberg, transpondo a situação para os carteiros da cidade. O problema estudado por Kuan Mei-Ko consistia em encontrar a menor distância de percurso para o carteiro, com as restrições de que todas as ruas devem ser visitadas e a distância percorrida pelo carteiro deve ser a mínima possível. A diferença principal entre o problema proposto por Mei-Ko e o problema das sete pontes de Königsbrg, é que no problema do carteiro é permitido passar mais que uma vez pela mesma aresta (rua). Ao fim do percurso, deve-se estar no ponto de partida. Este problema recebe o nome de Problema do Carteiro Chinês, PCC (KWAN, 1962; GRIBKOVSKAIA *et al*., 2007).
Algumas das aplicações do PCC, segundo Konowalenko (2011):

- Serviço de transporte escolar;
- Serviço de entregas de cartas e encomendas de correios;
- Limpezas de ruas usando varredores mecânicos;
- Leitura de medidores de consumo de água, energia, gás, entre outras.
- Coleta de lixo domiciliar.
- Distribuição de alguns produtos de consumo em larga escala, como água mineral, refrigerantes, leite, jornais, correspondências, etc.

2.3Algoritmos de Caminho Mínimo

Para Said (2007, p. 07), algoritmo é "[...] *uma sequência finita de passos, descritos em uma ordem lógica, que visam a atingir um objetivo bem definido*". Os algoritmos de caminho mínimo para um grafo têm por objetivo determinar a rota de menor tempo, distância ou custo entre um par de vértices.

Os algoritmos de menor caminho - ou caminho mínimo - são aplicados sobre grafos ponderados[4], em que se deseja achar o menor caminho entre dois nós. O caminho ótimo é o que representa uma sequência de arcos conectando o vértice de origem e o vértice de destino de tal forma que a soma dos valores no caminho é minimizada.

O valor do menor caminho consiste na soma dos pesos das arestas de um grafo a partir de um determinado vértice (GUIMARÃES, 2004).

Existem diferentes formulações para um problema de caminho mínimo que são:

a) de um vértice para outro vértice;

b) de um vértice para todos os outros vértices do grafo;

c) entre todos os vértices do grafo e

d) K- caminhos mínimos entre dois vértices.

2.4Algoritmos e programação linear

2.4.1 Algoritmo de Dijkstra

O algoritmo de Dijkstra foi criado pelo cientista da computação Edsger Wybe Dijkstra em 1952. Para Goodrich e Tamassia (2007), ao se tentar descobrir o caminho mais curto de um vértice inicial a qualquer outro ponto do grafo, o algoritmo de Dijkstra é uma das primeiras opções. Esse algoritmo é simples e apresenta resultados satisfatórios. O algoritmo de Dijkstra fornece a menor distância entre dois vértices, v_0 e v_i, para um grafo (orientado ou não orientado) cujas arestas têm pesos não negativos. O algoritmo de Dijkstra tem como objetivo encontrar o menor caminho entre dois vértices, porém, para encontrar a distância entre outros dois vértices deve-se aplicar novamente o algoritmo. O tempo computacional para o algoritmo de Dijkstra é $O([n(A)+n(V)]\log(n(V))$.

2.4.2 Algoritmo de Floyd

O algoritmo de Floyd ou Floyd-Warshall foi desenvolvido por Bernard Roy, Stephen Warshall e Robert Floyd em 1962. Esse algoritmo tem por objetivo encontrar o menor caminho entre todos os pares de vértices de um grafo valorado. O algoritmo encontra apenas os valores de tais caminhos e não a sequência que deve ser percorrida. O algoritmo de Floyd gera como saída a matriz de distâncias

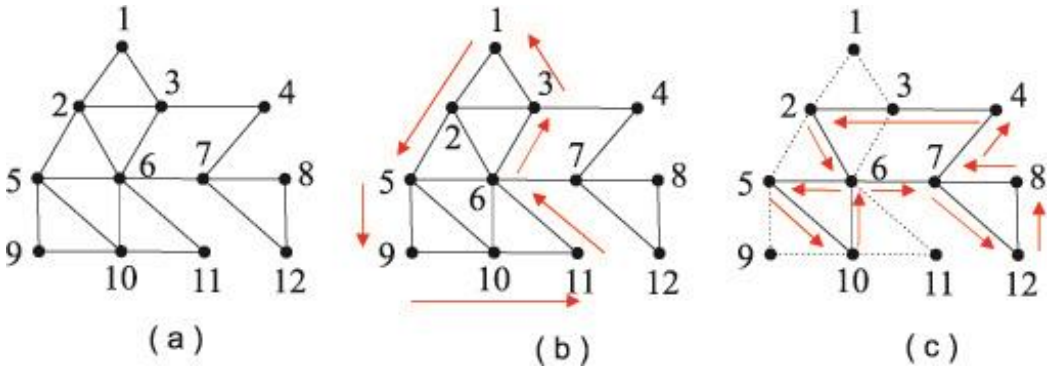
[4] Grafo Ponderado: aquele no qual as arestas possuem peso ou valor.

que contém os valores do menor caminho (μ_{ij}) entre cada par de vértices (i,j) e a matriz de roteamento final (FARBEY *et al*., 1967; LAND &STAIRS, 1967).

