ISSN 0717-9103
ISSN Online 0718-8307
Universidad del Bío-Bío

APLICAÇÃO E ANÁLISE DE ALGUNS PROCEDIMENTOS DE CONTRUÇÃO DE ROTA PARA O PROBLEMA DO CAIXEIRO VIAJANTE♠

APPLICATION AND ANALYSIS OF SOME PROCEDURES FOR THE CONSTRUCTIONS OF ROUTES TO TRAVELING SALESMAN PROBLEM

Paula Francis Benevides[1❖],Flavia Konowalenko[2], Deise Maria Bertholdi Costa[2], Luiz Fernando Nunes[1], Angela Olandoski Barboza[1]

RESUMO

O transporte, em geral, absorve em média a porcentagem mais elevada de custos do que qualquer outra atividade logística. Por isso, muitas empresas estão repensando seus processos para redução dos mesmos. A otimização da distribuição de produtos é um problema estudado há muito tempo por pesquisadores de diversas áreas. Este tipo de problema é classificado como de otimização combinatória. Dentre as modelagens podem ser citados o Problema do Caixeiro Viajante (PCV) e o Problema de Roteamento de Veículos (PRV). Estes têm por objetivo encontrar o menor caminho conectando-seNlugares de destino. O presente trabalho visa analisar e comparar, em termos de desempenho computacional e qualidade das soluções obtidas, as principais heurísticas de construção de rotas, além de um Algoritmo Genético para o PCV. Também aplicou-se o algoritmo 2-opt para melhoria das rotas geradas. Foram utilizados dados reais de uma distribuidora de produtos em uma determinada região da cidade de Curitiba (PR), Brasil. As coordenadas geográficas dos pontos de visitação foram extraídas do aplicativo online Google Earth, as quais foramconvertidas em coordenadas cartesianas, para posterior aplicação dos algoritmos utilizados. Os resultados obtidos foram comparados com as rotas reais que estão sendo utilizadas por um determinado representante da referida distribuidora.

Palavras chave: Problema do Caixeiro Viajante, Problemas de Roteamento de Veículos, Heurísticas, Algoritmo Genéticos.

ABSTRACT

The transport in general, absorb on average the highest percentage of costs than any other logistics activities. Therefore,many companies are rethinking their processes to reduce them. The optimization of product distribution is a much studied problem time by researchers from several areas. This type of problem is classified as combinatorial optimization. Among the modeling can be cited the Traveling Salesman Problem (TSP) and the problem of Vehicle Routing (PRV). These, aims to find the lowest path connecting N places of destination. The

♠IX Congreso del Instituto Chileno de Investigación Operativa, 26-29 de octubre, Pucon. Chile

[1]Dpto. de Matemática, Universidade Tecnológica Federal do Paraná - UTFPR. Curitiba, Paraná. Brasil.paula_benevides@hotmail.com, nunes@utfpr.edu.br, aobarboza@yahoo.com
[2]Programa de Métodos Numéricos em Engenharia. Universidade Federal do Paraná. Centro Politécnico, Jardim das Americas, C. P. 19011.Curitiba, Paraná. Brasil. flavia.konowalenko@hotmail.com, deise@ufpr.br.

Autor correspondente: ❖paula_benevides@hotmail.com

Recebido: 14.09.2011 Aceito: 27.04.2012

present work analyze and compare, in terms of computational performance and quality of solutions, the main route construction heuristics, and a genetic algorithm for the TSP. We also applied the algorithm 2-opt to improve the routes generated. We used real data a distributor of products in a certain region of the city of Curitiba (PR), Brazil. The geographical coordinates of places to visit were taken from the online application Google Earth, which were converted to Cartesian coordinates for application of algorithms used. The results were compared with the actual routes being used by a particular representative distributor of that.

Keywords: Traveling salesman problem, vehicle routing problem, heuristics, genetic algorithm.

