

Simulação de Monte Carlo e Métodos Heurísticos Paralelizados para a Resolução do Problema do Roteamento de Veículos Capacitados

Luis Gustavo Pereira - lgp1985@yahoo.com.br
(Universidade Tecnológica Federal do Paraná - UTFPR)

Wesley José Nogueira Medeiros - wesley_nogueira@yahoo.com.br
(Universidade Tecnológica Federal do Paraná - UTFPR)

Thiago André Guimarães - thiandgui@gmail.com
(Centro Universitário Franciscano do Paraná - UNIFAE)

Resumo:

O Problema de Roteamento de Veículos Capacitados (PRVC) busca determinar um conjunto de rotas para uma frota homogênea de veículos, partindo de um depósito central com destino a um conjunto de clientes que demandam determinado produto. Por ser um dos mais importantes e complexos problemas afetos à logística de distribuição, alternativas heurísticas para o problema são constantemente apresentadas. Nesse sentido, este artigo endereça a uma abordagem heurística baseada na estratégia “*Cluster First Route Second*” que consiste primeiramente em agrupar os clientes conforme a demanda para posteriormente construir rotas para os grupos formados. A abordagem emprega Simulação de Monte Carlo em conjunto com o clássico algoritmo de Gillett e Johnson (1976) para a resolução do problema do agrupamento, enquanto que as rotas para cada grupo são construídas pela heurística do vizinho mais próximo com refinamentos advindos de trocas *2-opt*. Empregaram-se ainda técnicas de processamento paralelo para redução do tempo de processamento computacional. Testes realizados com as instâncias clássicas disponíveis na literatura apontaram um bom desempenho em relação ao resultado ótimo dos problemas, com tempo de processamento computacional praticamente nulo para a resolução de cada instância.

Palavras chave: Roteamento de veículos capacitados; Procedimentos heurísticos; Agrupamento de pontos de demanda, Simulação de Monte Carlo.

Monte Carlo Simulation and Parallel Heuristic Methods to solve the Capacited Vehicle Routing Problem

Abstract

The Capacited Vehicle Routing Problem (CVRP) seeks to determine a set of routes for a homogeneous fleet of vehicles from a central warehouse bound to a set of customers who demands products. As one of the most important and complex problems related to the logistics of distribution, heuristics to the problem are constantly developed. On this way, this paper addresses the heuristic approach based on strategy “*Cluster First Route Second*” that consists of grouping customers according to demand to further build routes for the groups formed. The approach employs Monte Carlo simulation methods and Gillett and Johnson (1976) classical algorithm for solving the clustering problem, while the routes for each group are constructed by the nearest neighbor heuristic and refinements arising from exchanges by *2-opt*. It is also used parallel computing to reduce the computational time. Simulations performed with the classic instances available in the shown literature results on a good performance. The results took an advantage of consuming a computational processing time shorter than 1 second.

Key-words: Capacited Vehicle Routing Problem, Heuristics, Parallel Computing

1. Introdução

Uma das mais importantes abordagens para a melhoria da distribuição de produtos em redes logísticas é a do Problema do Roteamento de Veículos Capacitados (PRVC). Desde que foi proposto por Dantzig e Ramser (1959), o PRVC vem sendo estudado e solucionado sobre a ótica de inúmeras e diferentes abordagens. Abordagens estas que contemplam desde os métodos exatos de programação linear aos mais recentes empregos de heurísticas.

O objetivo deste trabalho é propor uma abordagem heurística diferente para a resolução do Problema de Roteamento de Veículos Capacitados (PRVC). Esta abordagem baseia-se na estratégia “*Cluster First Second Route*”, com a utilização do algoritmo do vizinho mais próximo para a roteirização dos veículos e o emprego de simulação de Monte Carlo juntamente com processamento paralelo para a redução do tempo computacional. Ao final do estudo, pretende-se analisar os resultados através de testes com instâncias da literatura.

Para tanto, o estudo se inicia com um referencial teórico, apresentado no item 2, onde o PRVC é formalmente definido. No item 3 discute-se os principais algoritmos necessários à construção das estratégias propostas. Neste mesmo tópico a abordagem completa para a resolução do problema é apresentada. No item 4, discute-se os resultados obtidos enquanto o item 5 tece as conclusões do estudo apontando sugestões para trabalhos futuros.

2. Referencial Teórico

Esta seção introduz formalmente o PRVC e apresenta a estratégia de resolução do mesmo em dois estágios: agrupamento e roteirização. Comentam-se também a técnica de processamento paralelo empregada para a minimização dos tempos computacionais.