2.4.3 Algoritmo de Hierholzer

Proposto em1873 por Carl Hierholzer, o algoritmo de Hierholzer é aplicado em grafo euleriano com objetivo de determinar um circuito euleriano. A ideia é partir de um vértice inicial v_0, percorrer as arestas do grafo e retornar ao vértice inicial v_0.Porém, pode ser obtido um ciclo que não inclua todas as arestas do grafo. Portanto, enquanto houver um vértice que possui arestas ainda não exploradas, comece um caminho neste vértice e tente voltar a ele, usando somente arestas ainda não percorridas. Para o entendimento do algoritmo de Hierholzer, considere o grafo da Figura 4alínea (a).Começando pelo vértice 1, e escolhendo aleatoriamente um aresta nunca visitada a cada vértice visitado, até voltar ao vértice 1. A Figura 4alínea (b) mostra um circuito obtido, que consiste na sequência 1, 2, 5, 9, 10, 11, 6, 3 e 1. Como nesta sequência ainda restam arestas a sem serem percorridas, recomeçamos a partir de um vértice desse circuito. Escolhendo o vértice 6, podemos obter, como ilustrado na Figura 4alínea (c), o circuito 6, 7, 12, 8, 7, 4, 3, 2, 6, 5, 10 , 6. Combinando esse circuito com o anterior,obtemos um novo circuito 1, 2, 5, 9, 10, 11, 6, 7, 12, 8, 7, 4, 3, 2, 6, 5, 10, 6, 3, 1. Esse circuito passa por todas as arestas, portanto esse é o circuito euleriano procurado.

Figura 4. Algoritmo de Hierholzer



Fonte: KONOWALENKO (2011)

2.4.4 Programação Linear

A programação linear é representada por equações e funções lineares tendo como aplicação o apoio à decisão, ocorre na condição quando se decide atingir um objetivo. Tem por resultado a alocação

ótima de recursos, por isso a programação linear é caracterizada como uma técnica de otimização. Segundo Gomes e Ribeiro (2004, p. 59) "*a programação linear lida basicamente com o problema de alocar recursos escassos a atividades que por eles "competem" entre si, e cujo modelo se representa por meio de expressões lineares*".

Em linhas gerais, a programação linear busca, entre as inúmeras tarefas ou atividades, descobrir a melhor distribuição dos recursos a fim de obter um valor ótimo do objetivo desejado (ANDRADE, 2007). Na PL (programação linear) um grafo é chamado de rede, um vértice de nó e uma aresta de arco.

A rede contém nós enumerados de 0 a *n*. As variáveis x_{ij} representam a quantidade de fluxo ao longo do arco não direcionado (*i,j*). Cada arco *(i,j)* tem um comprimento ou distância c_{ij} entre os nós *i* e *j*.

No Quadro 2 é apresentado os modelo lineares para o problema do caminho mínimo.

Quadro 2 . Modelo Linear para o problema do caminho mínimo.

Modelo:

$$\min C = \sum_{i=0}^{n}\sum_{j=0}^{n} C_{ij}\, x_{ij} \qquad 1.1$$

Sujeito a

$\sum_{i=0}^{n} x_{ij} \sum_{j=0}^{n} x_{kj} \cdots$, para nós intermediários.

$$k = 1,2,\dots,n-1 \text{(conservação do fluxo)} \qquad 1.2$$

$$\sum_{j=0}^{n} x_{0j} = 1\ (exigência\ de\ percurso) \qquad 1.3$$

$$\sum_{i=0}^{n} x_{in} = 1\ (exigência\ de\ percurso) \qquad 1.4$$

$$x_{ij}\ \epsilon\ (0,1) \qquad 1.5$$

Fonte: GOMES (2009)

No modelo matemático do caminho mínimo, a função objetivo (1.1) minimiza o custo total, ou seja, a distância total a ser percorrida para minimizar a rota. As restrições em (1.2) garantem a conservação do fluxo, as restrições em (1.3) exigência de percurso, em (1.4) exigência do percurso e, em (1.5) tem-se que as variáveis do problema são binárias.

No Quadro 3 é apresentado o Modelo Linear para o Problema do Carteiro Chinês não orientado.

