

SIMULAÇÃO DE MONTE CARLO E MÉTODOS HEURÍSTICOS PARALELIZADOS PARA A RESOLUÇÃO DO PROBLEMA DO ROTEAMENTO DE VEÍCULOS CAPACITADOS: UMA COMPARAÇÃO ENTRE ABORDAGENS

Thiago André Guimarães

Centro Universitário Franciscano do Paraná
thiandgui@gmail.com

Luis Gustavo Pereira

Universidade Tecnológica Federal do Paraná
lgp1985@yahoo.com.br

Wesley José Nogueira Medeiros

Universidade Tecnológica Federal do Paraná
nogueira.wjnm@gmail.com

Resumo

O Problema do Roteamento de Veículos Capacitados (PRVC) é uma das mais conhecidas abordagens em pesquisa operacional para resolução de problemas em logística de distribuição. Por sua complexidade combinatória o PRVC enseja o emprego de técnicas heurísticas para sua resolução. Neste sentido, este artigo apresenta e compara duas abordagens para a resolução do problema. O método emprega uma estratégia em dois estágios que consiste primeiramente em agrupar os clientes de acordo com a demanda e, posteriormente, construir as rotas para os grupos formados. Na primeira abordagem, o agrupamento dos pontos de demanda é realizado empregando técnicas de simulação de Monte Carlo, juntamente com o algoritmo de designação proposto por Gillet Johnson (1976). Para a segunda abordagem, foi elaborada uma heurística envolvendo os algoritmos clássicos de agrupamento e designação propostos por Teitz e Bart (1968) e Gillet Johnson (1976), respectivamente. Em ambas as abordagens, as rotas iniciais são geradas pela heurística do vizinho mais próximo e refinadas pelas trocas 2-opt. Comparativamente a primeira abordagem obteve desempenho superior à segunda, tanto em termos de eficácia, quanto em eficiência, ressaltando a influência da qualidade do agrupamento dos pontos de demanda sobre a solução final do problema.

Palavras-chave: roteamento de veículos capacitados; simulação de Monte Carlo; procedimentos heurísticos; agrupamento de pontos de demanda; paralelismo.

Abstract

The Capacitated Vehicle Routing Problem is a well-known approach in operations research to solve problems in distribution logistics. For its combinatorial complexity, the CVRP motivates the use of heuristic techniques for its resolution. In this sense, this paper presents and compares two approaches to solving the problem. The method employs a strategy in two stages. The first one builds clusters of customers according to the demand and the second stage builds routes for the costumers groups formed on the first stage. In the first approach, the demand point clustering is done using techniques of Monte Carlo simulation together with the assignment algorithm proposed by Gillet Johnson (1976). For the second approach, it was developed a heuristic evolving the classical clustering and assignment algorithms proposed by Teitz and Bart (1968) and Gillet and Johnson (1976), respectively. In both approaches, the initial routes were generated by the nearest neighbor heuristic and sequentially the routes were improved by the 2-opt procedure. Comparatively, the first approach achieved a higher performance than the second, both in terms of effectiveness, and efficiency, highlighting the influence of the clustering demand point quality on the global solution of the problem.

Keywords: capacitated vehicle routing problem; Monte Carlo simulation, heuristic procedures; demand point clustering, 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 Dantzing 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 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. Cada cliente deve ser atendido por apenas um veículo (embora haja variações no problema original que permitem entregas fracionadas) e, além disso, a demanda transportada não deve exceder sua capacidade de carregamento. O objetivo do PRVC é minimizar a distância total percorrida pela frota de veículos.

Comumente é reportada na literatura uma técnica em duas fases para a resolução do PRVC. Esta estratégia primeiramente agrupa os clientes conforme a localização e quantidade demandada para posteriormente construir rotas para os grupos formados (*cluster first and then route*). Garantido 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), ensejando um roteiro para cada grupo, que passe por todos os pontos e apresente a menor distância possível. Fisher e Jaikumar (1981) e Gillett e Johnson (1976) endereçam heurísticas do tipo “*cluster first and then route*” para a resolução do PRVC. Variações do problema clássico tratadas com essa mesma 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 pela não necessidade de retorno do veículo ao depósito após visitar o último cliente da rota.

