Análise Comparativa de Algoritmos Heurísticos para Resolução do Problema do Caixeiro-Viajante em Grafos Não Clusterizados

Daniela Brandão Nascimento (UNISUL) danibra@unisul.br
João Neiva de Figueiredo (UFSC) jneiva@deps.ufsc.br
Rafael Machado Casali (UFSC) rmcasali@mtm.ufsc.br
Sérgio Fernando Mayerle (UFSC) mayerle@deps.ufsc.br

Resumo

O problema do caixeiro viajante é um dos problemas mais estudados da área de Pesquisa Operacional. De uma maneira sucinta, tal problema consiste na determinação de um circuito de mínimo custo que passe por todos os vértices de um grafo exatamente uma vez.

Desde a idealização deste problema ainda não se concebeu um algoritmo de complexidade polinomial que propiciasse uma solução exata. Entretanto, foram desenvolvidos diversos procedimentos heurísticos que buscam encontrar soluções razoavelmente próximas da solução exata. Este artigo apresenta uma análise comparativa de algumas destas heurísticas, testadas para diferentes tamanhos de grafos, para os quais procurou-se identificar o método mais promissor.

Palavras-chave: Problema do Caixeiro Viajante, Roteamento de Veículos.

1. Introdução

Segundo Campello e Maculan (1994), o problema do caixeiro viajante (PCV) é um dos mais tradicionais e conhecidos problemas de programação matemática, sendo que a primeira menção ao mesmo é devida a Hassler Whitney em 1934 em um trabalho na Princeton University.

O PCV tem sido amplamente estudado devido a pelo menos três de suas características: possui grande aplicação prática, grande dificuldade de obtenção da solução exata, além de uma significativa relação com outros modelos (LAPORTE e MARTELLO, 1990).

Dentre as aplicações práticas mais conhecidas do PCV, destacam-se o sequenciamento das operações de máquinas em manufatura, otimização de perfurações de furos em placas de circuitos impressos e a maioria dos problemas de roteamento de veículos.

Através deste artigo, pretende-se avaliar a eficiência de diferentes métodos heurísticos de resolução do PCV, avaliando os resultados obtidos em cada um deles para três diferentes faixas de tamanho de grafos. Pretende-se com isso identificar o método que tende a alcançar os resultados mais próximos da solução exata, apresentando para cada uma das faixas de tamanho uma análise dos resultados obtidos, em termos de tempo de processamento e custo do circuito encontrado. Vale salientar que os grafos testados foram gerados aleatoriamente, de forma que seus respectivos vértices fossem distribuídos de maneira homogênea ao longo do espaço considerado.

O artigo está estruturado da seguinte forma:

a) caracterização do problema;
b) breve descrição dos métodos heurísticos avaliados;
c) caracterização das estruturas de armazenamento utilizadas na implementação dos

algoritmos testados;

d) apresentação dos resultados obtidos para cada faixa de tamanho de grafo;

e) conclusões finais.

2. O Problema do Caixeiro Viajante

De acordo com Goldbarg e Luna (2000), "os problemas de roteamento lidam em sua maior parte com passeios ou *tours* sobre pontos de demanda ou oferta". Tais pontos podem estar representando cidades, postos de trabalho, depósitos, etc, sendo que dentre os tipos de "passeios" um dos mais importantes é o denominado *circuito hamiltoniano*. Este nome foi dado em homenagem a Willian Rowan Hamilton que, em 1857, propôs um jogo que denominou *Around the World*. Este jogo consistia em encontrar uma rota através de vértices de um dodecaedro, que representava as 12 cidades mais importantes da época, de modo a iniciar e terminar o roteiro em um mesmo vértice, sem nunca repetir uma visita.

Na bibliografia disponível sobre o assunto pode-se encontrar diversas formulações para o PCV, bem como métodos exatos e não exatos (heurísticos) para sua resolução. O PCV foi formulado pela primeira vez em 1954 por Dantzig, Fulkerson e Johnson como um problema de programação 0-1 sobre um grafo $G = (N, A)$, como segue:

$$\textit{Minimizar} \quad z = \sum_{j=1}^{n} \sum_{i=1}^{n} c_{ij} x_{ij} \tag{1}$$

$$\textit{Sujeito a:} \quad \sum_{i=1}^{n} x_{ij} = 1 \qquad \forall j \in N \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i \in N \tag{3}$$

$$\sum_{(i,j) \in \langle S \rangle} x_{ij} \leq |S| - 1 \qquad \forall S \subset N \tag{4}$$

$$x_{ij} \in \{0,1\} \qquad \forall i, j \in N \tag{5}$$

onde:

c_{ij} é o custo de se transpor o arco $(i, j) \in A$;

$x_{ij} = 1$ se o arco $(i, j) \in A$ for escolhido para integrar a solução, e 0 em caso contrário;

$\langle S \rangle$ é um subgrafo de $G = (N, A)$, definido pelo subconjunto próprio de vértices $S \subset N$.