Quadro 3 . Modelo Linear para o problema do carteiro chinês

A formulação matemática do PCCND, segundo, é apresentada a seguir:

$$\textbf{min} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij} \qquad \textbf{2.1}$$

Sujeito a

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = 0, \qquad i = 1, \ldots, n. \qquad 2.2$$

$$x_{ij} + x_{ji} \geq 1, \forall\ (i,j)\ \epsilon\ A \qquad 2.3$$

$$x_{ij} \geq 0\ e\ inteiras. \qquad 2.4$$

Fonte: BODIN *et al*. (1983)

No modelo matemático do PCCND, a função objetivo (2.1) minimiza o custo total, ou seja, a distância total a ser percorrida. As restrições em (2.2) garantem a continuidade da rota, as restrições em (2.3) que nenhum arco deixará de ser visitado e, em (2.4), tem-se que as variáveis do problema são inteiras e não negativas.

3. Material e Métodos

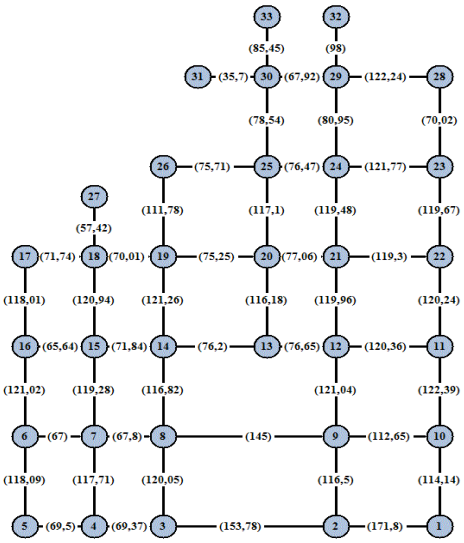
3.1 Material

Para este estudo foi utilizado um grafo que representa um bairro real. Para a implementação dos algoritmos foi utilizado como referência o mapa da Figura 5, obtendo o grafo referente às ruas do bairro Sagrado Coração de Jesus de São Miguel do Iguaçu. O grafo em questão apresenta 33 vértices e 50 arestas. Tanto o mapa como as distâncias entre cada um dos vértices foram obtidos por meio do *Google Maps*, serviço de pesquisa e visualização de mapas e imagens de satélites.

Figura 5. Bairro Sagrado Coração – São Miguel do Iguaçu - Pr

a) Mapa Bairro

b) Grafo do Bairro



Fonte: (a) Google Maps; (b) Próprio autor

3.2 Instrumentos computacionais

3.2.1 TORA

O *software* TORA é disponibilizado gratuitamente por Hamdy Taha, autor do livro "Pesquisa Operacional 8a Ed.". Esse *software* permite resolver problemas de programação linear, (PL), proporcionando visualizar o resultado de cada etapa instantaneamente. Neste *software* é possível acompanhar cada etapa dos cálculos, para os quais é utilizado o método Simplex[5], além de haver a possibilidade de ordenar quais variáveis entram e saem da base com correção automática de cada nova ação do usuário (TAHA, 2008).

3.2.2 *LINGO*®

O Lingo é um *software* interativo para resolução de problemas de programação de modelos lineares, quadráticos e inteiros. Neste projeto, utilizou-se o *LINGO*® para implementar o modelo linear do PCCND.

[5] Método Simplex é um algoritmo para resolver problemas de Programação Linear. Consultar: Marins, F. A. S. **Introdução à Pesquisa Operacional**. Editora Cultura. 2011, 176p.

3.2.3 Solver

O Solver faz parte de um pacote de programas algumas vezes chamado de ferramentas de teste de hipóteses. Com o Solver, você pode encontrar um valor ideal (máximo ou mínimo) para uma fórmula em uma célula – chamada de célula de objetivo – conforme restrições, ou limites, sobre os valores de outras células de fórmula em uma planilha. O Solver trabalha com um grupo de células, chamadas variáveis de decisão ou simplesmente de células variáveis, que participam do cálculo das fórmulas nas células de objetivo e de restrição. O Solver ajusta os valores nas células variáveis de decisão para satisfazer aos limites sobre células de restrição e produzir o resultado que você deseja para a célula objetiva[6].

Pode ser escolhido qualquer um dos três seguintes algoritmos ou métodos de solução na caixa de diálogo "Parâmetros do Solver":

- Gradação Reduzida Generalizada (GRG) Não Linear: usada para problemas simples não lineares.
- LP Simplex: usada para problemas lineares.
- Evolucionário: usada para problemas complexos.