Neste sentido, o presente estudo propõe duas heurísticas do tipo “*cluster first and then route*” para resolução do PRVC. A primeira emprega técnicas de simulação de Monte Carlo para agrupamento dos pontos enquanto que a segunda utiliza o algoritmo clássico de Teitz e Bart (1968). Para ambas, a roteirização emprega a heurística do vizinho mais próximo com melhoria 2-opt. Técnicas de processamento paralelo são introduzidas em ambos os casos para reduzir o tempo de processamento. Dessa forma é possível analisar e quantificar o efeito do agrupamento dos pontos sobre o roteiro final.

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 as duas heurísticas são apresentadas. No item 4, discute-se os resultados obtidos enquanto o item 5 tece as conclusões do artigo.

2. REFERENCIAL TEÓRICO

Esta seção introduz formalmente o PRVC e comenta sobre a técnica de processamento paralelo empregada na minimização dos tempos computacionais.

2.1. PROBLEMA DO ROTEAMENTO DE VEÍCULOS CAPACITADOS

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 $l(e)$. 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-rota não excede a capacidade C do veículo.

Uma instância exemplo é apresentada na Figura 1 a seguir. Dentro de cada nó está indicada a demanda referente ao próprio nó.

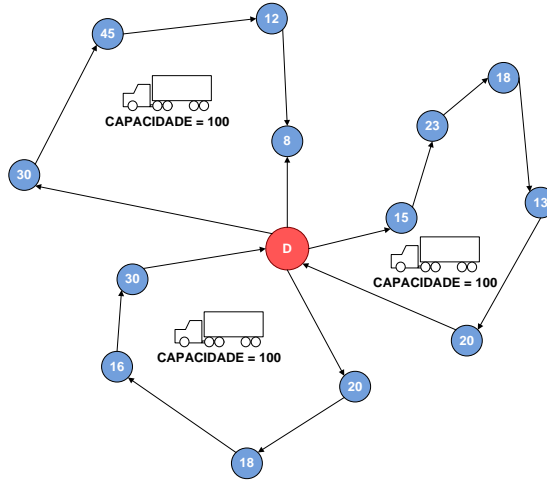


Figura 1 - Instância exemplo do PRVC Fonte: Elaborado pelos autores

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:

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

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

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

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

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 $X_{ijk} \in \{0, 1\}$ ($i \neq j; i, j \in \{0, 1, \dots, N\}$). A equação (1) é a função objetivo que 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. PROCESSAMENTO PARALELO

