

Abordagem Heurística Paralelizada para a Resolução do Problema do Roteamento de Veículos Capacitados

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

Luis Gustavo Pereira - lgp1985@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 os algoritmos clássicos de Teitz e Bart (1968) e 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 computacional. Testes realizados com as instâncias clássicas disponíveis na literatura apontaram um desempenho médio de 14,1% em relação ao resultado ótimo dos problemas, com a vantagem de consumir um tempo de processamento computacional inferior a 1 segundo para a resolução de cada instância.

Palavras chave: Roteamento de veículos capacitados; Procedimentos heurísticos; Agrupamento de pontos de demanda.

Parallel Heuristic Approach to solve the Capacitated Vehicle Routing Problem

Abstract

The Capacitated 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 classical algorithms Teitz and Bart (1968) and Gillett and Johnson (1976) 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 performance of 14,1% instead. The results took an advantage of consuming a computational processing time shorter than 1 second.

Key-words: Capacitated 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 introduzido 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 técnicas heurísticas e mateheurísticas.

O objetivo deste trabalho é propor uma nova abordagem heurística para a resolução do Problema de Roteamento de Veículos Capacitados (PRVC). Esta abordagem baseia-se na estratégia “*Cluster First Second Route*”, empregando heurísticas clássicas e técnicas de paralelismo para a redução do tempo de processamento 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 serão descritos as características do PRVC e comentado a proposta de utilização heurística. No item 3, serão apresentados os principais algoritmos necessários à construção das estratégias propostas. Nesse mesmo tópico a heurística completa é apresentada. No item 4, serão apresentados os resultados obtidos, enquanto o item 5 tece as conclusões e aponta 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-rota não excede a capacidade C do veículo.

Uma instância exemplo é apresentada na figura 1 a seguir:

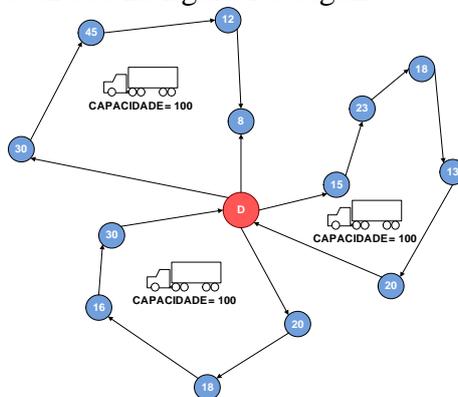


Figura 1 - Instância exemplo do PRVC Fonte: Elaborado pelo autor

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 $X_{ij}^k \in 0$ ou 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. Todavia, segundo os mesmos 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 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 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 práticas.

A construção da heurística proposta neste artigo requer a resolução dos problemas da p-medianas, da designação ou agrupamento (clusterização), da construção e 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. 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.

Dentre os métodos heurísticos disponíveis, destaca-se o algoritmo proposto por Teitz e Bart (1968) pela sua eficiência, estabilidade e poder de generalização. Tal algoritmo é detalhadamente descrito pela Figura 4 - Heurística de Teitz e Bart. Fonte: Teitz e Bart (1968) Figura 4 a seguir:

- Passo 1.** Construa um conjunto inicial S , com p elementos de V ;
- Passo 2.** Rotule todos os pontos $v_i \notin S$ como “não-analisados”;
- Passo 3.** Enquanto existirem pontos “não-analisados” no conjunto $(V - S)$, faça:
- 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 $\Delta_{j_0} = \max_{v_j \in S} \{\Delta_{ij}\}$;
 - Se $\Delta_{j_0} > 0$ faça $S \leftarrow S \cup \{v_i\} - \{v_j\}$ rotulando v_{j_0} como “analisado”;
 - Se $\Delta_{j_0} \leq 0$ rotule v_i como “analisado”.
- 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.

Figura 4 - Heurística de Teitz e Bart. Fonte: Teitz e Bart (1968)

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 5 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 5 - 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 está exposto na Figura 6 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 6 - 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 7 abaixo, é ilustrada a sistemática de troca de arcos para a abordagem 2-opt que será utilizada na heurística proposta no trabalho:

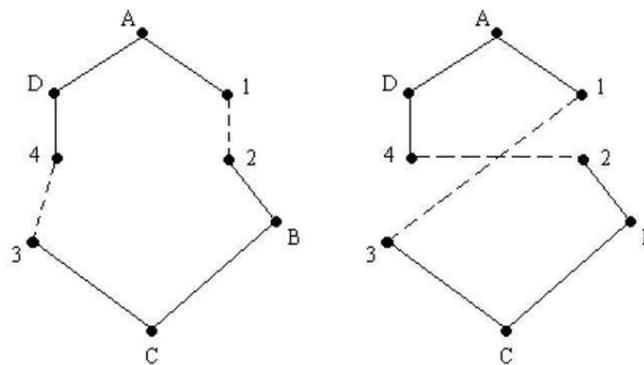


Figura 7 - Troca 2-opt entre os trechos 1-2 e 3-4 Fonte: Adptado de Costa (1997)