Na formulação acima, (1) determina a minimização da função de custo total da rota em formação; (2) e (3) asseguram que cada vértice possui exatamente um sucessor e um predecessor e a restrição (4) garante o descarte de subrotas que não formam um circuito hamiltoniano.

Tal formulação destaca a natureza combinatória do problema, onde fica claro que solucionar um PCV é determinar uma "permutação legal de mínimo custo dos vértices do grafo" (GOLDBARG e LUNA, 2000).

Dentre outras, as formulações propostas por Miller, Tucker e Zemlin (1960), Fox, Gavish e Graves (1980), também associam a resolução do PCV a problemas de programação 0-1, que propiciam a obtenção de soluções exatas. De acordo com Bodin et. al. (1983), o procedimento de relaxação Lagrangeana pode propiciar bons resultados para problemas que envolvem roteamento combinatorial. Christofides (1979) aborda em seu artigo o problema de atribuição e matching, bem como a relaxação do PCV para a obtenção do menor caminho hamiltoniano.

No entanto, dependendo do tamanho do grafo, a aplicação destas formulações matemáticas pode se tornar inviável. Em função disso vários outros métodos, de natureza heurística, e que não garantem a obtenção da solução ótima, têm sido propostos.

3. Descrição dos Algoritmos Heurísticos Avaliados

Segundo Bodin *et. al.* (1983), as heurísticas propostas podem ser classificadas como procedimentos de construção de rotas, procedimentos de melhoramento de rotas e procedimentos compostos (construção + melhoramento). Estes procedimentos requerem a determinação da matriz de custos dos caminhos mínimos entre todos os pares de vértices do grafo. Enquanto os procedimentos de construção seguem alguma regra que determina o vértice e a posição de sua inserção na rota em formação, os procedimentos de melhoramento, partem de uma rota inicial conhecida, e tentam reduzir o comprimento total do percurso por meio da realização de permutações entre alguns dos vértices da rota.

No presente artigo, apresentam-se os resultados obtidos para o PCV com o uso de alguns dos métodos heurísticos de construção e de melhoramento apresentados em Bodin *et. al.* (1983), e que se encontram descritos na seqüência.

3.1. Métodos de construção

Foram implementados os seguintes métodos de construção, apresentados originalmente em Rosenkrantz, Stearns e Lewis (1977):

a) <u>Vizinho Mais Próximo – VMP</u>

Passo 1: Comece o caminho em um vértice qualquer;

Passo 2: Encontre o vértice não pertencente ao caminho que esteja mais próximo do último vértice adicionado; adicione-o no final do caminho em formação;

Passo 3: Repita o passo 2 até que o caminho compreenda todos os vértices, e quando isto ocorrer una o último vértice ao primeiro, fechando a rota.

Este algoritmo requer da ordem de n^2 operações. Para melhorar o resultado, freqüentemente se aplica este algoritmo a todos os vértices e se escolhe a melhor solução.

b) <u>Inserção do Mais Próximo – IMP</u>

Passo 1: Comece com um subgrafo composto apenas pelo i-ésimo vértice;

Passo 2: Encontre um vértice $k \neq i$ tal que c_{ik} seja mínimo e forme a subrota $i - k - i$;

Passo 3: Dada uma subrota, encontre um vértice k não pertencente à mesma, mas que seja o mais próximo de todos vértices nela inserido;

Passo 4: Encontre o arco (i, j), nesta subrota, para o qual $c_{ik} + c_{kj} - c_{ij}$ é mínimo e insira k entre i e j;

Passo 5: Caso seja encontrado um ciclo hamiltoniano, encerre; caso contrário vá para o passo 3.