3.3 Metodologia

Primeiramente, foi utilizado o *software* TORA para a implementação dos algoritmos Dijkstra e Floyd. Por meio desses algoritmos, foram calculados os custos de percorrer as ruas desse bairro, partindo de um vértice fixo, v_0, até um vértice qualquer, v_i, com a diferença que o algoritmo de Floyd apresenta a matriz de distâncias e a matriz de roteamento final, enquanto o algoritmo de Dijkstra fornece apenas a menor distância entre esses vértices. Ainda no *software* TORA, verificou-se a rota mínima para percorrer o grafo entre dois vértices predeterminados. A seguir realizou-se a implementação do modelo linear PCCND, obtendo-se a matriz de adjacência. Para essa etapa utilizou-se programação linear no *software LINGO*®, com o algoritmo apresentado no Quadro 4.

Quadro 4. Implementação do Modelo Linear do Problema do carteiro Chinês não direcionado para obtenção da matriz de adjacência e custo total da rota

```
ALGORITMO DO PCCND DESENVOLVIDO NO LINGO®

SETS:
PONTOS /1..33/: ;
```

[6] Disponível em <https://support.office.com/>

```
ROTAS(PONTOS, PONTOS): d,x;
ENDSETS
DATA:
d=@ole ('Mat1.xls','distancias');
ENDDATA
! função objetivo;
MIN = FO;
FO = @SUM(ROTAS(i,j):d(i,j)*x(i,j));
! restrições;
@FOR(PONTOS(k):@SUM(ROTAS(i,k)|d(i,k)#NE#0:x(i,k))=
@SUM(ROTAS(k,j)|d(k,j)#NE#0:x(k,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)));

DATA:
@ole('Mat1.xls','FO','x')=FO,x;
ENDDATA

END
```

Fonte: Adaptado de MORO (2014).

Com o algoritmo de Hierholzer baseada nas informações da implementação do PCCND no Lingo, determinou-se o caminho euleriano para o mesmo problema.

Utilizando a ferramenta Solver do Excel, foi encontrado o caminho mínimo entre dois vértices de um subgrafo, utilizando programação linear.

4. Resultados e discussões

4.1 Análises dos resultados através do *software* TORA

4.1.1 Algoritmo Dijkstra

O algoritmo Dijkstra forneceu os valores entre um vértice inicial fixov_0e um vértice final v_i. O resultado apresentado na Tabela 1é o melhor caminho a ser percorrido na situação do carteiro precisar entregar uma determinada correspondência no último vértice, utilizando como referência sair do vértice v_1 e chegar ao vértice v_{33}. Para a inclusão dos dados na matriz, foram omitidos os valores em centímetros, visto que o *software* trabalha apenas com valores inteiros. Para a sequência de vértices a ser percorrida, o algoritmo vai selecionando a cada iteração o menor caminho entre as opções de

vértices. A Tabela 1 abaixo mostra parte do resultado das iterações. É possível verificar que são apresentadas apenas a distâncias de alguns vértices.

Tabela 1. Algoritmo Dijkstra

Vértice	Distâncias	Rota
1	0,0	-
2	171,00	1-2
3	324,00	1-2-3
4	393,00	1-2-3-4
5	462,00	1-2-3-4-5
6	505,00	1-10-9-8-7-6
7	438,00	1-10-9-8-7
8	371,00	1-10-9-8
9	226,00	1-10-9
10	114,00	1-10
11	234,00	1-10-11
12	347,00	1-10-9-12
13	423,00	1-10-9-8-14
14	487,00	1-10-9-8-14
15	557,00	1-10-9-8-7-15
...	...	...
33	828,00	1-10-11-22-23-28-29-30-33

Fonte: Próprio autor

Desta forma o algoritmo Dijkstra executado no *software* TORA definiu como rota ideal para percorrer do v_1 vértice até o v_{33} a sequência que tem 1-10-11-22-23-28-29-30-33, o que corresponde a uma distância total de828 m. Utilizando o *software* TORA, basta alterar o vértice de partida e o vértice de chegada para encontrar o caminho mais curto entre dois vértices. Como o algoritmo calcula apenas a melhor distância para um par predeterminado de vértices, é necessário alterar os dados de vértice inicial e vértice final cada vez que o usuário necessitar. Se o carteiro fizesse o caminho intuitivo de v_1 até v_{33}, ou seja, seguindo a sequência 1-10-11-22-23-28-29-30-33, teria uma distância total de 828 m.

4.1.1 Algoritmo de Floyd

Para a implementação do algoritmo de Floyd, os valores em centímetros também foram omitidos. O *software* mostra como retorno uma matriz $n \times n$, com todas as distâncias mínimas entre dois vértices. A Tabela 2 mostra um resumo dos resultados da matriz das distâncias.