INTRODUÇÃO

O Problema do Caixeiro Viajante (PCV) consiste em estabelecer uma única rota que passe por cada nó de um grafo, uma e apenas uma vez, retornando ao nó inicial no final do percurso. Este roteiro Hamiltoniano deve ser feito de modo que a distância total percorrida seja mínima. O conjunto de rotas possíveis é o resultado de todas as combinações possíveis e pode ser calculado por $(n-1)!$, sendo n o número de nós. Este problema pertence à classe de problemas conhecida por NP-Hard, isto é, não existem algoritmos com limitação polinomial capazes de resolvê-lo. Assim a quantidade de passos de um algoritmo que possa solucioná-lo otimamentenão pode ser dada por uma função polinomial do tamanho de sua entrada. Logo,apenas os problemas de pequeno porte podem ser solucionados de forma ótima. Problemas maiores tornam-se inviáveis através dos métodos exatos,haja vista o esforço computacional que seria exigido para resolvê-los. Muitas abordagens de algoritmos heurísticos, que fornecem soluções factíveis próximas da ótima, têm sido desenvolvidas para resolver os problemas NP-Hard. Apresentando soluções parciais e ótimas para o problema, visto que, devido ao grande número de grandezas que influenciam o processamento computacional, como a capacidade de carga, velocidade, número de veículos, tempo e distância, o mesmo precisaria de uma grande capacidade computacional para apresentar soluções exatas.

Formulação Matemática para o PCV

Seja o grafo G (N, A) onde N representa o conjunto de vértices $(|N| = n)$ e A o conjunto de arcos. Se for admitido que os custos ou distâncias mínimas entre os nós da rede são dados pela matriz $C = \lfloor c_{ij} \rfloor$ simétrica, isto é, considera-se que $c_{ij} = c_{ji}$, assumindo ainda que $c_{ij} = +\infty, \ \forall i \in N$, e considerando a matriz $X = \lfloor x_{ij} \rfloor$ das variáveis de decisão do problema, onde $x_{ij} = \begin{cases} 1, \text{ se o arco } a_{ij} \in \text{rota} \\ 0, \text{ se o arco } a_{ij} \notin \text{rota} \end{cases}$ tem-se a seguinteformulaçãode programação linear inteira para o problema, devida aGolden (1977), (Bodin *et al*., 1983):

Minimizar $\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$

s.a.: $\sum_{i=1}^{n} x_{ij} = 1 \qquad (j = 1, 2, ..., n)$

$\sum_{j=1}^{n} x_{ij} = 1 \qquad (i = 1, 2, ..., n)$

$X = (x_{ij}) \in S$

$x_{ij} = 0 \text{ ou } x_{ij} = 1 \qquad (i, j = 1, 2, ..., n)$

Os dois primeiros grupos de restrições garantem que exatamente um arco (i, j) emana de cada nó da rota e exatamente um arco (i, j) é direcionado para cada nó j da rota. A penúltima restrição contém um conjunto S que pode ser qualquer conjunto de restrições que impeçaa formação de subrotas. Estas restrições são chamadas restrições de quebra de subrotas e podem ser, entre outras:

$$S = \left\{ (x_{ij}) : \sum_{i \in Q} \sum_{j \notin Q} x_{ij} \geq 1, \text{ para todo subconjunto próprio não vazio Q de N} \right\};$$

$$S = \left\{ (x_{ij}) : \sum_{i \in R} \sum_{j \in R} x_{ij} \leq |R| - 1, \text{ para todo subconjunto não vazio R de } \{2, 3, ..., n\} \right\}$$

$S = \{(x_{ij}) : y_i - y_j + nx_{ij} \leq (n-1), \text{ para } 2 \leq i \neq j \leq n \text{ para alguns números reais } y_i\}$, onde $y_i = t$ se o nó *i* é visitado no passo t da rota ou $y_i = 0$, caso contrario.

Heurísticas de Construção de Rotas

Heurísticas de construção de rotas para o Problema do Caixeiro Viajante são algoritmos que geram um circuito viável partindo de conjunto inicial de vértices, e modificando esse conjunto a cada iteração utilizando algum critério de escolha. Este processo busca boas soluções a um custo computacional razoável, porém, sem ser capaz de garantir otimalidade ou até, em vários casos, de estabelecer quão perto de uma dada solução viável esta dada solução ótima. Inicialmente as rotas serão calculadas utilizando as seguintes técnicas de construção de rotas (Bodin *et al*., 1983):

- Procedimento do vizinho mais próximo;
- Inserção do mais próximo;
- Inserção do mais distante;
- Inserção mais rápida.

Existem ainda outros procedimentos de construção de rotas que podem ser citados tais como Inserção do Mais Barato, Inserção Arbitrária, Cobertura Convexa, Inserção do Maior Angulo e outros descritos detalhadamente em Bodin *et al*., (1983).