Leijen e Hall (2007) indica que os computadores de vários núcleos tornaram-se o padrão industrial aplicado atualmente. Infelizmente, segundo os mesmos autores, a maioria dos aplicativos utiliza um só núcleo e não apresenta aumento de velocidade quando executados 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 1 a seguir:

```
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 mesmos possuam alto grau de independência linear. A Figura 3 exemplifica a utilização da extensão PLINQ – *AsParallel()* – em um trecho do código fonte desenvolvido 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 desenvolvido. Fonte: Elaborado pelo autor

3. ESTRATÉGIAS DE RESOLUÇÃO

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.

A construção das heurísticas propostas neste artigo requer a resolução dos problemas de agrupamento de pontos de demanda e posteriormente a roteirização para cada grupo formado. Assim, as subseções a seguir discutem esses problemas de forma pormenorizada. Por fim, as abordagens são apresentadas.

3.1. AGRUPAMENTO DOS PONTOS DE DEMANDA

O problema das p -medianas é introduzido com o objetivo de determinar, no grafo formado pelos clientes, p -pontos principais que representem sementes ou depósitos fictícios. A cada uma destas medianas, os demais pontos de demanda serão designados, de modo a formar p -conjuntos iniciais (p -grupos) de pontos, cada um com “capacidade” igual à capacidade de carregamento dos veículos, que é a mesma para todos eles (frota homogênea). A estratégia de resolução reside na teoria dos grafos que tem por objetivo localizar facilidades (instalações) 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. Uma descrição mais formalizada para o problema é apresentada a seguir.

Para um grafo $G(V, E)$, define-se número de *out*-transmissão e *in*-transmissão, para cada ponto $v_i \in V$, como sendo, respectivamente:

$$\sigma_0(v_i) = \sum_{v_j \in V} \theta_j \cdot \omega(v_i, v_j)$$
$$\sigma_t(v_i) = \sum_{v_j \in V} \theta_j \cdot \omega(v_i, v_j)$$

onde, o vetor $\omega(v_i, v_j)$ é a distância entre o vértice v_i e v_j , e θ_j é o peso associado ao ponto v_j . De acordo com Christofides (1975), são chamadas de *out*-mediana e *in*-mediana de um grafo, respectivamente, os pontos \bar{v}_i e \bar{v}_j que satisfazem as seguintes condições:

$$\begin{aligned}\sigma_0(\bar{v}_0) &= \min_{v_i \in V} [\sigma_0(v_i)] \\ \sigma_t(\bar{v}_t) &= \min_{v_i \in V} [\sigma_t(v_i)]\end{aligned}$$

Para generalizar estes conceitos de *out*-transmissão e *in*-transmissão para p -medianas, considera-se V_p um subconjunto do conjunto de pontos V do grafo $G(V, E)$, que possui p elementos, isto é, a cardinalidade de V_p é p (ou, simplesmente, $|V_p| = p$). Define-se ainda, $\omega(V_p, v_j) = \min_{v_i \in V} [\omega(v_i, v_j)]$ e $\omega(v_j, V_p) = \min_{v_i \in V} [\omega(v_j, v_i)]$, onde $\omega(V_p, v_j)$ representa a distância do subconjunto de pontos V_p até o ponto v_j e $\omega(v_j, V_p)$ indica a distância do vértice v_j até o subconjunto V_p . Analogamente ao procedimento para um só vértice, definem-se os números de *out*-transmissão e *in*-transmissão, respectivamente, para o conjunto V_p , da seguinte forma:

$$\begin{aligned}\sigma_0(V_p) &= \sum_{v_j \in V} \theta_j \cdot \omega(V_p, v_j) \\ \sigma_t(V_p) &= \sum_{v_j \in V} \theta_j \cdot \omega(v_j, V_p)\end{aligned}$$

Finalmente, denomina-se de p -*out*-mediana e p -*in*-mediana os conjuntos \bar{V}_{po} e \bar{V}_{pt} , respectivamente, para os quais:

$$\begin{aligned}\sigma_0(\bar{V}_{po}) &= \min_{V_p \subset V} [\sigma_0(V_p)] \\ \sigma_t(\bar{V}_{pt}) &= \min_{V_p \subset V} [\sigma_t(V_p)]\end{aligned}$$

Neste trabalho, considera-se um grafo não orientado, tornando indiferente o conceito de p -*in*-mediana ou p -*out*-mediano, que doravante será denominado apenas de p -mediana, simplificando consideravelmente a notação utilizada.

O problema das p -medianas pode ser resolvido de forma exata (HAKIMI, 1965), usando enumeração exaustiva ou programação inteira. Todavia, para problemas de grande porte (grande quantidade de pontos), torna-se necessário recorrer a métodos aproximados (heurísticos) tendo em vista o esforço computacional requerido pelos métodos exatos ser muito elevado.

3.1.1. RESOLUÇÃO DO PROBLEMA DAS P -MEDIANAS VIA SIMULAÇÃO DE MONTE CARLO

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. Com este avanço, a estratégia aplicando o método estocástico de Monte Carlo para resolução do problema das p -medianas, para posterior designação dos pontos de demanda às medianas geradas se apresenta viável.

Para a determinação destes p -pontos iniciais, foi utilizado o gerador de números pseudoaleatórios existente no *.NET Framework 4*. Essa ferramenta garante que a escolha dos pontos iniciais é feita de maneira estocástica com distribuição normal. A orientação citada 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. Dessa forma, para cada p -ponto do problema gera-se dois números pseudoaleatórios, sendo um para a coordenada x e outro para a coordenada y do ponto.

Os números pseudoaleatórios pertencem ao intervalo entre a mínima e a máxima abcissa e a mínima e máxima ordenada definida anteriormente. 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. A Figura 4 apresenta o algoritmo.

	Procedimento MonteCarlo (destinos, quantidadeVeiculos, capacidadeVeiculo)
1	Início
2	$xMin \leftarrow \text{Min destinos.X}$
3	$xMax \leftarrow \text{Max destinos.X}$
4	$yMin \leftarrow \text{Min destinos.Y}$
5	$yMax \leftarrow \text{Max destinos.Y}$
6	Para $iteração \leftarrow 1$ até $quantidadeVeiculos$ faça
7	RetornoIterado $nova\ mediana(X \leftarrow \text{aleatório} * (xMax - xMin) + xMin,$
8	$Y \leftarrow \text{aleatório} * (yMax - yMin) + yMin,$
9	$capacidade \leftarrow capacidadeVeiculo)$
10	FimPara
11	Fim

Figura 4 - Implementação do Monte Carlo, Fonte: Elaborado pelos Autores

3.1.2. RESOLUÇÃO DO PROBLEMA DAS P-MEDIANAS VIA TEITZ E BART (1968)

Ainda que exista na literatura uma vasta gama de abordagens heurísticas para a resolução do problema das p -medianas, destaca-se o algoritmo proposto por Teitz e Bart (1968) pela sua eficiência, estabilidade e poder de generalização. Nele, inicialmente escolhe-se um conjunto S formado por p pontos, considerado como uma aproximação do conjunto \bar{v} das medianas. Verifica-se se algum ponto $v_i \in (V - S)$ pode substituir, de acordo com o algoritmo apresentado a seguir, algum ponto $v_j \in V$, produzindo um novo conjunto S' tal que: $S' = S \cup \{v_i\} - \{v_j\}$ e $\sigma(S') < \sigma(S)$. Se isto for possível, substituímos v_j por v_i e S' é considerado uma nova aproximação para o conjunto S , onde nenhuma substituição de pontos produza um número de transmissão menor. O algoritmo é detalhadamente descrito a seguir.