Tabela 2. Matriz de distâncias mínimas apresentadas pelo algoritmo de Floyd, executado no *software* TORA

Vértices	N28	N29	N30	N31	N32	N33
N1	554,00	676,00	743,00	778,30	774,00	**828,00**
N2	668,00	790,00	762,00	797,00	888,00	847,00
N3	810,00	688,00	621,00	656,00	786,00	706,00
N4	874,00	752,00	685,00	720,00	850,00	770,00
N5	942,00	820,00	753,00	788,00	918,00	838,00
N6	824,00	702,00	635,00	670,00	800,00	720,00
N7	757,00	635,00	568,00	603,00	733,00	653,00
N8	690,00	568,00	501,00	536,00	666,00	586,00
N9	552,00	674,00	646,00	681,00	772,00	731,00
N10	440,00	562,00	629,00	664,00	660,00	714,00
N11	318,00	440,00	507,00	446,00	538,00	592,00
N12	438,00	560,00	537,00	572,00	658,00	622,00
N13	514,00	528,00	461,00	496,00	626,00	546,00
N14	574,00	452,00	385,00	420,00	550,00	470,00
N15	645,00	523,00	456,00	491,00	621,00	541,00
N16	710,00	588,00	521,00	556,00	686,00	606,00

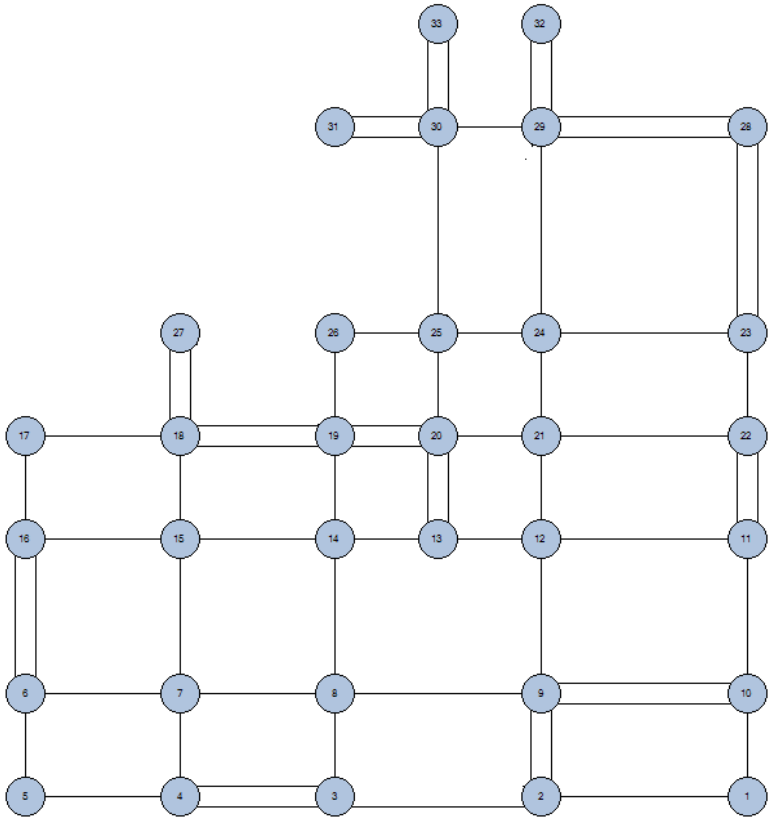
Fonte: Próprio autor

O resultado do algoritmo Floyd se faz mais eficiente que o de Dijkstra pelo fato de mostrar todas as iterações, e a distância entre todos os pares de vértices. No entanto, o resultado do menor caminho a ser percorrido entre o v_1 e o v_{33}é o mesmo apresentando no algoritmo de Dijkstra, ou seja, para ir do v_1 vértice até o v_{33}, o menor caminho totaliza 828,00 m, com a mesma sequência de vértices.

4.2 Análise dos resultados da Programação Linear

Utilizando o *software LINGO*®, através de programação linear para implementar o PCCND, verificou-se a quantidade de vezes que a aresta (i, j) é atravessada, ou seja, quantas vezes o carteiro deve percorrer a mesma rua, pois os vértices devem ser de grau par para que seja possível encontrar um caminho euleriano. Dessa forma, foi construída a matriz de adjacência. A implementação do algoritmo oportunizou passar por todas as arestas, porém não apresenta o caminho euleriano. A função objetivo faz o cálculo da minimização da rota, multiplicando a quantidade de vezes que cada aresta foi percorrida pelo seu custo. Assim, passando por todas as arestas, o resultado da função objetivo é 6315,75 m, ou seja, essa é a menor distância para o carteiro percorrer todas as arestas, partindo do vértice 1 e retornando a ele. Trabalhos como o de Moro (2014), implementaram o algoritmo de Fleury, o qual apresenta o percurso euleriano que o carteiro deve tomar como rota. A Figura 6 mostra o grafo referente à matriz de adjacência retornado pelo *LINGO*®, apresentando as duplicações das arestas.