Após a aplicação dessas técnicas podem ser promovidas Heurísticas de Melhorias de rotas. Dentre as técnicas que se pode utilizar para realizar estas melhorias existem a 2-opt e 3-opt.

Vizinho mais Próximo

É a mais intuitiva das heurísticas e trabalha da seguinte forma: o ciclo inicia com um nó v_i atual e, a partir deste, encontra o próximo nó v_k de forma que a distância entre os dois nós seja mínimo. Ao encontrar este nó v_k, o algoritmo insere o mesmo no final do ciclo e repete a operação até que todos os nós pertençam a solução. Finalmente o ciclo é fechado ligando o nó inicial ao nó final. Não é permitido visitar mais de um nó duas vezes ou modificar a escolha, ou seja, após um nó ser inserido em uma determinada posição da rota, ele não sofrerá nenhuma modificação de posicionamento.

Algoritmo:
P1: *Escolha um nó inicial*
P2: *Encontre o nó mais próximo do último nó adicionado na rota e adicione-o na rota*
P3: *Repita o passo 2*

Inserção do Mais Próximo

Segundo Goldbarg (2005) o algoritmo da inserção do mais próximo é uma heurística que possui um processo onde três níveis de decisão são envolvidos: a escolha do vértice a ser inserido na solução; a posição de inserção desse novo vértice; a decisão de um ciclo inicial.

Algoritmo:
P1: *Inicie com um sub-grafo contendo apenas o nó i*
P2: *Encontre o nó k tal que c_{ik} seja mínima e forme a rota i –k –i*
P3: *Dada a sub-rota, encontre o nó k não pertencente à sub-rota mais próximo de qualquer nó da sub-rota*
P4: *Encontre o arco (i, j) na sub-rota que minimiza* $c_{ik} + c_{kj} - c_{ij}$
Insira k entre i e j
P5: *Volte ao passo 3 até formar um circuito Hamiltoniano*

Inserção do Mais Distante
Idêntica à heurística Inserção do Mais Próximo, exceto no passo 2 onde se escolhe a cidade k não pertencente ao ciclo, mais distante de qualquer cidade do ciclo.

Algoritmo:
P1: *Inicie com um sub-grafo contendo apenas o nói;*
P2: *Encontre o nó k tal que c_{ik} seja máxima a e forme a rota i –k – i;*
P3: *Dada a sub-rota, encontre o nó k não pertencente à sub-rota mais distante dequalquer nó da sub-rota;*
P4: *Encontre o arco (i, j) na sub-rota que minimiza* $c_{ik} + c_{kj} - c_{ij}$
Insira k entre i e j;
P5: *Volte ao passo 3 até formar um circuito Hamiltoniano.*

Inserção Mais Rápida

Algoritmo:
P1: *Tome um nó inicial para formar um circuito T com 1 nó e 0 arcos;*
P2: *Dado o conjunto T_k, ache o nó z_k não pertencente à T_k mais próximo de um nóy_k em T_k*
P3: *Seja T_{k+1} a rota com k + 1 nós inserindo z_k imediatamente em seguida a y_k;*
P4: *Repita P2 e P3 até a formação do circuito Hamiltoniano.*

O tempo de computação envolvido nos algoritmo acima é de ordem n^2.

Heurísticas de Melhorias de Rotas
As heurísticas de melhorias de rota conhecidas por 2-opt e 3-opt, surgiram em meados da década de 60 e consistem em permutar arcos em uma rota inicial factível, buscando encontrar uma rota de menor custo.

Na heurística 2-opt, dois arcos são desligados e substituídos outros doisde modo que a distância total na nova rota formada seja menor que na rota inicial. Analogamente ocorre na heurística 3-opt, onde três arcos são permutados. Estes procedimentos terminam geralmente em um ótimo local, e são considerados métodos eficientes para resolver o Problema do Caixeiro Viajante. Mais tarde foi criada outra heurística ainda mais forte do que a 2-opt e 3-opt. Chama-se heurística k-opt, onde são permutados k arcos (k ≥ 3). Embora a qualidade das soluções desta última técnica melhore à medida que k aumenta, elas se tornam quase impraticáveis sob ponto de vista computacional quando k ≥ 4. Algumas aplicações práticas destas heurísticas podem ser vistas em Costa(1997) e Zamboni (1997).