Este algoritmo também requer da ordem de n^2 operações.

c) Inserção do Mais Barato – IMB

Passo 1: Comece com um subgrafo composto apenas pelo i-ésimo vértice;

Passo 2: Encontre um vértice $k \neq i$ tal que c_{ik} seja mínimo e forme a subrota $i-k-i$;

Passo 3: Encontre o par (i, j) pertencente à subrota e k não pertencente à subrota tal que $c_{ik} + c_{kj} - c_{ij}$ seja mínimo e insira k entre i e j;

Passo 4: Caso seja encontrado um ciclo hamiltoniano, encerre; caso contrário vá para o Passo 3.

Este algoritmo requer da ordem de $n^2 \log(n)$ operações.

d) Inserção do Mais Distante – IMD

Passo 1: Comece com um subgrafo composto apenas pelo i-ésimo vértice;

Passo 2: Encontre um vértice $k \neq i$ tal que c_{ik} seja máximo e forme a subrota $i-k-i$;

Passo 3: Dada uma subrota, encontre um vértice k que não pertence a esta subrota, mas que seja o mais distante de todos os vértices pertencentes à mesma;

Passo 4: Encontre o arco (i, j) nesta subrota para o qual $c_{ik} + c_{kj} - c_{ij}$ é mínimo e insira k entre i e j;

Passo 5: Caso seja encontrado um ciclo hamiltoniano, encerre; caso contrário vá para o Passo 3.

Este algoritmo requer da ordem de n^2 operações.

e) Inserção Rápida – IR

Passo 1: Escolha qualquer vértice para compor um circuito inicial T_1;

Passo 2: Dado o circuito T_k com k vértices, encontre o vértice z_k que não faça parte de T_k, mas que seja o mais próximo a um vértice arbitrário y_k em T_k;

Passo 3: Forme T_{k+1} inserindo z_k imediatamente adiante de y_k em T_k;

Passo 4: Repita os passos 2 e 3 até que se forme um circuito hamiltoniano.

Este algoritmo requer da ordem de n^2 operações.

f) União de Rotas – UR

Passo 1: Faça S_1 consistir de n subrotas, cada uma delas contendo um único vértice;

Passo 2: Para cada $i < n$, encontre um arco (a_i, b_i) tal que $C_{a_i b_i} = \min(C_{xy})$ para x e y em diferentes subrotas em S_i. Então, S_{i+1} é obtido a partir de S_i através da união de subrotas contendo a_i e b_i;

Passo 3: Repita o passo 2 até que todas as subrotas tenham sido unidas.

Este algoritmo requer da ordem de n^2 operações.

3.2. Método de melhoramento

Foi implementado o método de melhoramento conhecido por 2-opt, o qual consiste nos seguintes passos:

Passo 1: Encontre uma rota inicial;

Passo 2: Tente melhorar a rota utilizando o procedimento 2–opt: a partir de dois arcos (a,b) e (c,d) não adjacentes da rota, verifique a possibilidade de substituição destes arcos pelos arcos (a,c) e (d,b). Caso esta substituição promova uma redução de custo da rota, efetue a troca. Senão, mantenha os arcos originais;

Passo 3: Execute o passo 2 para todas as combinações possíveis de dois arcos não adjacentes do grafo, até que não seja mais possível obter reduções de custo.

4. Implementação

Trabalhou-se somente com grafos não-orientados e completos. A armazenagem das informações deu-se através de ponteiros e listas encadeadas, conforme descrição a seguir:

– utilizou-se uma estrutura de dados para armazenar as informações referentes a cada um dos vértices. Esta estrutura contempla as seguintes informações: coordenadas, identificador do vértice, vértices predecessor e sucessor em cada uma das rotas, etc.;

– definiu-se como "custo" da conexão entre pares de vértices a respectiva distância euclidiana;

– calculados os custos entre os vértices, utilizou-se um procedimento de ordenação pelo método QuickSort para listá-los em ordem crescente. Tais informações foram armazenadas através de uma estrutura em forma de lista, com objetivo de facilitar a implementação das várias heurísticas avaliadas;

– utilizou-se uma estrutura de dados em forma de lista para armazenar as rotas geradas a partir de cada uma das heurísticas implementadas.

Os grafos testados foram gerados aleatoriamente, a partir da especificação do número de vértices por parte do usuário, e posteriormente gravados em arquivos-texto. O programa dispõe também de um procedimento para leitura de arquivos, o que permite a reprodução das análises apresentadas neste trabalho.