2.1 Problema de Roteamento de Veículos Capacitados (PRVC)

O PRVC pode ser formalmente definido sob o seguinte aspecto: seja $G(V,E)$ um grafo não direcionado contendo o conjunto de vértices $V = \{0, 1, \dots, n\}$, onde o vértice “0” representa o depósito, enquanto que todos os outros vértices representam os clientes, sendo que cada cliente i possui uma demanda d_i . Cada arco $e \in E$ possui um comprimento não negativo $\ell(e)$. Dado G e dois números positivos e inteiros (K e C), o PRVC consiste em encontrar um conjunto de rotas para os K veículos, que atenda as restrições:

- (i) Cada rota inicia e termina no depósito;
- (ii) Cada cliente é visitado por um único veículo;
- (iii) A demanda total de todos os clientes de uma sub-rotas não excede a capacidade C do veículo.

Uma instância exemplo é apresentada na figura 1 a seguir:

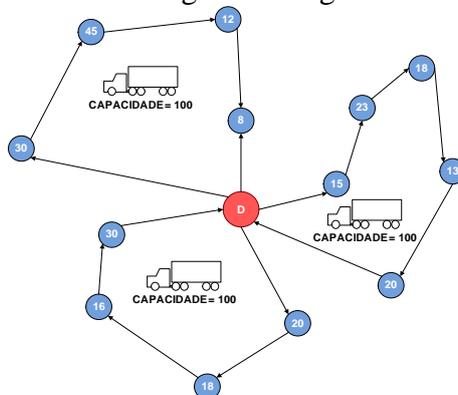


Figura 1 - Instância exemplo do PRVC Fonte: Elaborado pelo autor

O objetivo do PRVC é minimizar o somatório das distâncias de todas as sub-rotas. O problema é fortemente *NP-hard*, dado que é uma generalização do Problema do Caixeiro Viajante (PCV) demandando, portanto, elevado esforço computacional para obtenção da solução ótima através de busca exaustiva. Baseado em Lin *et al.* (2009), o PRVC pode ser modelado como um problema de programação inteira misto como segue:

$$\text{Minimizar} \quad \sum_{i=0}^N \sum_{j=0}^N \sum_{k=1}^K C_{ij} X_{ijk} \quad (1)$$

$$\text{Sujeito à:} \quad \sum_{i=0}^N \sum_{j=0}^N X_{ijk} d_i \leq Q^k \quad 1 \leq k \leq K, \quad (2)$$

$$\sum_{j=1}^N X_{ijk} = \sum_{j=1}^N X_{jik} \leq 1, \quad \text{para } i=0 \text{ e } k \in \{1, \dots, K\}, \quad (3)$$

$$\sum_{k=1}^K \sum_{j=1}^N X_{ijk} \leq K, \quad \text{para } i=0, \quad (4)$$

onde C_{ij} é o custo para se ir do cliente i para o cliente j ; K é a quantidade de veículos disponível; N é o número de clientes; Q^k é a capacidade de carregamento do veículo k e finalmente d_i é a demanda do cliente i . As variáveis $X_{ij}^k \in \{0, 1\}$ ($i \neq j$; $i, j \in \{0, 1, \dots, N\}$).

A equação (1) é a função objetivo do problema que, conforme já comentado, minimiza o somatório das distâncias de todas as sub-rotas. As restrições (2) referem-se à capacidade de carregamento do veículo e atendimento das demandas de cada cliente, onde $X_{ijk}=1$, se o veículo k viaja do cliente i para o cliente j diretamente e $X_{ijk} = 0$, caso contrário.

As restrições (3) garantem que toda rota tem início e fim no depósito, enquanto que a restrição (4) especifica que há no máximo K rotas partindo do depósito.

2.2 Estratégia Cluster First Route Second

A estratégia *Cluster First Route Second* para a resolução do PRVC baseia-se na aplicação de duas fases elementares de cálculo para a obtenção da solução final. Essa estratégia é comumente reportada na literatura e consiste, primeiramente em agrupar os clientes conforme a localização e quantidade demandada para posteriormente construir rotas para os grupos formados. Garantindo que a demanda dos clientes pertencentes a um grupo não exceda a capacidade do veículo, o PRVC recai na resolução do Problema do Caixeiro Viajante (PCV) ao ensejar um roteiro para cada grupo que passe por todos os pontos e apresente, ao final, a menor distância possível.

Fisher e Jaikumar (1981) e Gillett e Johnson (1976) são exemplos de autores que apontam heurísticas do tipo *Cluster First Route Second* para a resolução do PRVC. Variações do problema clássico tratadas com essa abordagem encontram-se em Sariklis and Powell (2000), que trabalham para a resolução do *OVRP* (*Open Vehicle Routing Problem*). O *OVRP* se diferencia do PRVC por não necessitar do retorno do veículo ao depósito ao término da rota.

