

DEPARTAMENTO DE COMPUTAÇÃO
D E C O M

Resolução do problema de Roteamento de Veículos com Frota
Heterogênea e Janelas de Tempo

Tiago Araújo Neves – 01.1.4126

Marcone Jamilson Freitas Souza
Alexandre Xavier Martins

Relatório Técnico DECOM 01/2004

U F O P
UNIVERSIDADE FEDERAL DE OURO PRETO

Tiago Araújo Neves

RESOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE
VEÍCULOS COM FROTA HETEROGÊNEA E JANELAS DE
TEMPO

MONOGRAFIA SUBMETIDA AO DEPARTAMENTO DE CIÊNCIA DA
COMPUTAÇÃO DA UNIVERSIDADE FEDERAL DE OURO PRETO COMO PARTE
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DE APROVAÇÃO NA
DISCIPLINA DE PROJETO ORIENTADO.

Aprovada por:

Prof. Dr. Marcone Jamilson Freitas Souza
Doutor pela COPPE/Universidade Federal do Rio de Janeiro
Departamento de Ciência da Computação, UFOP

Prof. Alexandre Xavier Martins
Mestrando pela Universidade Federal de Minas Gerais
Departamento de Ciência da Computação, UFOP

Prof. Dr. Álvaro Guarda
Doutor pelo Institut National Polytechnique de Grenoble - França
Departamento de Ciência da Computação, UFOP

OURO PRETO, MG – BRASIL
JUNHO DE 2004

Resumo

Este trabalho tem seu enfoque no Problema de Roteamento de Veículos com Frota Heterogênea e Janelas de Tempo (PRVFHJT) de uma empresa de transporte. Este problema consiste na construção de rotas, que contém uma série de clientes a serem visitados e que serão percorridas por veículos, de tal forma que as rotas sejam executadas com o menor custo possível.

O número elevado de clientes e de suas possíveis ordens de visita para constituir as rotas, associado com a necessidade de respeitar as regras operacionais das empresas e de minimizar custos, torna altamente complexa a atividade de construção de rotas. Na literatura o PRVFHJT é considerado NP-difícil, o que dificulta, ou até mesmo impossibilita a resolução de casos reais por métodos de programação matemática, ditos exatos. Desta forma, o problema é normalmente abordado por técnicas heurísticas.

Nesse trabalho, o PRVFHJT foi tratado levando-se em consideração a metaheurística *Simulated Annealing* (SA), utilizando como movimentos a permutação de clientes e veículos na solução: podendo as trocas ocorrer entre clientes da mesma rota (intra-rota), entre clientes de rotas diferentes (inter-rota), entre um veículo e um cliente (diminuição/aumento de rota) e entre veículos (troca de rotas).

A função objetivo utilizada pelo método visa a eliminação de inviabilidades, tais como a sobrecarga de um veículo, o não atendimento de um cliente e outras, e ainda visa a redução dos custos operacionais, que consistem em minimizar a distância das rotas percorridas e redução dos custos de espera.

O sistema computacional desenvolvido foi implementado utilizando-se a linguagem de programação Java, com as ferramentas TextPad e Kawa.

Para simplificação do problema, a interface com o usuário não foi desenvolvida, focando assim na resolução do problema. Com o intuito de validar o sistema implementado, testou-se o sistema desenvolvido comparando seus resultados com os produzidos por um software de roteirização de mercado, o Logware.

Agradecimentos

A Deus, pela saúde e pela ajuda que mais ninguém pode dar.

À minha família, em especial à minha avó Tereza e aos meus pais Willes e Consolação pelo apoio e incentivo durante a realização do curso de Ciência da Computação.

Ao professor Marcone Jamilson Freitas Souza, pela orientação para a realização deste trabalho, e principalmente pelo companheirismo e amizade.

Aos professores Marcelo de Almeida Maia, Lucília Camarão Figueredo e Alexandre Xavier Martins, pelo apoio e confiança.

A todos os professores do DECOM, em especial ao professor Tiago Garcia de Senna Carneiro, pelos ensinamentos transmitidos.

Aos grandes amigos que fiz em Ouro Preto. Rodrigo de Oliveira Furtado, Raphael Eustáquio Vilella, André Junqueira dos Santos, Deângelo Carvalho de Oliveira e Rodrigo Geraldo Ribeiro.

Aos amigos de outras cidades, Anderson Ferreira, Rafael Ferreira, Haroldo Gomes, Danilo Gomes, Klaus Lana, Renato Silva, Mateus Laureço, Carlos Dias e Alexandre Santos, que sempre que precisei, estenderam a mão para me ajudar.

As Amigas Marina, Greyce e Tathielle.

A todos os demais, que de alguma forma contribuíram para a realização deste trabalho.