Para cada heurística implementada, o programa disponibiliza ao usuário a distância referente ao caminho gerado, bem como o tempo de processamento.

É importante salientar que a escolha do "vértice inicial" (primeiro vértice a ser inserido no caminho) pode interferir significativamente nos resultados obtidos. Desta forma, definiu-se como vértice inicial àquele referente ao par associado à menor distância verificada entre todos os pares de vértices, isto é, tomou-se sempre o primeiro elemento da "lista de custos em ordem crescente", mencionada anteriormente.

5. Grafos Testados e Resultados Obtidos

Com o objetivo de testar a eficiência dos algoritmos apresentados, considerou-se três diferentes tamanhos de grafos, conforme o número de vértices:

a) grafos pequenos (de 30 a 50 vértices);

b) grafos médios (de 250 a 500 vértices);

c) grafos grandes (de 1000 a 1200 vértices).

Para cada uma destas classes de tamanho foram testados 50 grafos gerados aleatoriamente, registrando, para cada grafo e algoritmo, o comprimento da rota obtida e o tempo de processamento. O quadro apresentado a seguir sintetiza os resultados obtidos com a utilização de um microcomputador Pentium III 733 MHz com 256 MB de memória RAM, para cada uma das situações analisadas. No Quadro são apresentados os resultados obtidos com a aplicação exclusiva dos métodos de construção de rotas e com o método de melhoramento 2-opt.

		Heurística mais Eficiente Sem 2-opt		Heurística mais Eficiente Com 2-opt	
Tamanho do Grafo	**Critério de Eficiência**	**Tipo**	**Freqüência de obtenção da maior eficiência (em %)**	**Tipo**	**Freqüência de obtenção da maior eficiência (em %)**
Pequeno	Menor Distância	IMD	94	IMD	30
Pequeno	Tempo Processamento	Todas	-	Todas	-
Médio	Menor Distância	IMB	58	VMP	98
Médio	Tempo Processamento	IR	90	VMP	50
Grande	Menor Distância	IMB	94	VMP / UR	68
Grande	Tempo Processamento	IR	100	VMP	100

Quadro 1 – Resumo dos resultados obtidos com a implementação das heurísticas de construção de rotas.

Através dos resultados obtidos a partir dos grafos testados, verifica-se uma tendência da heurística mais eficiente estar associada ao número de vértices.

Como se pode constatar através dos dados apresentados no Quadro 1, respectivamente, a heurística IMD propiciou os melhores resultados para os grafos classificados como pequenos. Com relação aos grafos médios e grandes, a heurística IMB mostrou ser a mais eficiente no quesito “menor distância”, sendo que tal tendência foi modificada a partir da utilização da heurística de melhoria 2-opt. A utilização desta heurística implicou no aumento de eficiência da heurística VMP, que passou a propiciar caminhos com as menores distâncias para grafos médios e grandes.

Verifica-se também que a utilização da heurística de melhoria 2-opt modificou de maneira significativa a tendência de maior eficiência das heurísticas de construção, quando implementadas de maneira isolada, exceto no caso dos grafos pequenos, para os quais a heurística IMD tende a ser a mais eficiente em qualquer situação.

Há que se observar que, para o caso dos grafos “grandes”, as heurísticas IMD e IMB convergiram num tempo de processamento relativamente grande, quando comparadas às demais. Com relação a esta última (IMB), o tempo de processamento ultrapassou 1 minuto para todos os grafos classificados como grandes. Tal fato pode ser motivado pelo tipo de estrutura de dados utilizada na programação, que para o caso específico da heurística IMB pode não ser o ideal. No entanto, e mesmo que se opte futuramente por re-implementar esta heurística com base em uma outra estrutura de dados, entende-se que a mesma é “mais cara” sob o ponto de vista computacional.

O quadro a seguir apresenta a diferença percentual média entre custo da rota obtida com cada heurística e a melhor heurística aplicada ao grafo, para cada classe de grafo testada, com e sem a aplicação do melhoramento 2-opt. Apenas para efeito comparativo, apresenta-se, também, o ganho médio obtido com a aplicação da heurística de *Guided Local Search* (GLS), proposta por Voudouris (1997), sobre o melhor resultado obtido entre todas as heurísticas aplicadas. Note-se que a heurística proposta por Voudouris obteve o melhor resultado para todos os grafos testados, porém com um tempo de processamento significativamente maior ao gasto pelas demais heurísticas estudadas.