<p>Passo 1. Construa um conjunto inicial S, com p elementos de V;</p> <p>Passo 2. Rotule todos os pontos $v_i \notin S$ como “não-analisados”;</p> <p>Passo 3. Enquanto existirem pontos “não-analisados” no conjunto $(V - S)$, faça:</p> <ol style="list-style-type: none"> Selecione um vértice “não-analisado” $v_i \in (V - S)$, e calcule a redução Δ_{ij} do número de transmissão, $\forall v_j \in S: \Delta_{ij} = \sigma(S) - \sigma(S \cup \{v_i\} - \{v_j\})$; Faça $\Delta_{ijo} = \max_{v_j \in S} \{\Delta_{ij}\}$; Se $\Delta_{ijo} > 0$ faça $S \leftarrow S \cup \{v_i\} - \{v_j\}$ rotulando v_{jo} como “analisado”; Se $\Delta_{ijo} \leq 0$ rotule v_i como “analisado”. <p>Passo 4. Se durante a execução do passo 3 ocorrer modificações no conjunto S, volte para o passo 2. Caso contrário PARE. O conjunto S será uma aproximação para o problema das p-medianas.</p>
--

Figura 5 - Heurística de Teitz e Bart. Fonte: Teitz e Bart (1968)

3.2. DESIGNAÇÃO DOS PONTOS ÀS MEDIANAS

Uma vez obtido o conjunto de medianas S , faz-se necessário designar de maneira ótima (ou quase ótima) os pontos pertencentes ao conjunto $(V - S)$ a elas formando, desta maneira, os agrupamentos serão atendidos pela frota disponível de veículos. A designação é realizada respeitando a capacidade da mediana (determinada pela capacidade de carregamento do veículo). Para a designação dos pontos, Gillet e Johnson (1976) propuseram um procedimento heurístico conhecido como algoritmo de Gillet e Johnson.

A ideia básica do referido algoritmo é selecionar os pontos que apresentam a maior razão entre as distâncias às duas medianas mais próximas para serem designados com prioridade. Este procedimento prioriza a designação de pontos próximos às medianas. A figura 5 apresenta o algoritmo.

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 6 - Algoritmo de Gillet e Johnson. Fonte: Gillet e Johnson (1976)

3.3. PROBLEMA DA ROTEIRIZAÇÃO: GERAÇÃO DA SOLUÇÃO INICIAL

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 na 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 está exposto na Figura 7 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 7 - Algoritmo VMP. Fonte: Cover e Hart (1967)

3.4. PROBLEMA DA ROTEIRIZAÇÃO: REFINAMENTO 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 Kernighan (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 artigo, 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.

Aqui 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 7 abaixo, é ilustrada a sistemática de troca de arcos para a abordagem 2 -opt que será utilizada na heurística proposta no trabalho:

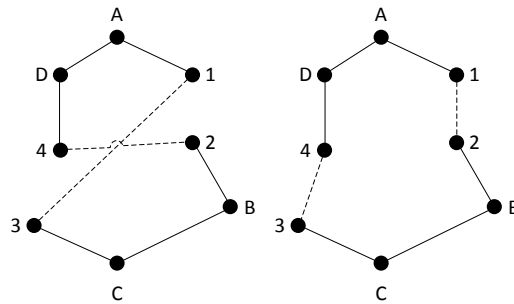


Figura 8 - Troca 2-opt entre os trechos 1-2 e 3-4 Fonte: Adptado de Costa (1997)

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).

3.5. HEURÍSTICAS PROPOSTAS

A primeira abordagem para a resolução do PRVC inicia-se com o agrupamento dos clientes via simulação de Monte Carlo, ajustado *a priori*, pela definição da quantidade de medianas conforme a quantidade de veículos disponíveis. Na segunda abordagem, a definição da p-medianas é feita através do algoritmo de Teitz e Bart (1968), também ajustado para a quantidade de veículos. A designação dos pontos às medianas é feito pelo algoritmo de Gillet e Johnson (1976), formando os agrupamentos de clientes para serem roteirizados. Na sequência cada agrupamento é roteirizado, sendo que a rota inicial é gerada pela heurística construtiva do vizinho mais próximo e refinada pelas heurísticas de melhoria 2-opt. As duas abordagens completas são apresentadas nas Figura 9 e Figura 10.