Os passos do algoritmo para efetuar melhoramentos 2-opt e 3-opt são os seguintes:

Algoritmo:
P1: Obtenha uma rota inicial factível, utilizando algum dos procedimentos de formação de

rotas;

P2: Desligar 2 arcos da rota atual para o caso de 2-opt e 3 arcos para o caso 3-opt, reconectando os nós por meio de arcos diferentes daqueles que foram desconectados, formando uma nova rota. Se o comprimento da rota nova for menor que o comprimento da rota anterior, troque a rota atual pela rota nova;

P3: Prossiga no passo 2 até que nenhuma melhoria possa ser alcançada.

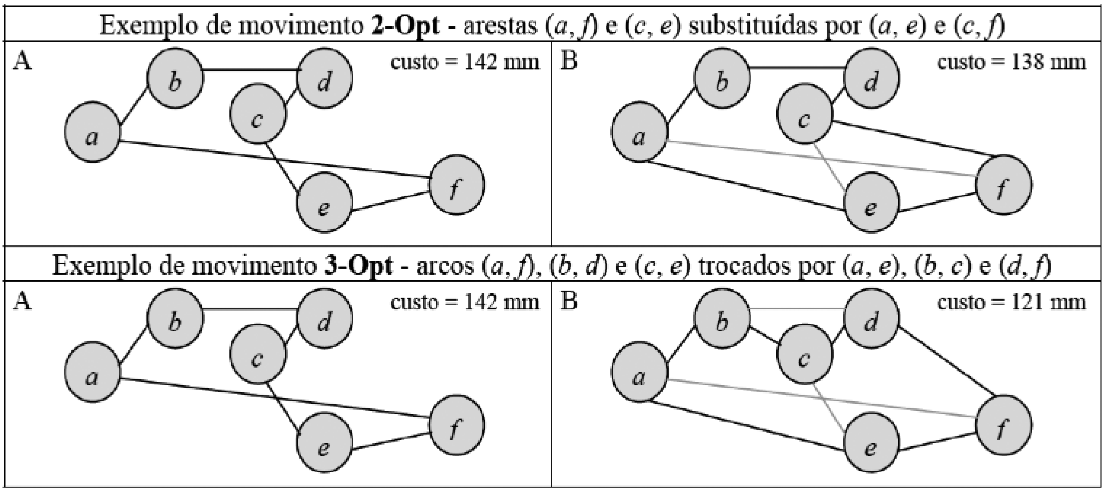

Figura 1. Exemplos de substituição 2-Opt e3-Opt.
(Fonte: Ganhoto, 2004)

Um Algoritmo Genético para o Problema do Caixeiro Viajante

Existem diversas abordagens para o problema do caixeiro viajante utilizando algoritmos genéticos. Elas diferem entre si não apenas na questão dos parâmetros, mas também na forma de representar as soluções viáveis, de selecionar os indivíduos para reprodução, na maneira de definir os operadores genéticos (Whitley et al., 1989). Mayerle (1994) desenvolveu um AG cujos testes revelaram resultados bastante satisfatórios, comparativamente a outras técnicas clássicas para resolver o problema. Este algoritmo também foi utilizado com sucesso em Bezerra (1995).

Considere-se $X = \{x_1, x_2, \ldots, x_n\}$ o conjunto das n cidades pelas quais o caixeiro viajante deve passar, retornando ao ponto de partida no final da jornada. Assumindo que o custo da viagem entre cada par de cidades é conhecido e dado por $w(x_i, x_j), \ \forall x_i, x_j \in X, \ \text{sendo} \ w(x_i, x_j) > 0$, sendo , define-se a estrutura do cromossomo, a função de *fitness* e os operadores de *crossover* e mutação utilizados a seguir.

Estrutura do Cromossomo

Existem várias maneiras de codificação do espaço de busca para o PCV. Dentre as principais representações estão: a ordinal, por caminho (ou inteiros) e por adjacência. Neste trabalho optou-se em utilizar a Representação por Caminho, onde o cromossomo é formado pela sequência dos nós na solução.

Por exemplo, o cromossomo c_1 = {1, 2, 3, 4, 5, 6, 7} representa diretamente a solução r_1{1, 2, 3, 4, 5, 6, 7}. Esta representação é talvez a mais natural de uma solução para o PCV e de simples implementação.