A estratégia “*Cluster First Route Second*” também é endereçada por Steiner *et al.* (2000) para a resolução de problema de roteamento em transporte escolar. Os autores sugerem o emprego da estratégia tanto para o roteamento capacitado quanto para o roteamento não capacitado de veículos.

2.3 Processamento Paralelo

Leijen e Hall (2007) indicam que os computadores de vários núcleos tornaram-se o padrão industrial aplicado atualmente. Infelizmente, segundo os mesmo autores, a maioria dos aplicativos utiliza um só núcleo e não apresenta aumento de velocidade quando executada em uma máquina com vários núcleos. Dessa forma, para aprimorar o desempenho de um *software* torna-se necessário executá-lo em vários processadores em paralelo.

Novas ferramentas de programação foram desenvolvidas para empregar os recursos de paralelismo em computadores multinucleados. Dentre elas, Leijen e Hall (2007) destaca a classe *Parallel* presente no *.NET Framework 4* e exemplificada na Figura 2 abaixo:

```
System.Threading.Tasks.Parallel.For(0, N, i =>
{
    // Processa i aqui
});
```

Figura 2 - Exemplo de trecho paralelizado Fonte: Leijen e Hall (2007)

O presente estudo emprega a linguagem de programação C# 4.0 em conjunto com a extensão de linguagem denominada *Language Integrated Query* (LINQ), além da classe *Parallel*. O LINQ possibilita uma manipulação de conjuntos de dados e o a classe *Parallel* permite o devido processamento paralelo pelo *software*.

Embora represente um grande passo à frente, Toub (2011) reforça que a classe *Parallel* só chega a abordar superficialmente as funcionalidades disponíveis para paralelismo. Conforme o mesmo autor, um dos mais formidáveis avanços em paralelização foi o realizado pelo *.NET Framework 4* através da introdução da extensão *Parallel LINQ* (PLINQ). Essa extensão permite o processamento paralelo dos dados internos de um conjunto, desde que os mesmo possuam alto grau de independência linear.

A Figura 3 abaixo exemplifica a utilização da extensão PLINQ – *AsParallel()* – dentro do *software* criado neste trabalho:

```
var medianaProxima = (from m in mMedianas.AsParallel()
                      orderby m % designando.cliente.Coordenada
                      where m.CapacidadeDisponivel >=
designando.cliente.Demanda
                      select m).First();
```

Figura 3 - Exemplo de aplicação de PLINQ no *software* criado Fonte: Elaborado pelo autor

Visando reduzir o tempo de processamento e aumentar a capacidade de análise do *software*, a heurística proposta nesse trabalho bem como os algoritmos utilizados na sua implementação foram programados utilizando os conceitos de processamento paralelo supracitados.

3. Heurística proposta

Problemas de grande porte como o PRVC desmotivam o emprego de modelos exatos de programação linear dado que o tempo de processamento aumenta exponencialmente com o incremento do número de clientes (nós). Devido a isto, métodos heurísticos vêm sendo propostos para a resolução do PRVC no intuito de se produzir bons resultados com tempo de processamento factíveis e adequados as necessidade empresariais.

A construção da heurística proposta neste artigo requer a resolução dos problemas da p-medianas, da designação ou agrupamento (clusterização), da roteirização de veículos e da melhoria de rotas. Assim, as subseções a seguir discutem esses problemas de forma pormenorizada para que a nova abordagem heurística seja devidamente apresentada ao final.

3.1. O problema das p -medianas

O problema das p -medianas tem por objetivo determinar, em um grafo formado por um conjunto de pontos (clientes), p -pontos principais que representam sementes ou depósitos fictícios. Os demais pontos (clientes) serão designados a cada uma destas medianas de modo a formar p -conjuntos iniciais (p -grupos) de clientes onde cada conjunto contenha a “capacidade limite” igual à capacidade de carregamento dos veículos. Ressalta-se que neste trabalho a capacidade do grupo é a mesma para todos eles (frota homogênea de veículos).

A estratégia de resolução reside na teoria dos grafos que tem por objetivo localizar facilidades ao longo de uma rede viária. Estas facilidades (p -medianas) devem ser escolhidas de forma a minimizar a soma da distância de cada um dos pontos à facilidade mais próxima, ponderada por um fator de demanda.