	Procedimento SMC&GJ+VMP+2-opt (instância)
1	Início
2	iteração ← 0
3	Enquanto iteração < Número Máximo faz
4	SoluçãoInicial ← ProcessarMonteCarloComGilletJohnson(instância)
5	Roteirização ← ProcessarVizinhoMaisProximo(SoluçãoInicial, instância)
6	Resposta ← 2-Opt(Roteirização, instância)
7	iteração ← iteração + 1
8	Se Resposta.CustoTotal < Solução.CustoTotal então
9	Solução ← Resposta
10	FimSe
11	FimEnquanto
12	Retorna Solução
13	Fim

Figura 9 - Abordagem 1 – Simulação de Monte Carlo. Fonte: Elaborado pelos Autores

	Procedimento TB&GJ+VMP+2-opt (instância)
1	Início
2	iteração ← 0
3	Enquanto iteração < Número Máximo faz
4	SoluçãoInicial ← ProcessarTeitzBartComGilletJohnson(instância)
5	Roteirização ← ProcessarVizinhoMaisProximo(SoluçãoInicial, instância)
6	Resposta ← 2-Opt(Roteirização, instância)
7	iteração ← iteração + 1
8	Se Resposta.CustoTotal < Solução.CustoTotal então
9	Solução ← Resposta
10	FimSe
11	FimEnquanto
12	Retorna Solução
13	Fim

Figura 10 - Abordagem 2 – Teitz e Bart. Fonte: Elaborado pelos Autores

4. RESULTADOS OBTIDOS

Os experimentos computacionais foram realizados sobre as instâncias clássicas da literatura, disponíveis no sítio <http://www.branchandcut.org>. Foram selecionadas 31 instâncias de 6 classes distintas. A diferença existente entre as classes de instâncias pode variar desde a região onde os

clientes estão distribuídos até a densidade dos mesmos em relação à área do plano. Quanto ao hardware utilizado, às heurísticas foram executadas em um processador Intel® Core™2 Quad Q6600, 2.4 GHz com 4 GB de RAM e sistema operacional Windows 7TM, 64-Bits.

Para a primeira abordagem, que emprega simulação de Monte Carlo para resolução das p -medianas, foram feitas 1000 iterações para cada instância. Já na segunda abordagem, esse número foi limitado em 100, visto que o algoritmo de Teitz e Bart (1968) demanda um tempo computacional consideravelmente maior.

A tabela 1, apresentada a seguir, contém os resultados para o conjunto de 31 instâncias testadas. Na primeira coluna está o nome da instância. Já na segunda (*BKS – Best Known Solution*) apresenta-se a melhor solução conhecida (solução ótima). As quatro próximas apresentam para abordagem 2, a melhor solução encontrada dentre todas as iterações, o desvio percentual em relação à BKS, o tempo de processamento da iteração que gerou a melhor solução e o tempo total de processamento para todas as iterações. As outras quatro colunas contém as mesmas informações para a abordagem 1.

Tabela 1 - Síntese dos Resultados Obtidos. Fonte: Elaborado pelos Autores

Instancia	BKS	TEITZ & BART (ABORDAGEM 2)				SIMUL. DE MONTE CARLO (ABORDAGEM 1)			
		Melhor Solução Encontrada	Desvio (%)	Tempo Melhor Solução (s)	Tempo Total (s)	Melhor Solução Encontrada	Desvio (%)	Tempo Melhor Solução (s)	Tempo Total (s)
A-n33-k6	742	807	8,7	0,01	2,12	761	2,5	0,00	2,06
A-n34-k5	778	836	7,5	0,01	1,77	790	1,5	0,01	2,26
A-n37-k6	949	1026	8,1	0,01	2,71	989	4,2	0,00	2,25
A-n39-k5	822	860	4,6	0,02	2,26	852	3,6	0,00	3,00
A-n39-k6	831	895	7,6	0,03	3,10	868	4,4	0,00	2,54
A-n44-k6	937	964	2,9	0,03	3,82	984	5,0	0,00	3,04
A-n48-k7	1073	1153	7,5	0,04	6,04	1133	5,6	0,00	3,59
A-n62-k8	1288	1395	8,3	0,12	13,20	1403	8,9	0,00	4,96
A-n80-k10	1763	1935	9,7	0,27	34,53	1920	8,9	0,02	6,60
B-n31-k5	672	689	2,5	0,01	1,22	683	1,7	0,00	1,96
B-n35-k5	955	1042	9,1	0,01	1,84	962	0,7	0,00	2,43
B-n38-k6	805	863	7,2	0,02	2,36	828	2,8	0,00	2,46
B-n44-k7	909	979	7,7	0,03	4,87	942	3,6	0,00	2,97
B-n50-k8	1312	1436	9,5	0,07	7,93	1361	3,7	0,00	3,85