Figura 6. Grafo Matriz de Adjacência



Fonte: Próprio autor

Observa-se na Figura 6, que 14 arestas foram duplicadas, fazendo com que todas as arestas tenham grau par, tornando o grafo euleriano. Fazendo-se o somatório de todas as arestas, encontra-se o resultado da função objetivo, ou seja, os mesmos 6315,75 m que resultaram da implementação do PCCND no *LINGO*®.

4.3 Resultados Algoritmo de Hierholzer

A partir do resultado apresentado pelo PCCND do *LINGO*®, aplicou-se manualmente o algoritmo de Hierholzer. Na primeira etapa, obtiveram-se 17 subciclos, apresentados na Tabela 2.

Tabela 3. Subciclos gerados na primeira etapa do algoritmo de Hierholzer

Ordem	Rota no subciclo
1	1-10-9-2-1

2	2-9-8-3-2
3	3-4-3
4	4-7-6-5-4
5	6-16-6
6	7-8-14-13-20-19-14-15-7
7	9-12-13-20-21-22-11-10-9
8	11-12-21-24-23-22-11
9	15-16-17-18-15
10	18-19-18
11	18-27-18
12	19-20-25-26-19
13	23-28-23
14	24-25-30-33-30-29-24
15	28-29-28
16	29-32-29
17	30-31-30

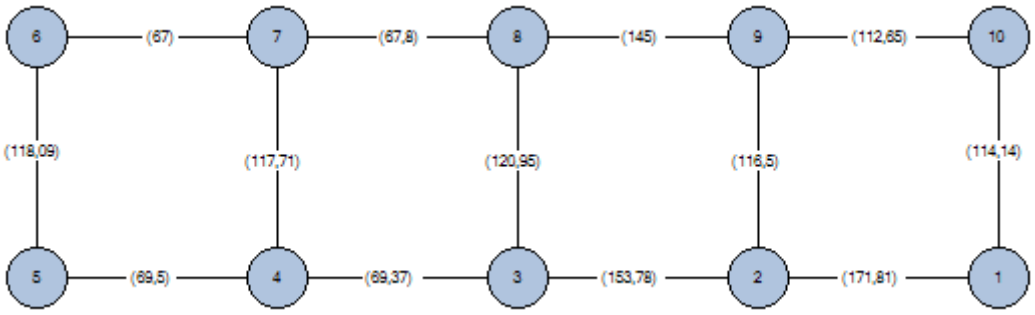
Fonte: Próprio autor

Após aplicar a segunda etapa do algoritmo de Hierholzer, efetuando os encaixes dos subciclos, obteve-se a rota ótima, que confirma o resultado apresentado pelo *LINGO®*, de um percurso de 6315,75 m. A rota que resolve o PCCND é dada pela sequência de vértices: 1-10-9-2-9-12-13-20-21-22-11-12-21-24-25-30-31-30-33-30-29-24-23-28-29-32-29-28-23-22-11-10-9-8-3-4-7-8-14-13-20-19-14-15-16-17-18-19-20-25-26-19-18-27-18-15-7-6-16-6-5-4-3-2-1.

4.4 Resultados da aplicação caminho mínimo na ferramenta Solver

Para a simplificação do problema e melhor apresentação dos resultados, optou-se por aplicaro modelo linear para o problema do caminho mínimo em apenas um subconjunto de 10 vérticesdo grafo. A Figura 7 mostra o subconjunto de vértices escolhido, sendo ele o dos vértices $v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}$.

Figura 7. Grafo de um subconjunto de vértices



Fonte: Próprio autor

Supôs-se a situação de sair do vértice v_1 com o objetivo de chegar ao vértice v_6. Na Figura 8, observam-se todas as distâncias entre dois vértices, x_{ij}. O valor 1 representa a ida até o vértice, e -1 a volta.

Figura 8. Resolução Solver para a minimização de rota

	x12	x21	x1,10	x10,1	x2,3	x3,2	x2,9	x9,2	x3,4	x4,3	x3,8	x8,3	x4,5	x5,4	x4,7	x7,4	x5,6	x6,5	x6,7	x7,6	x7,8	x8,7	x8,9	x9,8	x9,10	x10,9	fo	
	171,81	171,81	114,14	114,14	153,78	153,78	116,5	116,5	69,37	69,37	120,95	120,95	69,5	69,5	117,11	117,11	118,04	118,04	67	67	67,8	67,8	145	145	112,65	112,65	506,53	
nó 2	1	-1			-1	1	-1	1																			0	0
nó 3					1	-1			-1	1	-1	1															0	0
nó 4									1	-1			-1	1	-1	1											0	0
nó 5													1	-1			-1	1									0	0
nó 10			1	-1																					1	-1	0	0
nó 9							1	-1															1	-1	-1	1	0	0
nó 8												1	-1								1	-1	-1	1			0	0
nó 7															1	-1			1	-1	-1	1					0	0
nó inicial(1)																											1	1
nó final (6)																	1			1							1	1

solução	x12	x21	x1,10	x10,1	x2,3	x3,2	x2,9	x9,2	x3,4	x4,3	x3,8	x8,3	x4,5	x5,4	x4,7	x7,4	x5,6	x6,5	x6,7	x7,6	x7,8	x8,7	x8,9	x9,8	x9,10	x10,9
	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1