O problema pode ser resolvido de forma exata, como sugeriu Hakimi (1965), usando enumeração exaustiva ou programação inteira. Porém, segundo Juan *et al* (2010), recentes avanços no desenvolvimento de geradores números pseudoaleatórios vem abrindo novas perspectivas para o uso de simulação de Monte Carlo em problemas de otimização.

Takes (2010) também opina no mesmo sentido, enfatizando a aplicabilidade do método estocástico de Monte Carlo para a resolução das variantes do *Vehicle Routing Problem* (PRV) em virtude do PRVC ser um problema *np-hard* de otimização combinatória.

Nessa visão, o presente trabalho optou por empregar o método estocástico de Monte Carlo para resolução do problema das p -medianas, no claro intuito de gerar um grupo de pontos a serem empregados como pontos iniciais (depósitos fictícios) para o agrupamento dos clientes.

Para a determinação destes p -pontos iniciais, foi utilizado o gerador de números pseudoaleatórios existente no *.NET Framework 4*, devidamente orientado para determinar esses p -pontos iniciais dentro do grafo inicial de clientes existentes nas instâncias do PRVC. Essa ferramenta computacional garante que a escolha dos pontos iniciais será feita de maneira estocástica e baseada em uma séria estatística de distribuição normal.

A orientação citada anteriormente baseia-se na delimitação de uma área para a atuação da simulação de Monte Carlo. Essa área foi criada através de um retângulo definido pelas mínimas e máximas coordenadas dos pontos do grafo, em seus respectivos eixos de abscissas e ordenadas. Tal delimitação visa otimizar o processo de geração de pontos pelo método de Monte Carlo, uma vez que elimina a possibilidade de serem escolhidos pontos demasiadamente distantes e externos ao grafo do PRVC.

Cabe ressaltar que essa etapa contribui demasiadamente para o funcionamento e a precisão da heurística proposta neste estudo. Principalmente porque a determinação dos p -pontos iniciais será pré-requisito para que o algoritmo de Gillett e Johnson (1976) consiga designar os demais pontos e formar os grupos (clusters) de clientes.

3.2. O problema da designação de grupos (clusterização)

Definido o conjunto das medianas é necessário designar os pontos de maneira a se compor o agrupamento para roteirização posterior. Esta designação objetiva minimizar a distância entre os pontos do conjunto e a respectiva mediana, respeitando a capacidade da própria mediana.

Neste trabalho, a designação dos pontos é feita através do algoritmo de Gillett e Johnson (1976). O princípio do algoritmo é construir uma lista ordenada dos pontos a serem designados considerando distâncias entre esses pontos e as duas medianas mais próximas. O algoritmo é descrito pela figura 4 a seguir:

Seja: t_i^1 e t_i^2 : Primeira e segunda mediana mais próxima ao ponto i ;
 c_i^1 e c_i^2 : Distância à primeira e segunda mediana mais próxima ao ponto i ;

Passo 1. Calcular a distância entre cada nó ainda não designado até cada um dos depósitos “medianas” correspondentes que ainda possuem capacidade;

Passo 2. Para cada ponto i , obter t_i^1 e t_i^2 e as respectivas distâncias c_i^1 e c_i^2 .

Passo 3. Para cada ponto i , calcular a razão $r_i = c_i^1 / c_i^2$.

Passo 4. Ordenar os pontos de acordo com r_i em ordem decrescente.

Passo 5. Designar todos os pontos i da lista ordenada para t_i^1 até sua capacidade ser atingida. Se houver algum ponto ainda não designado volte ao passo 1.

Figura 4 - Algoritmo de Gillett e Johnson para agrupamento. Fonte: Gillett e Johnson (1976)

3.3 O Problema da Roteirização: Geração de uma Solução Inicial

Inúmeros são os algoritmos existentes para a roteirização de veículos tendo um conjunto de clientes já definido. A resolução do problema do roteamento enseja a obtenção de uma solução inicial para posterior refinamento. Para a abordagem proposta neste trabalho foi utilizado o algoritmo do Vizinho Mais Próximo (VMP) pela sua ampla divulgação na literatura existente e fácil implementação computacional.

O algoritmo de roteirização VMP foi originalmente proposto por Cover e Hart (1967) e consiste composição da rota com base na inserção sequencial de pontos através de um ponto inicial conforme a menor distância entre este e os seus demais pontos (vizinhos). Após determinar todas as distâncias entre o ponto inicial e os demais pontos do cluster, os demais pontos são ordenados de maneira decrescente para possibilitar a escolha e determinação do nó mais próximo que será designado à rota. O algoritmo é exposto na figura 5 a seguir:

Passo 1. Iniciar a roteirização a partir do depósito e designar este ponto como ponto i visitado ($i \in S$), onde S é o grupo de clientes visitados;

Passo 2. Para cada ponto $j \notin S$, obter d_{ij} sendo este a respectiva distância entre o ponto i e j .