B-n52-k7	747	769	2,9	0,05	6,34	780	4,4	0,00	4,00
B-n57-k9	1598	1726	8,0	0,10	13,14	1636	2,4	0,00	3,94
B-n78-k10	1221	1327	8,7	0,31	33,35	1350	10,6	0,00	6,48
E-n22-k4	375	456	21,6	0,01	0,58	375	0,0	0,00	1,24
E-n23-k3	569	635	11,7	0,01	0,54	569	0,0	0,00	1,63
E-n30-k3	534	536	0,3	0,01	0,77	536	0,3	0,00	2,24
G-n262-k25	6119	6717	9,8	40,81	4092,07	6676	9,1	0,06	34,88
M-n101-k10	820	835	1,8	0,48	67,61	909	10,9	0,02	11,10
M-n121-k7	1034	1097	6,1	0,31	49,14	1089	5,3	0,11	25,15
M-n200-k17	1373	1443	5,1	7,72	939,09	1555	13,3	0,06	27,86
P-n16-k8	450	473	5,0	0,00	0,50	451	0,3	0,00	0,83
P-n20-k2	216	227	4,9	0,00	0,37	219	1,4	0,00	1,58
P-n21-k2	211	215	1,8	0,00	0,37	215	1,8	0,00	1,80
P-n22-k2	216	220	1,9	0,00	0,40	220	1,9	0,00	1,81
P-n45-k5	510	539	5,6	0,03	2,73	535	4,8	0,00	4,09
P-n50-k7	554	599	8,2	0,06	6,47	594	7,2	0,00	3,62
P-n51-k10	741	801	8,1	0,07	11,89	810	9,3	0,00	2,97
Média			6,9	1,63	171,4		4,5	0,01	5,78

Pelos resultados obtidos, verifica-se que em termos médios, a abordagem 1, que emprega a simulação de Monte Carlo para a resolução do problema das p -medianas, produz resultados superiores à utilização do algoritmo de Teitz e Bart (1968) para o mesmo fim (abordagem 2). Destaca-se o tempo computacional bastante reduzido em virtude do processamento paralelo utilizado. O tempo médio de processamento para cada instância na abordagem 1 foi praticamente nulo, demandando em média 5,78 segundos, contra 171,4 da abordagem 2. Destacam-se ainda as instâncias E-n22-k4 e E-n23-k3, onde a abordagem 1 obteve o resultado ótimo (desvio igual à 0%), com tempo de processamento para todas as 1000 iterações pouco superior à 1 segundo. Os tempos computacionais indicado como zero, na coluna

11 (tempos para processamento de 1 iteração) não alcançaram representatividade ao nível de precisão apresentada.

Mesmo desconsiderando-se as instâncias G-n262-K25 e M-n200-K1 ainda temos tempos médios 2,42x mais rápidos para o processamento de todas as instâncias, sendo que a abordagem 1 teve uma ordem de grandeza a mais na execução de 1000 instâncias, contra as 100 analisadas na abordagem 2.

Os resultados da abordagem 2 foram superiores à abordagem 1 para as instâncias grifadas em vermelho. Em termos médios, o desvio da abordagem 1 foi de 4,5% em relação à melhor solução conhecida, ao passo que a abordagem 2 produziu um desvio 6,9% superior. Sendo assim, conclui-se que a técnica que emprega simulação de Monte Carlo produz um melhor agrupamento dos pontos de demanda e justamente esse melhor agrupamento define o melhor resultado da rota.

Ainda que a abordagem 1 apresente resultados superiores, há casos em que ocorre o contrário. Pelos desvios gerados, os resultados corroboram a eficiência de ambas as heurísticas para resolução do PRVC, visto que a introdução de técnicas de processamento paralelo possibilitou a resolução de todas as instâncias em tempo muito reduzido. A critério de visualização, as Figura 11 e Figura 12 a seguir apresentam a resolução gráfica para as instâncias E-n23-k e E-n22-k4 nas duas abordagens. É possível notar que os agrupamentos formados por simulação de Monte Carlo são mais regionalizados, minimizando os cruzamentos entre as sub-rotas. A solução da abordagem 1 é o ótimo.

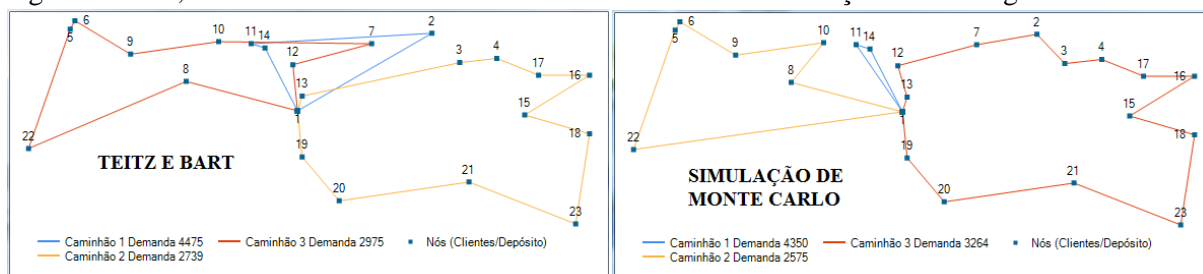


Figura 11 - Solução para a instância E-n23-k3

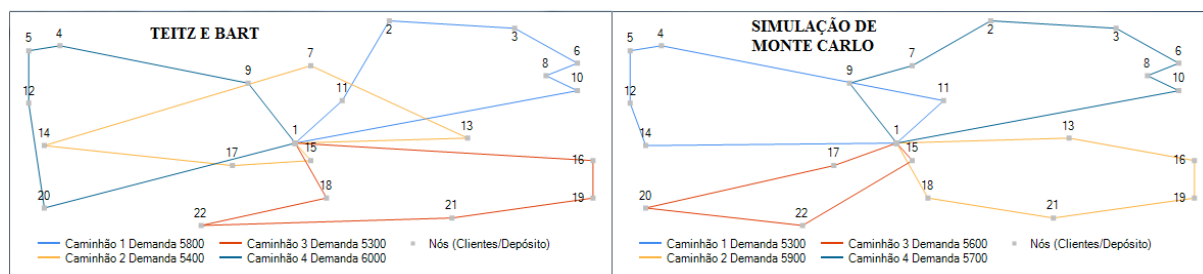


Figura 12 - Solução para a instância E-n22-k4

5. CONCLUSÕES