	Diferenças Percentuais Médias (%)					
	Grafos Pequenos		Grafos Médios		Grafos Grandes	
Heurística	**Sem 2-opt**	**Com 2-opt**	**Sem 2-opt**	**Com 2-opt**	**Sem 2-opt**	**Com 2-opt**
Vizinho mais Próximo	16,17	4,16	5,31	0	5,04	0
Inserção do mais Próximo	11,58	7,38	4,13	5,54	4,22	6,74
Inserção do mais Distante	13,47	3,35	2,12	3,46	4,72	5,69
Inserção do mais Barato	8,21	6,60	1,60	4,28	0,02	5,09
União de Rotas	11,18	6,44	3,64	4,00	3,64	4,57
Inserção Rápida	26,96	4,92	19,69	2,89	19,28	3,28
$\frac{\text{Melhor Heurística - Rota GLS}}{\text{Rota GLS}} \times 100$	6,07		8,20		8,33	

Quadro 2 – Diferença percentual média do custo da rota, para cada heurística e classe de grafo testada, com e sem a aplicação da heurística de melhoramento 2-opt e ganhos promovidos pela heurística de *Guided Local Search*.

6. Conclusões Finais

O estudo ora apresentado teve como objetivo subsidiar decisões associadas à implementação de diferentes heurísticas concebidas para resolução do Problema do Caixeiro Viajante. Acredita-se que o mesmo possa fornecer informações úteis no que diz respeito à eficiência de tais heurísticas de construção e melhoramento de rotas, em função do tamanho do grafo. É importante salientar que os estudos restringiram-se à análise de grafos gerados aleatoriamente, com seus respectivos vértices dispersos de maneira homogênea, o que pode ocorrer em algumas das aplicações do PCV.

No entanto, e principalmente quando se trata de problemas de roteamento de veículos, muitas vezes os pontos de entrega e coleta encontram-se agrupados em clusters. Para esta situação, nada se pode concluir com base nos estudos realizados a respeito da eficiência das heurísticas avaliadas.

Com relação à possibilidade de futuras pesquisas associadas ao tema, considera-se interessante avaliar a influência da escolha do "nó inicial" na eficiência das heurísticas estudadas. Tal pesquisa poderá eventualmente indicar que a escolha deste parâmetro afeta o desempenho de uma determinada heurística, conduzindo a possíveis alterações nos resultados apresentados neste trabalho.

Referências

BODIN, L.; GOLDEN, B.; ASSAD, A.; BALL, M. 1983. Routing and Scheduling of Vehicles and Crews: The State of the Art. An International Journal Computers & Operations Research, Vol. 10, No. 2, p. 79 – 92.

CAMPELLO, C. R., MACULAN, N. 1994. Algoritmos e Heurísticas. Editora da Universidade Federal Fluminense, Niterói, RJ.

CHRISTOFIDES, N. 1979. The Traveling Salesman Problem. Combinatorial Optimization. Wiley Chichester, p. 131 – 149.

DANTZIG, G. B.; FULKERSON, D. R.; JOHNSON, S. M. 1954. Solution of a Large Scale Traveling Salesman

Problem. Opns. Res 2, p. 393 – 410.

FOX, K., GAVISH, B., GRAVES, S. 1980. An n-constraint formulation of the (time-dependent) traveling salesman problem. Operation Research 28: 1018 – 1021.

GOLDBARG, M. C.; LUNA, H. P. L. 2000. Otimização Combinatória e Programação Linear: Modelos e Algoritmos. Ed. Campus. p. 397 – 410.

LAPORTE, G.; MARTELLO, S. 1990. The Selective Traveling Salesman Problem. Discrete Appl. Math. 26. p. 193 – 207.

MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. 1960. Integer Programming Formulations and Traveling Salesman Problems, Journal of the Association for Computing Machinery, 7. 326 – 329.

ROSENKRANTZ, R.; STERNS, R; LEWIS, P. 1977. An Analysis of Several Heuristics for the Traveling Salesman Problem. SIAM j. Com. 6. p. 563 – 581.

VOUDOURIS, Christos; *Guided Local Search for Combinatorial Optimization Problems*; PhD. Thesis; Dept. Computer Science; University of Essex; 1997.