Passo 3. Escolher o menor d_{ij} e marcar o ponto j como visitado;

Passo 4. Faça o ponto j ser o ponto i ;

Passo 5. Se todos os pontos já foram visitados, encerrar o algoritmo;

Passo 6. Caso contrário, retorne ao Passo 2.

Figura 5 - Algoritmo VMP. Fonte: Cover e Hart (1967)

3.4. Problema da Roteirização: Melhoria da Solução Inicial

O problema da melhoria de rotas busca aumentar a eficiência do trajeto, ou seja, refinar uma solução inicial obtida para uma rota a ser seguida, *a priori*, por um veículo. Uma das melhores abordagens heurísticas para a resolução desse problema é a proposta por Lin e Kernigham (1973) denominada troca de arcos k-opt, onde as trocas 2-opt (2 arcos) e 3-opt (3 arcos) são as mais utilizadas.

Os métodos k-opt buscam a melhoria de um trajeto pela substituição de “k” arcos no roteiro estabelecido anteriormente, isto é, “k” arcos são removidos do roteiro e substituídos por outros “k” arcos. Caso alguma melhoria seja detectada, a troca é aceita e o novo arco passa a compor a solução incumbente. Neste presente trabalho esta dinâmica se repete até que nenhuma troca outra troca resulte em melhoria.

Conforme apontado por Laporte (1999), o processo de melhoria k-opt termina em um mínimo

local e possui ordem de complexidade $O(nk)$. Quanto maior for o valor de k , melhor será a solução, entretanto o esforço computacional requerido também será maior. Isto leva a um *trade-off* entre qualidade e tempo computacional. Dessa forma, trocas 4-opt e superiores ensejam um custo computacional muitas vezes superior à melhoria da solução obtida.

Neste trabalho optou-se por utilizar a troca 2-opt pela sua simplicidade de implementação, uma vez que esta estratégia de melhoria realiza a comparação somente entre dois trechos para então determinar se a inversão de sentido entre eles é mais econômica em relação à rota inicial. Caso seja, a nova rota passa a ser a rota preferencial.

Na Figura 4 abaixo, é ilustrada a sistemática de troca de arcos para a abordagem 2-opt que será utilizada na heurística proposta no trabalho:

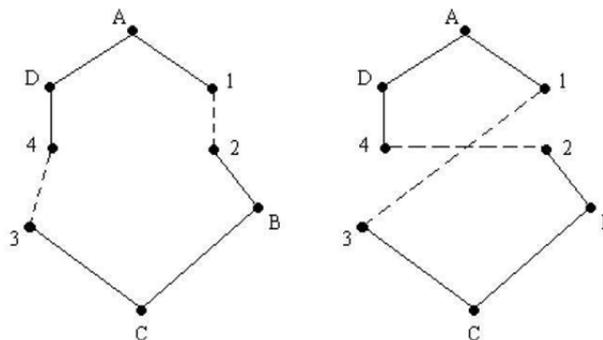


Figura 4 - Troca 2-opt entre os trechos 1-2 e 3-4 Fonte: Adptado de Costa (1997)

Cabe destacar que a geração de boas soluções iniciais é fundamental para que se obtenham boas soluções finais com os métodos k -opt para a melhoria de rotas (COSTA, 1997; CAMPOS, 2008).

3.5 Heurística Proposta

A heurística proposta neste trabalho foi desenvolvida e implementada computacionalmente seguindo a concatenação de algoritmos representada através da Figura 7 a seguir:

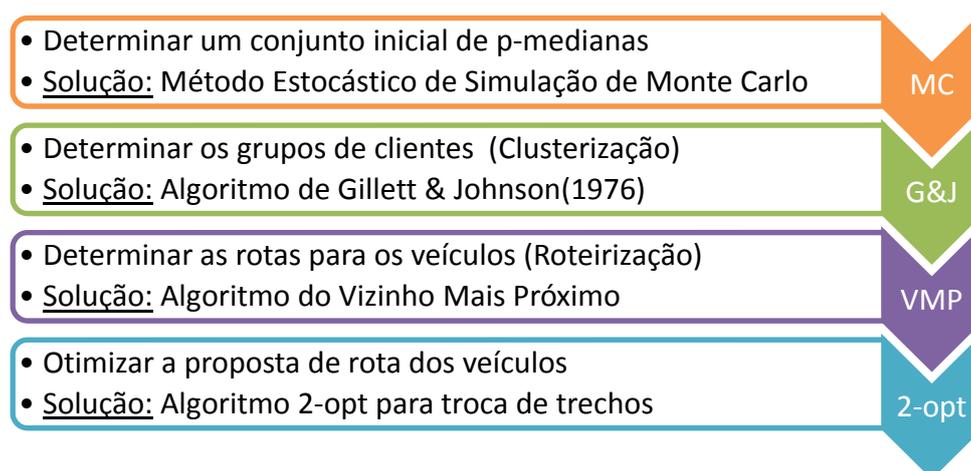


Figura 5 - Heurística proposta para a resolução do PRVC Fonte: Elaborado pelo autor

4. Resultados e Discussões

A abordagem proposta neste trabalho foi avaliada sobre as instâncias clássicas da literatura, disponíveis no sítio <http://www.branchandcut.org>. Utilizou-se dois conjuntos de instâncias resolvidas na otimalidade em Augerat *et al.* (1995). O primeiro conjunto (A – composto de 27 instâncias) diferencia-se do segundo conjunto (B – composto de 11 instâncias) pela dispersão

especial dos pontos (clientes). Nas instâncias do conjunto A os pontos estão dispersos com relativa uniformidade, enquanto que o conjunto B possui pontos dispersos em regiões específicas.

Os testes foram realizados em um processador Intel® Core™2 Quad Q6600, 2.4 GHz com 4 GB de RAM e sistema operacional Windows 7™, 64-Bits. O código fonte do software desenvolvido encontra-se disponíveis em <http://prvc.codeplex.com>.

Para as colunas das tabelas apresentadas a seguir, da esquerda para a direita, tem-se: o nome da instância, a solução ótima (distância euclidiana total para todas as sub-rotas), a solução obtida por este trabalho, o desvio percentual em relação à solução ótima da instância e o tempo de processamento computacional. O número de clientes (incluindo o depósito) está indicado no nome da instância após a letra “n” e o número de veículos após a letra “k”.

A Tabela 1 a seguir apresenta os resultados obtidos para as instâncias da Classe A:

Instância	Número de Pontos	Solução Ótima	Solução Obtida	Desvio da Solução	Tempo (ms)
A-n32-k5	32	784	822,1	4,9%	2
A-n33-k5	33	661	696,2	5,3%	1
A-n33-k6	33	742	784,3	5,7%	2
A-n34-k5	34	778	816,6	5,0%	2
A-n36-k5	36	799	847,9	6,1%	2
A-n37-k6	37	949	989,8	4,3%	2
A-n38-k5	38	730	778	6,6%	1
A-n39-k5	39	822	890	8,3%	2
A-n39-k6	39	831	896,3	7,9%	2
A-n44-k6	44	937	1020,6	8,9%	2
A-n45-k7	45	1146	1251,7	9,2%	2
A-n48-k7	48	1073	1178,2	9,8%	3
A-n53-k7	53	1010	1122,5	11,1%	3
A-n54-k7	54	1167	1267	8,6%	3
A-n55-k9	55	1073	1184,8	10,4%	3
A-n60-k9	60	1354	1507,1	11,3%	3
A-n62-k8	62	1288	1387,5	7,7%	3
A-n63-k10	63	1314	1469,4	11,8%	3
A-n63-k9	63	1616	1815,9	12,4%	4
A-n64-k9	64	1401	1534,3	9,5%	3
A-n65-k9	65	1174	1323,6	12,7%	3
A-n80-k10	80	1763	1985,3	12,6%	5
Média				8,6%	2,5

TABELA 1 – Resultados obtidos pela heurística proposta para o conjunto de instâncias A

Cabe destacar que esta heurística proposta neste estudo resolveu o conjunto de 22 instâncias selecionadas da classe A com um desvio médio percentual em relação ao ótimo de 8,6%. Destaca-se um melhor desempenho para instâncias até 50 pontos com desvios inferiores à 10%. Para as instâncias maiores (de 50 a 80 pontos) o desvio não foi superior à 12,7%. Acerca do tempo de processamento a média para o conjunto de instâncias não ultrapassou 5 milissegundos com valores médios de 2,5 milissegundos (praticamente instantâneo).

Essa redução no tempo de processamento corrobora a utilização de técnicas de paralelismo (*Parallel Computing*), possibilitando a realização de inúmeros cálculos simultâneos através de núcleos diferentes existentes em um mesmo processador.