Função de Aptidão

A função de aptidão ou função de *fitness* representa a capacidade de um indivíduo de se adaptar ao meio ambiente. Neste algoritmo adotou-se como *fitness* de um cromossomo r_i, o custo total do circuito, onde n é o número de cidades a ser visitadas.

$$Fitness(r_i) = w(x_n, x_1) + \sum_{j=1}^{n-1} w(x_j, x_{j+1}) \tag{1}$$

Para problemas de minimização, quanto menor for o custo total do circuito, maior será o *fitness* do cromossomo correspondente e consequentemente maiores serão as chances deste indivíduo sobreviver e vir a se reproduzir (Mayerle, 1994).

Processo de Seleção
A principal ideia do processo de seleção é permitir que os indivíduos mais adaptados (melhor *fitness*) tenham maior chance de se reproduzir. Barboza (2005) afirma que a seleção elitista consiste em copiar ou reproduzir os melhores indivíduos da população atual para a próxima geração, garantindo que estes cromossomos não sejam destruídos nas etapas de recombinação e mutação. Sua vantagem é que se no caso ótimo global for descoberto durante o processo de busca, o AG deve convergir para tal solução.

Considerando que a população possui m cromossomos dispostos em ordem crescente dos seus custos, isto é $C_1 \le C_2 \le \ldots\ldots \le C_m$, onde C_i é o custo associado ao cromossomo r_i, a escolha de um indivíduo para ser submetido aos operadores genéticos é feita considerando uma distribuição de probabilidade inversamente proporcional ao índice dos cromossomos. Assim, quanto menor for o índice de um cromossomo, maior é a probabilidade do mesmo ser selecionado. De acordo com esta distribuição, a função de seleção é a seguinte (Mayerle, 1994):

$$\text{Select}(R) = \left\{ r_j \in R \,/\, j = m + 1 - \left\lceil \frac{-1 + \sqrt{1 + 4.\text{Rnd}(m^2 + m)}}{2} \right\rceil \right\}$$

onde $\text{Rnd} \in [0{,}1)$ é um número aleatório uniformemente distribuído, $\lceil b \rceil$ é o menor inteiro maior do que b e $R = \{r_1, r_2, \ldots\ldots, r_m\}$ é o conjunto ordenado dos cromossomos, de modo que $C_1 \le C_2 \le \ldots\ldots \le C_m$ e $C_i = Fitness(r_i)$.

Operador de Cruzamento
De acordo com Potvin (1996) existem diversos operadores de cruzamento desenvolvidos para manipular as representações descritas acima e estão classificados em: os que preservam a posição absoluta das cidades e os que preservam a ordem relativa.

Dentre os operadores que mantém a posição absoluta estão: o PMX (*Partially Mapped Crossover)* proposto por Goldberg e Lingle (1985 apud POTVIN, 1996) e o CX (*Cycle Crossover)*, proposto por Oliver (1987 apud POTVIN, 1996). E entre os operadores que mantém a ordem relativa está o OX (*Order Crossover)* desenvolvido por Davis (1985 apud POTVIN, 1996).

Além destes operadores, Silva e Oliveira (2006), fizeram um estudo comparativo utilizando uma versão adaptada do HX (*Heuristic Crossover)*, o qual se destacou sobre os demais operadores, sobretudo quando o número de nós aumenta.

- **Operador HX:** O operador HX utiliza a distancias entre as cidades, isto é, o tamanho dos arcos. O mesmo pode ser descrito da seguinte forma:

Algoritmo:
P1: Escolha uma cidade aleatória inicial ***i*** de um dos pais.

P2: Calcular as distâncias entre a cidade aleatória ***i*** escolhida e suas vizinhas (*i -1 e* i + 1) em ambos os pais.
P3: No cromossomo filho, escolher para suceder a cidade i, a vizinha da mesma cujadistância até i seja a menor possível, a qual formará uma subrota.
P4: Se a inserção de qualquer das cidades vizinhas de i fecharem um ciclo, antes que todas as cidades sejam incluídas, sortear outra cidade (dentre aquelas que não pertencem a atual subrota). Esta cidade sorteada sucederá i na subrota.
P5: Repita os passos 2, 3 e 4 até que todas as cidades sejam incluídas na rota.