Caminho 1-10-9-8-7-6
distan 506,53

Fonte: Próprio autor

A solução do problema do caminho mínimo fez a verificação entre os vérticesv_1 até v_6, tendo como objetivo sair do v_1 e chegar ao v_6. Obteve-se como resultado o percurso de minimização 1-10-9-8-7-6, com distância de 506,59 m.

Supondo que o carteiro utilizasse como rota intuitiva a rota máxima, fez-se uma maximização da rota no Solver. Supondo que o carteiro intuitivamente faça essa escolha, diferente da sugerida pela PL anterior, obtêm-se os resultados apresentados na Figura 9.

Figura 9. Resolução Solver para a maximização de rota

	x12	x21	x1,10	x10,1	x2,3	x3,2	x2,9	x9,2	x3,4	x4,3	x3,8	x8,3	x4,5	x5,4	x4,7	x7,4	x5,6	x6,5	x6,7	x7,6	x7,8	x8,7	x8,9	x9,8	x9,10	x10,9	fo	
	171,81	171,81	114,14	114,14	153,78	153,78	116,5	116,5	69,37	69,37	120,95	120,95	69,5	69,5	117,11	117,11	118,04	118,04	67	67	67,8	67,8	145	145	112,65	112,65	582,5	
nó 2	1	-1			-1	1	-1	1																			0	0
nó 3					1	-1			-1	1	-1	1															0	0
nó 4									1	-1			-1	1	-1	1											0	0
nó 5													1	-1			-1	1									0	0
nó 10			1	-1																						1	-1	0	0
nó 9							1	-1															1	-1	-1	1	0	0
nó 8											1	-1										1	-1	-1	1		0	0
nó 7															1	-1			1	-1	-1	1					0	0
nó inicial(1)																											0	1
nó final (6)																	1			1							1	1

solução	x12	x21	x1,10	x10,1	x2,3	x3,2	x2,9	x9,2	x3,4	x4,3	x3,8	x8,3	x4,5	x5,4	x4,7	x7,4	x5,6	x6,5	x6,7	x7,6	x7,8	x8,7	x8,9	x9,8	x9,10	x10,9
	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0

Caminho	1-2-3-4-5-6
distan	582,5

Fonte: Próprio autor

O caminho de maximização é dado pela sequência 1-2-3-4-5-6,totalizando 582,50 m. Comparando a rota máxima com a rota mínima, verifica-se um aumento de 75,91 metros, que representa um acréscimo de 14,98% sobre o percurso. Portanto, a utilização dos recursos computacionais para os algoritmos de rota, mostrou-se mais eficiente que os caminhos intuitivos.

5. Conclusão

A execução deste trabalho mostrou-se importante, dadas as aplicações dos algoritmos de rotas em situações práticas. A redução de custos e recursos é uma necessidade eminente da sociedade, visto que além de maximizar os lucros das empresas envolvidas, proporciona grandes vantagens ao meio ambiente uma vez que tem como intenção a minimização dos impactos sociais, com a redução de emissão de poluentes uma vez que procura uma rota mais eficiente. O projeto teve por objetivo apresentar alguns algoritmos que são utilizados para encontrar esses caminhos de custos mínimos. O *software* TORA proporcionou verificar as distâncias entre os vértices através do algoritmo de Dijkstra e Floyd, fazendo uma análise de ambos. Verificou-se que o rota de menor custo proposto pelos dois algoritmos através do *software* foram iguais. Na programação linear para a implementação do PCCND a obtenção da matriz de adjacência foi satisfatória que possibilitou encontrar o caminho Euleriano através do algoritmo de Hierholzer. O problema do caminho mínimo para o subconjunto de vértices proporcionou a verificação dos benefícios da utilização da tecnologia para minimizar os custos.

Coloca-se como sugestão para trabalhos futuros implementar computacionalmente o algoritmo de Hierholzer, bem como algoritmos que resolvem o mesmo problema, como o de Fleury. Também sugere-se ampliar os estudos sobre o Problema do Carteiro Chinês, incluindo as outras vertentes, que resolvem problemas para grafos orientados.