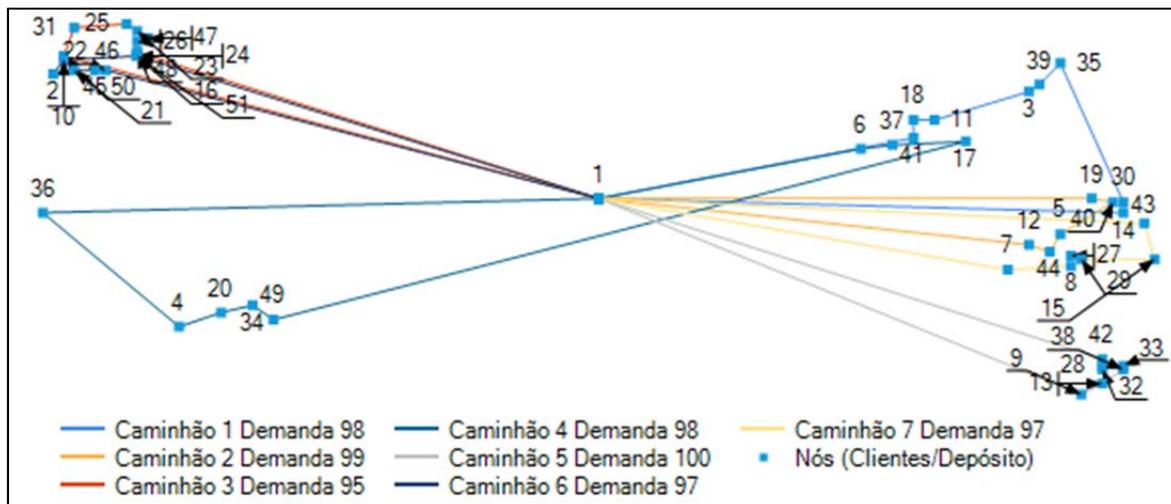
A Tabela 2, a seguir, apresenta os resultados obtidos para as instâncias da Classe B:

Instância	Instância	Solução Ótima	Solução Obtida	Desvio da Solução	Tempo (ms)
B-n31-k5	31	672	682,9	1,6%	1
B-n34-k5	34	788	804	2,0%	1
B-n35-k5	35	955	975,7	2,2%	2
B-n38-k6	38	805	829,8	3,1%	2
B-n39-k5	39	549	579	5,5%	3
B-n41-k6	41	829	863,9	4,2%	2
B-n43-k6	43	742	782	5,4%	2
B-n44-k7	44	909	952,7	4,8%	2
B-n45-k5	45	751	788	4,9%	2
B-n50-k7	50	741	775,6	4,7%	2
B-n50-k8	50	1312	1394,4	6,3%	4
B-n51-k7	51	1032	1046,2	1,4%	3
B-n52-k7	52	747	772,8	3,5%	2
B-n57-k9	57	1598	1689	5,7%	3
B-n63-k10	63	1496	1633	9,2%	3
B-n66-k9	66	1316	1427,1	8,4%	4
B-n68-k9	68	1272	1358,6	6,8%	4
B-n78-k10	78	1221	1346,3	10,3%	5
Média				5,0%	2,6

TABELA 2 – Resultados obtidos pela heurística para o conjunto de instâncias B

O desempenho superior para instâncias do grupo B (desvio médio de 5,0 %) em relação às instâncias do grupo A (desvio médio de 8,6%) podem ser justificados pela dispersão pontual dos clientes. Tal característica possibilita a geração de grupos concentrados para o grupo B, reduzindo a distância do roteiro entre os agrupamentos. Destacam-se as instâncias B-n51-k7 e B-n31-k5 cujo desvio não atingiu 2%. O tempo computacional médio para o mesmo conjunto de dados é de 2,7 milisegundos.

(a) A Instância A-n33-k5



(d) Instância B-n51-k7

Figura 6 – Resolução gráfica para as instâncias selecionadas. Fonte: Elaborado pelos autores.

5. Considerações Finais

O PRVC é um dos mais estudados problemas de otimização, principalmente pela sua grande aplicabilidade em redes logísticas. Mesmo possuindo uma formulação exata relativamente simples, sua resolução analítica é de alta complexidade. Dessa maneira, torna-se inviável o emprego da sua resolução analítica para problemas onde haja um grande número de clientes a serem atendidos. Tal fato propicia, neste contexto, a utilização de heurísticas como instrumentos viáveis de resolução do PRVC, conforme demonstrado pelos resultados obtidos.