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

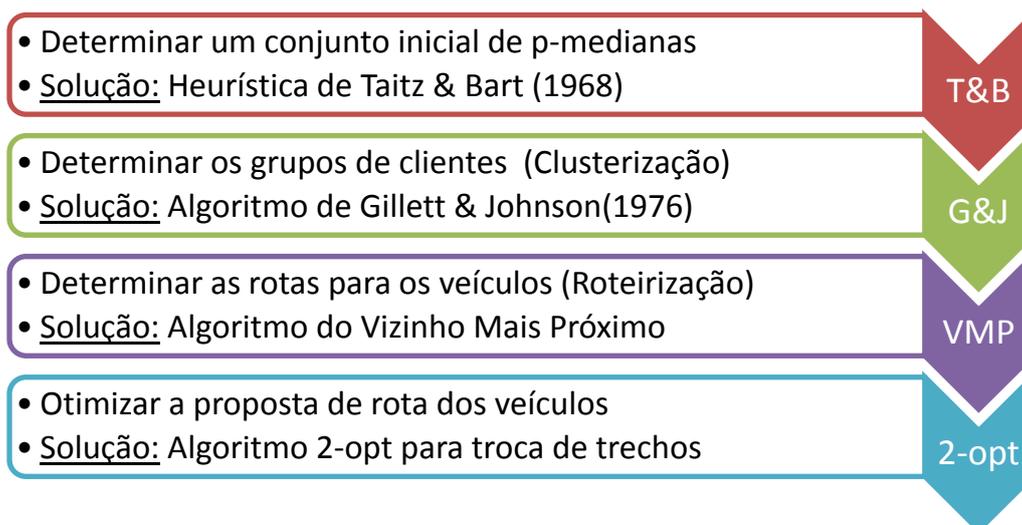


Figura 8 - Heurística proposta para a resolução do PRVC Fonte: Elaborado pelo autor

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 espacial 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 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	951,3	21,3%	20
A-n33-k5	33	661	783	18,5%	13
A-n33-k6	33	742	830,6	11,9%	19
A-n34-k5	34	778	865	11,2%	23
A-n37-k6	37	949	1037,5	9,3%	33
A-n38-k5	38	730	866,1	18,6%	21
A-n39-k5	39	822	922,8	12,3%	34
A-n39-k6	39	831	917,2	10,4%	39
A-n44-k6	44	937	1017,9	8,6%	46
A-n45-k6	45	944	1068,4	13,2%	55
A-n45-k7	45	1146	1329,8	16,0%	65
A-n46-k7	46	914	1067,3	16,8%	75
A-n48-k7	48	1073	1203,7	12,2%	81
A-n55-k9	55	1073	1287,1	20,0%	233
A-n60-k9	60	1354	1624,6	20,0%	277
A-n62-k8	62	1288	1436,3	11,5%	206
A-n63-k10	63	1314	1485,8	13,1%	397
A-n63-k9	63	1616	1935,2	19,8%	361
A-n64-k9	64	1401	1694,2	20,9%	357
A-n65-k9	65	1174	1456,1	24,0%	326
A-n69-k9	69	1159	1417,7	22,3%	355
A-n80-k10	80	1763	2020	14,6%	606
Média				15,7%	165,5

TABELA 1 – Resultados obtidos pela heurística proposta para o conjunto de instâncias A – Fonte: Elaborado pelos Autores

Cabe destacar que esta heurística proposta neste estudo resolveu o conjunto de 22 instâncias selecionadas da classe. Demonstra-se pelos resultados um certo poder de generalização da heurística, visto que os resultados não apresentaram grande dispersão em relação dos desvios percentuais da solução ótima. Acerca do tempo de processamento a média para o conjunto de

instâncias não ultrapassou 81 milissegundos para instâncias até 50 pontos com valores médios de 165,5 milissegundos para o conjunto selecionado. 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.

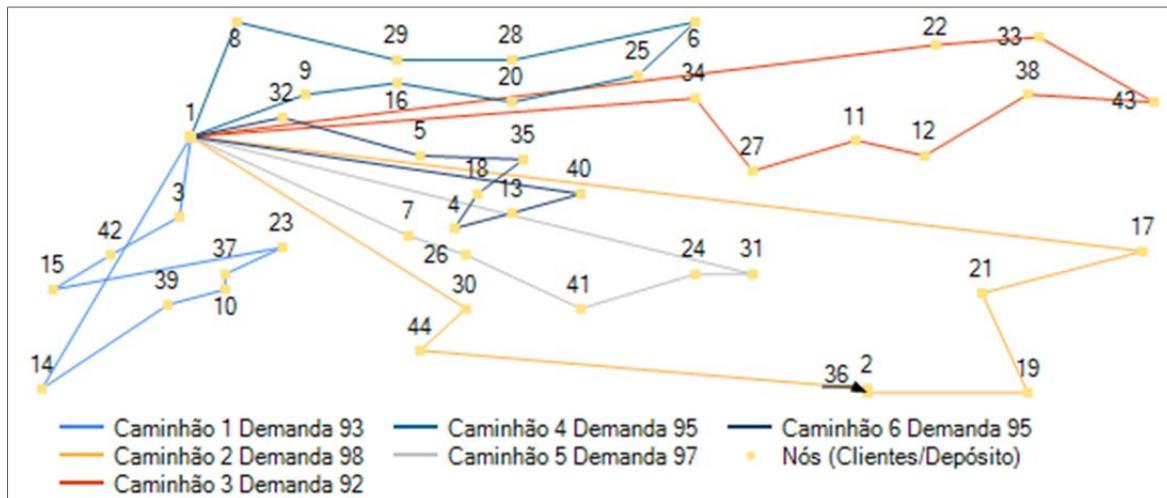
A Tabela 2, a seguir, apresenta os resultados obtidos para as instâncias da Classe B:

Instância	Número de Pontos	Solução Ótima	Solução Obtida	Desvio da Solução	Tempo (ms)
B-n31-k5	31	672	694,1	3,3%	14
B-n34-k5	34	788	940,95	19,4%	20
B-n35-k5	35	955	1053,36	10,3%	16
B-n38-k6	38	805	864,49	7,4%	31
B-n43-k6	43	742	832,19	12,2%	41
B-n44-k7	44	909	989,48	8,9%	75
B-n45-k5	45	751	929,75	23,8%	53
B-n50-k7	50	741	865,3	16,8%	120
B-n50-k8	50	1312	1448,12	10,4%	122
Média				12,5%	54,7

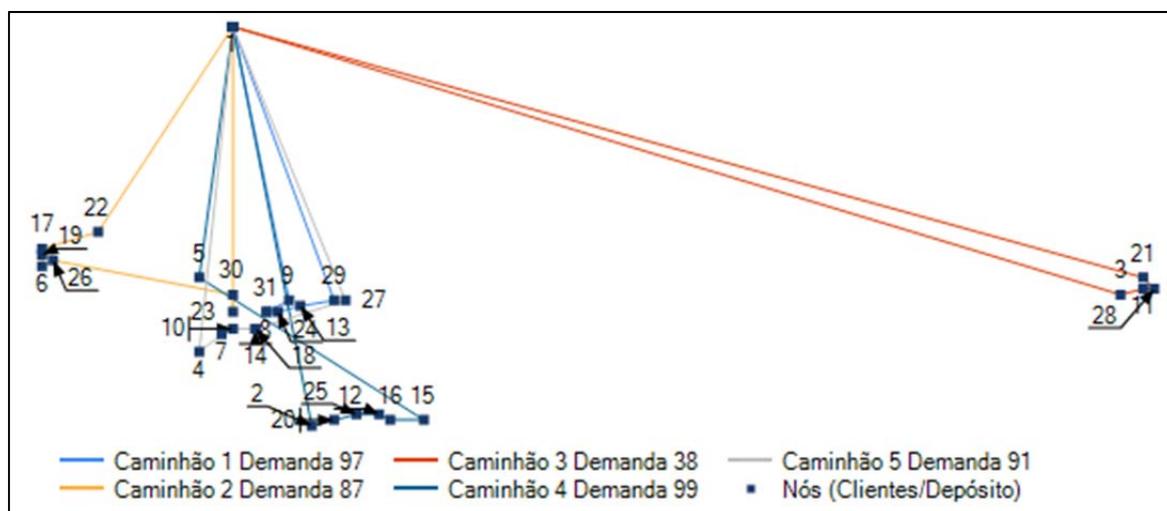
TABELA 2 – Resultados obtidos pela heurística para o conjunto de instâncias B

O desempenho superior para instâncias do grupo B (desvio médio de 12,5 %) em relação às instâncias do grupo A (desvio médio de 15,7%) 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-n31-k5 e B-n38-k6 cujo desvio não atingiu 8%. O tempo computacional médio para o mesmo conjunto de instâncias foi de 54,7 milissegundos.

A Figura 9 a seguir apresenta uma resolução gráfica para a instância A-n44-k6 (a) que possui um desvio de 8,6 % em relação à solução ótima, em conjunto com a instância B-n31-k5 (b) com desvio de 3,3%, destacando a dispersão pontual dessa instância.



(a) Instância A-n44-k6



(b) Instância B-n31-k5

Figura 9 – Resolução gráfica para instâncias selecionadas. Fonte: Elaborado pelos autores.

5. Considerações Finais

O PRVC é um dos mais analisados 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: a determinação de grupos de clientes a serem atendidos em função da capacidade do veículo 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 significativamente reduzido, 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 15,7% para o grupo A de instâncias e 12,5% para o grupo B. Essa margem possibilita a aplicabilidade adequada da heurística proposta em problemas práticos cotidianos, sem demandar um tempo computacional elevado.

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 ou mesmo métodos estocásticos.

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.
- LEIJEN, D.; HALL, J.** *Otimize o código gerenciado para máquinas de vários núcleos*. MSDN Magazine, p. 31-33, Outubro 2007.
- LIN, S.; KERNIGHAN, B. W.** *An effective heuristic algorithm for the traveling salesman problem*. Operations Research, v. 21, p. 498-516, 1973.
- 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.
- 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.