Abstract: This study work to present and implement some methods that solve the problem of minimizing routes and their computational instruments in which the implementation of such methods is possible. To contextualize this problem, we used the route of a postman in a municipal district of São Miguel do Iguaçu, western Paraná. Held previously a study of graph theory, the Chinese Postman Problem (CCP) and basics of linear programming. In this paper we analyzed the Dijkstra and Floyd algorithms and route algorithms using linear programming and Hierholzer algorithm. Excel software was used, Lingo and TORA to implement such algoritmos. Found is that the use of technology has facilitated the implementation of algorithms and it was possible to optimize the route carried by the postman.

Key-words: graph theory**,** Floyd algorithm, Chinese postman problem.

Referências

ANDRADE, E. L. **Introdução à pesquisa operacional: métodos e modelos para a análise de decisão**. 3ª ed. -. Rio de Janeiro: Livros Técnicos e científicos, 2004, 192 p.

BOAVENTURA NETO. P. O.; JURKIEVICZ, S. **Grafos: Introdução e prática**. São Paulo: Blucher, 2009.

BODIN, L.; GOLDEN, B.; ASSAD, A.; BALL, M. Routing and Scheduling of vehicles and crews: the state of the art. England, **Pergamon Press**, vol. 10. n. 2, 1983 (Special Issue).

BONCHEV, D.; ROUVRAY, D. H. **Chemical Graph Theory: introduction and fundamentals.**1. ed. Amsterdam: Gordon and Breach Science Publishers S. A, 1991.

FARBEY, B. A.; LAND, A. H.; MURCHLAND, J.D. The Cascade algorithm for finding all shortest distances in a directed graph. **Management Science**, 1967, v. 14, p.19-28.

GERSTING, J. L.**Fundamentos matemáticos para a ciência da computação: um tratamento moderno de matemática discreta**. Rio de Janeiro: LTC,2013.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear : modelos e algoritmos**. Rio de Janeiro. Campus, 2000.

GOMES, C. F. S.; RIBEIRO, P. C. C. **Gestão da Cadeia de Suprimentos integrada à Tecnologia da Informação**. São Paulo : Pioneira Thomson Learning, 2004.

GOMES, M. J. N. O problema do carteiro chinês, algoritmos exatos e um ambiente MVI para análise de suas instâncias: sistema XNÊS. Pesquisa Operacional,Rio de Janeiro , v. 29, n. 2, p. 323-363, Aug. 2009.

GOODRICH, M. T; TAMASSIA, R. **Estrutura de dados e algoritmos em Java**. 4. ed. – Porto Alegre : Bookman, 2007.

GUEDES, A. L. P. **Algoritmos e Teoria dos Grafos**. Disponível em http://www.inf.ufpr.br/andre/Disciplinas/BSc/CI065/ Acessado em 20 de setembro de 2015.

GUIMARÃES, J. O.**Teoria dos Grafos**. Disponível <http://www.dc.ufscar.br/jose/>Acessado em 20 de agosto de 2015.

JUKIEWICZ, S. **Grafos: uma introdução**. OBMEP. Rio de Janeiro. 2009.

KONOWALENKO, F. **Problema do carteiro chinês não-orientado e misto para a otimização de rotas na cidade de Irati/PR. 2011**. 108 f. Dissertação de Mestrado, Universidade Federal do Paraná, Curitiba-PR, 2011.

KWAN, M. K. Graphic programming using odd and even points. **Chinese Mathematics**, 1, 1962, p. 273-277.

MARIANI, A. C. **Teoria dos Grafos**. UFSC/ CTC/ INE. Disponível em:<http://www.inf.ufsc.br/grafos/temas/custo-minimo/dijkstra.html>.Acesso em 21-04-2015.

MIRANDA, C. L.; DAUDT, C.G.**Análise da complexidade do algoritmo de Floyd-Warshall.** Instituto de informática, Universidade Federal do Rio Grande do Sul: 2010

MORO, F. M. **O problema do carteiro chinês aplicado na otimização de rotas usadas na coleta de lixo reciclável: um estudo de caso**. Universidade Tecnológica Federal do Paraná – Medianeira - PR- 2014.

PAES, F.G.**Otimização de rotas para a coleta do lixo doméstico: um tratamento grasp do problema do carteiro Chinês. (PCCM).**. 116 f. Universidade Estadual do Norte Fluminense, Campos dos Goytacazes, RJ: 2004

RADUAN, A. C. **Roteirização parcialmente dinâmica aplicada a serviços de campo.** Escola Politécnica de São Paulo (Dissertação de Mestrado). São Paulo, 2009.

SAID, R. **Curso de Lógica de Programação**. São Paulo: Digerati Books, 2007.

SHARP, J. **Microsoft Visual C# 2008: passo a passo / John Sharp**. Porto Alegre:Bookman, 2008.

TAHA, H. A. **Pesquisa Operacional: uma visão geral**, 8ª ed., Pearson Prentice Hall, São Paulo, 2008.