Descrição do Algoritmo Proposto

Considerando as definições anteriores, e usando os princípios da evolução das espécies, pode-se construir um algoritmo genético, cujos principais passos são os seguintes (Mayerle, 1994; Bezerra, 1995):

P1: *Construção da população inicial*: gere os m cromossomos aleatórios representando os roteiros Hamiltonianos. Calcule os custos $C_i = [Fitness(r_i)]$, $(1 \leq i \leq m)$, (1 ≤ i ≤ m) de todos os cromossomos gerados, construindo uma lista $R = (r_1, r_2, \ldots, r_m)$, de forma que $C_1 \leq C_2 \leq \ldots \leq C_m$; faça *k* = 0, defina o erro ε e o número máximo de iterações k_{max};
Obs. Os cromossomos são construídosde modo que a probabilidade de uma cidade suceder outra seja inversamente proporcional ao quadrado da distância que as separa;
P2: *Teste*: se $C_n - C_1 \leq \varepsilon$ ou $k \geq k_{max}$, então PARE e apresente o cromossomo r_1;
P3: *Seleção Natural*: selecione dois cromossomos $r_p = Select(R)$ e $r_q = Select(R)$ com $r_p \neq r_q$;
P4: *Reprodução*: faça $r_f = Crossover(r_p, r_q)$;
P5: Calcule $F_f = Fitness(r_f)$ e caso o fitness do cromossomo filho for melhor do que o fitness do pior cromossomo elimine esse cromossomo, inserindo o cromossomo r_f na lista R, mantendo a ordem crescente dos custos; faça k=k+1, e volte ao P2.

TESTES E RESULTADOS COMPUTACIONAIS

As Heurísticas de construção de rotas e Melhorias de Rotas para o PCV foram implementadas em Visual Basic 6. O Algoritmo Genético foi implementado em MATLAB 5.3, no qual foramutilizados 100 cromossomos e no máximo 2000 iterações.Os testes computacionais foram realizados em um notebook com processador Intel(R) Core (TM)2 Duo T5550, 1,83 GHz e 2,99 GB de RAM.

Para realizar a comparação das heurísticas explanadas na seção anterior, utilizou-secinco conjuntos de pontos de visitação, no qual cada um desses conjuntos representa um dia da semana de um representante da distribuidora de produtos. Os endereços destes pontos foram inseridos no *Google Earth* para obtenção de suas coordenadas geográficas. Após esse procedimento, realizou-se a transformação das coordenadas geográficaspara cartesianas, para que pudessem ser calculadas as distâncias euclidianas entre os pontos que o representante precisa visitar.

Na tabela 1 temos os dados das rotas atuais praticadas pelo representante e os resultados computacionais obtidos com as Heurísticas de Construção de Rotas, Melhorias de Rotas e com o Algoritmo Genético.

Tabela 1. Comparação dos resultados (custos em metros) das heurísticas e metaheuristicas.

Dia da semana	**Segunda** 25 pontos	**Terça** 26 pontos	**Quarta** 28 pontos	**Quinta** 20 pontos	**Sexta** 33 pontos
Rota atual	15808,84	25198,19	25185,87	36156,98	20691,49
Vizinho mais próximo	10694,08	16641,36	19173,03	23779,66	20137,86
Vizinho mais próximo com 2-opt	10593,35	16451,17	18940,7	23320,39	19793,07
Inserção do mais próximo	10483,08	17199,55	19362,35	23506,59	19294,12
Inserção do mais próximo com 2-opt	9910,512	16986,96	19274,71	23506,59	18542,89
Inserção do mais distante	10054,84	17331,3	18789,62	21309,23	18344,68
Inserção do mais distante com 2-opt	9879,759	17322,14	18598,65	21309,23	18301,71
Inserção do mais rápido	11438,71	17532,03	20656,51	23733,36	19857,29
Inserção do mais rápido com 2-opt	10979,42	17367,07	19478,81	23501,2	18512,03
Algoritmo Genético	9842,52	16451,17	18728,42	21309,23	18314,63

CONCLUSÕES

Neste trabalho foram realizadas testes com as Heurísticas de Construção e Melhoria de Rotaimplementadas. Em todos os dias da semana houve melhoria na rota já praticada pelo representante.

Em relação à comparação de desempenho entre as Heurísticas, a Inserção do Mais Distante com melhoria 2-opt foi a que apresentou melhores resultados juntamente com o Algoritmo Genético.
Em todas as heurísticas o tempo computacional foi de aproximadamente 7 segundos.