O presente trabalho apresentou uma nova heurística baseada na estratégia “*Cluster First Route Second*” para resolver o PRVC. A estratégia utilizada diferencia-se por duas características básicas: o emprego de métodos estocásticos para a geração de uma solução inicial de grupos de clientes e a roteirização do veículo dentro de cada grupo em função do algoritmo VMP. Outro diferencial deste trabalho foi a utilização de processamento paralelo (*Parallel Computing*) na elaboração do software. Essa construção computacional resultou em um tempo de processamento insignificante, evidenciando a ampla vantagem de sua utilização em problemas com alta complexidade de cálculos e dados.

No intuito de validar a precisão da heurística, foram calculados os desvios percentuais em relação às soluções ótimas do problema, com desvio médio de 9,1% para o grupo A de instâncias e 6,3% para o grupo B. Essa margem possibilita uma ampla aplicabilidade da heurística proposta em problemas práticos cotidianos, tendo em vista a necessidade prática de boas soluções com tempo computacional reduzido.

Por fim, como os resultados ainda não estão precisamente próximos da solução ótima, sugere-se o desenvolvimento de trabalhos que visem aperfeiçoar a geração da solução inicial da heurística (problema das p-medianas). Há que se buscar essa melhoria na heurística através da mudança do algoritmo inicial por métodos mais avançados como, por exemplo, algoritmos genéticos.

Referências

AUGERAT, P.; BELENGUER, J.; BENAVENT, E.; CORBERÁN, A.; NADDEF, D.; RINALDI, G. *Computational Results with a Branch and Cut Code for the Capacitated Vehicle Routing Problem*. Technical Report 949-M, Université Joseph Fourier, Grenoble, France, 1995.

CAMPOS, D. S. *Integração de problemas de carregamento e roteamento de veículos, com janela de tempo e frota heterogênea*. 2008. 121 f. Tese (Doutorado em Engenharia de Produção) Universidade de São Paulo, São Paulo.

COSTA, D. M. B. *Aplicação de Algumas técnicas da Pesquisa Operacional na Otimização de Serviços Postais*. 1997. Dissertação (Mestrado em Métodos Numéricos em Engenharia), Universidade Federal do Paraná, Curitiba.

COVER, T; HART, P.; *Nearest neighbor pattern classification*. Stanford: Information Theory, IEEE Transactions on vol.13, no.1, pp. 21- 27, Jan 1967

DANTZIG, G.; RAMSER, R. *The truck dispatching problem*. Management Science, v. 6, p. 80–91, 1959.

FISHER, M. L.; JAIKUMAR, R. *A generalized assignment heuristic for vehicle routing*. Networks, v.11, p.109–124, 1981.

GILLETT, B.; JOHNSON, J. *Multi-terminal vehicle-dispatch algorithm*. Beaumont: Omega, v. 4, p. 711–718, 1976.

HAKIMI, S. L. *Optimum Distribution of Switching Centers in a Communication Network and Some Related Graph Theoretic Problems*. Operations Research, v. 13, p. 462-475, 1965.

LAPORTE, G.; GENDREAU, M.; POTVYN, J.; SEMET, F. *Classical and modern heuristics for the vehicle routing problem*. International Transactions in Operational Research, v. 7, n. 4-5, p. 285-300, 1999.

JUAN, A.; FAULÍN, J.; CABALLÉ, S., BARRIOS, B.; RUIZ, R. *The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing problem*. Applied Soft Computing Journal. no 1, p. 215-224. 2010.

LEIJEN, D.; HALL, J. *Otimizar o código gerenciado para máquinas de vários núcleos*. MSDN Magazine, p. 31-33, Outubro 2007.

SARIKLIS D.; POWELL S. *A heuristic method for the open vehicle routing problem*. Journal of the Operational Research Society, v. 51, p.564–573, 2000.

STEINER, M.; ZAMBONI, L.; COSTA, D.; CARNIERI, C.; SILVA, A. *O Problema de Roteamento no Transporte Escolar*, Revista Pesquisa Operacional, v. 20, n. 01, p. 83-99, 2000.

TAKES, F. W. *Applying Monte Carlo Techniques to the Capacitated Vehicle Routing Problem*. Master (MSc) Computer Science, Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands. 2010. Disponível em: <<http://www.liacs.nl/assets/Masterscripties/2010-01FrankTakes.pdf>>

TEITZ, M. B.; BART, P. *Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted*. Operations Research, v. 16, p. 955-961, 1968.

TILLMAN, F. A.; CAIN, T. M. *An upperbound algorithm for the single and multiple terminal delivery problem*, Management Science, v. 18, n. 11, p. 664-682, 1972.

TOUB, S. *O passado, o presente e o futuro da paralelização de aplicativos .NET*. MSDN Magazine, p. 48-52, Agosto 2011.