O presente artigo apresentou duas propostas para a resolução do Problema do Roteamento de Veículos Capacitados, com base na estratégia “*Cluster First and then Route*”, que consiste primeiramente em agrupar os pontos de demanda para posteriormente construir as rotas para os grupos formados.

Para a primeira abordagem proposta, empregou-se simulação de Monte Carlo em conjunto com o algoritmo de Gillet Johnson (1976) para a resolução do problema do agrupamento. A roteirização dos pontos foi feita através a heurística do vizinho mais próximo, com refinamento 2-opt. A segunda abordagem diferencia-se da primeira por utilizar o algoritmo clássico de Teitz e Bart (1968) para a resolução do problema das p -medianas no lugar da simulação de Monte Carlo. Em ambos os casos, técnicas de processamento paralelo foram utilizadas para reduzir o tempo de processamento.

Os resultados obtidos demonstram que ambas as abordagens são eficientes para a resolução do problema, visto que, em termos médios, o desvio da solução com relação ao melhor resultado conhecido foi de 4,5% para a primeira abordagem (via simulação) e 6,9% para a segunda abordagem (via Teitz e Bart), com tempo de processamento bastante reduzido para ambas.

Todavia, a análise comparativa entre elas aponta um desempenho superior da abordagem 1 sobre a abordagem 2, demonstrando que a técnica de simulação de Monte Carlo é mais eficiente para a resolução do problema das p-medianas. Neste sentido, conclui-se que a qualidade do agrupamento formado é fundamental para a obtenção de boas soluções finais para o PRVC.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] CHRISTOFIDES, N., **Graph Theory: An algorithmic approach**. New York: Academic Press Inc, 1975.
- [2] COSTA, D. M. B. **Aplicação de Algumas técnicas da Pesquisa Operacional na Otimização de Serviços Postais**. Dissertação (Mestrado em Métodos Numéricos em Engenharia), Universidade Federal do Paraná, Curitiba, 1997.
- [3] COVER, T.; HART, P.; Nearest neighbor pattern classification. Stanford: Information Theory, **IEEE Transactions on Operational Research**, v.13, n.1, p. 21- 27, 1967.
- [4] DANTZIG, G.; RAMSER, R. The truck dispatching problem. **Management Science**, v. 6, p. 80–91, 1959.
- [5] FISHER, M. L.; JAIKUMAR, R. A generalized assignment heuristic for vehicle routing. **Networks**, v.11, p.109–124, 1981.
- [6] GILLETT, B.; JOHNSON, J. Multi-terminal vehicle-dispatch algorithm. **Omega**, v. 4, p. 711–717, 1976.
- [7] 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.
- [8] 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.
- [9] LAPORTE, G.; GENDREAU, M.; POTVIN, J. Y.; 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.
- [10] LEIJEN, D.; HALL, J. Otimize o código gerenciado para máquinas de vários núcleos. **MSDN Magazine**, p. 31-33, Outubro 2007.
- [11] LIN, S.; KERNIGHAN, B. W. An Effective Heuristic Algorithm for the Traveling Salesman Problem, **Operations Research**, v. 21, p. 498-516, 1973.
- [12] LIN, S. W.; LEE, Z. J.; YING, K. C.; LEE, C. Y. Applying hybrid meta-heuristics for capacitated vehicle routing problem. **Expert Systems with Applications**, v. 36, p. 1505–1512, 2009.
- [13] 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.
- [14] TEITZ, M. B.; BART, P. Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted. **Operations Research**, v. 16, p. 955-961, 1968.
- [15] TOUB, S. O passado, o presente e o futuro da paralelização de aplicativos .NET. **MSDN Magazine**, p. 48-52, Agosto 2011.