O uso dos Algoritmos Genéticos na busca de uma solução para o PCV permite encontrar rotas que, mesmo não tendo garantia de oferecer a melhor solução, são de boa qualidade. Considerando os resultados obtidos e levando em conta a dificuldade de se resolver o PCV para um número elevado de pontos, percebe-se a grande vantagem do uso dos Algoritmos Genéticos na busca de uma solução de boa qualidade para o problema.

REFERÊNCIAS

Barboza, A.O. Simulação e Técnicas da Computação Evolucionária Aplicadas a Problemas de Programação Linear Inteira Mista. Tese de Doutorado. Curitiba, BR: UTFPR. 2005

Bezerra, O.B. Localização de postos de coleta para apoio ao escoamento de produtos extrativistas - Um estudo de caso aplicado ao babaçu. Dissertação de Mestrado. Florianópolis, BR: UFSC. 1995

Bodin, L.D et al. “Routing and Scheduling of vehicles and crews - the state of the art”. Comput. Operational Research. 1983, vol 10, p. 63-211.

Costa, D.M.B. Aplicação de Algumas Técnicas da Pesquisa Operacional na Otimização dos Serviços Postais. Dissertação de Mestrado. Curitiba, BR: UFPR.1997

Ganhoto, M.A. Abordagens para problemas de roteamento. Dissertação de Mestrado, Universidade Estadual de Campinas, BR. 2004.

Goldberg, D.E. Genetic Algorithms in Search, Optimization e Machine Learning. Addison-Wesley Publishing Company, Inc. 1989

Goldbarg, M. C.; Luna, H. P. L. Otimização Combinatória e Programação Linear – Modelos e Algoritmos. Editora Campus. Rio de Janeiro. 2005

Grefenstette, J.J. "Optimization of control parameters for genetic algorithms". IEEE Transactions on Systems, Man and Cybernetics. 1986. vol 16, num.1, p.122-128.

Holland, J.H. "Genetic algorithms". Scientific American. 1992 vol 267, num.1, p.44-50.

Mayerle, S.F. Um Algoritmo Genético para o Problema do Caixeiro Viajante. Artigo de Circulação Interna do Departamento de Engenharia de Produção e Sistemas da UFSC. 1994

Michalewicz, Z. Genetic Algorithms + Data Structures = Evolution Programs. 3rd. editions. Berlin, Germany. Springer Verlag.1996

Nunes, L. F. Algoritmos Genéticos aplicados na abordagem de um problema real de roteamento de veículos. Dissertação de Mestrado. Curitiba, BR: UFPR. 1998.

Oliveira, H.C.B. et al. Adaptação do Algoritmo Genético no Tratamento do Problema de Roteamento de Veículos com Janela de Tempo. INFOCOMP (UFLA), vol 3, num 2, 2004. p. 51-58.

Potvin, J. Y. "Genetic algorithms for the traveling salesman problem". Annals of Operations Research. 1996. vol 6, p.339- 370.

Rocha, M.L; Ochi, L.S. Uma meta-heurística baseada em algoritmos genéticos não convencionais para o problema de roteamento de veículos multi-depósitos. Ed. Lorena Pradenas. Resumes Extendidos de Primeira ELIO - Optima 97 - Conception, Chile, 1997.

Schopf, E.C et al. Metaheuristicas Aplicadas na Solução do Problema do Caixeiro Viajante com Demandas Heterogêneas. XXIV ENEGEP – Florianópolis. 2004

Silva, A. F; Oliveira, A.C. Algoritmos genéticos: alguns experimentos com os operadores de cruzamento ("Crossover") para o problema do caixeiro viajante assimétrico. XXVI ENEGEP – Fortaleza. 2006

Teles, M.L; Gomes, H.M. Comparação de algoritmos genéticos e programação quadrática seqüencial para otimização de problemas de engenharia. Revista Teoria e Pratica na Engenharia Civil, 2010, vol10, num 15, 29–39.

Zamboni, L.V.S. Técnicas de Roteirização de Veículos Aplicadas ao Transporte Escolar. Dissertação de Mestrado. Curitiba, BR:UFPR. 1997

Whitley, D; Starkwether, T; Fuquay, D. Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator; Proceedings of the Third International Conference on Genetic Algorithms, George Mason University. 1989