Índice

| | |
|-----------------------------------------------------------------------|-----------|
| RESUMO..... | 3 |
| AGRADECIMENTOS | 4 |
| ÍNDICE..... | 5 |
| LISTA DE FIGURAS..... | 6 |
| LISTA DE TABELAS..... | 7 |
| 1 INTRODUÇÃO | 8 |
| 1.1 ESTRUTURA DO TRABALHO..... | 8 |
| 2 REVISÃO BIBLIOGRÁFICA..... | 9 |
| 2.1 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS - PRV..... | 9 |
| 2.2 PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM FROTA HETEROGÊNEA..... | 10 |
| 2.3 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELAS DE TEMPO..... | 11 |
| 2.4 MÉTODOS UTILIZADOS | 11 |
| 2.4.1 <i>Método metaheurístico Simulated Annealing</i> | 11 |
| 2.4.2 <i>Método dos Quadrados Mínimos</i> | 13 |
| 3 METODOLOGIA..... | 15 |
| 3.1 FORMULAÇÃO MATEMÁTICA DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS | 15 |
| 4 MODELAGEM..... | 18 |
| 4.1 ESTRUTURA | 18 |
| 4.2 BASE DE DADOS..... | 19 |
| 4.3 MODELAGEM DA SOLUÇÃO..... | 20 |
| 4.4 ESTRUTURA DE VIZINHAÇA..... | 21 |
| 4.5 PENALIDADES | 24 |
| 4.6 CÁLCULO DOS CUSTOS..... | 24 |
| 5 RESULTADOS OBTIDOS..... | 25 |
| 6 CONCLUSÕES..... | 27 |
| 6.1 SUGESTÕES PARA TRABALHOS FUTUROS | 27 |
| 7 REFERÊNCIAS BIBLIOGRÁFICAS | 28 |

Lista de Figuras

| | |
|------------------------------------------------------------------------------|----|
| Figura 2.4.1.1 Algoritmo Simulated Annealing..... | 12 |
| Figura 2.4.2. 1 Distância de um ponto $(x_i; y_i)$ à reta $y = a + bx$ | 13 |
| Figura 4. 1 Arquitetura do Sistema..... | 18 |
| Figura 4.3. 1 Objetos e o que representam | 20 |
| Figura 4.3. 2 Objetos, oque representam e onde | 20 |
| Figura 4.3. 3 Um Exemplo de solução | 21 |
| Figura 4.4 1 Soluções vizinhas | 21 |
| Figura 4.4 2 Movimentação Intra-Rota | 22 |
| Figura 4.4 3 Movimentação Inter-Rota | 22 |
| Figura 4.4 4 Movimentação Diminuição/Aumento de Rota..... | 23 |
| Figura 4.4 5 Movimentação Troca de Rota | 23 |

Lista de Tabelas

| | |
|--------------------------------------------------------------------------------|----|
| Tabela 5. 1 resultados comparativos para o problema sem janelas de tempo | 25 |
| Tabela 5. 2 resultados comparativos para o problema com janelas de tempo..... | 26 |
| Tabela 5. 3 comparativo das médias dos resultados..... | 26 |

1 Introdução

O Problema de roteamento de veículos com frota heterogênea com janela de tempo (PRVFHJT) é um problema derivado da classe dos problemas de roteamento de Veículos, o qual será descrito a seguir juntamente com seus derivados.

O PRV ou problema de roteamento de veículos consiste em construir um conjunto de rotas de veículos, cada uma começando e terminando no depósito. Os veículos possuem todos a mesma capacidade, e o objetivo da resolução do problema é reduzir o custo das rotas.

O Problema de Roteamento de Veículos com Frota Heterogênea (PRVFH) tem as mesmas características do PRV, exceto que neste problema os veículos possuem capacidades diferentes [4]. Neste ponto a literatura divide o PRVFH em duas “classes” diferentes: a mais comum, na qual o conjunto de veículos possui uma quantidade ilimitada de cada tipo de veículos. Aqui, o objetivo foca-se em encontrar a melhor frota e otimizar as rotas. A segunda classe, que possui um número fixo de cada tipo de veículo, tem por objetivo otimizar o uso de uma frota (com número de veículos bem definidos) no processo de roteamento.

De fato a segunda classe na classificação de Geandreau é a que mais se aproxima da realidade uma vez que as companhias de transporte na maioria das vezes já possuem uma frota. Um uso prático da primeira classificação seria o caso de uma companhia que ainda está adquirindo sua frota ou uma companhia que esteja disposta a mudar a sua frota, o que, em ambos os casos, requer uma grande quantia de capital, o que nem sempre é fácil para empresas que estão começando suas atividades.

Para aproximar mais o problema tratado da realidade, deve-se levar em conta que os clientes não podem ser atendidos a qualquer horário. Em outras palavras não é de se esperar que uma loja esteja disposta a receber sua mercadoria fora do seu horário de funcionamento. Para isso é necessário modelar os períodos de tempo nos quais os clientes podem ser atendidos. Tal artefato é chamado de **janela de tempo** e é modelado no formato de um intervalo (início e fim), dentro do qual um cliente pode ser atendido. Cada cliente pode possuir várias janelas de tempo. O PRVFHJT trata das janelas de tempo e dos outros aspectos mencionados anteriormente e também está sujeito à divisão de Geandreau.

O problema abordado neste trabalho será um pouco mais complexo do que o PRVFHJT, pois visará realizar também o sequenciamento dos veículos. O sequenciamento visa relacionar os veículos e o tempo, levando em consideração que um veículo que terminou uma tarefa pode ser alocado para a realização de outra, tornando o uso da frota mais inteligente.

1.1 Estrutura do trabalho

O presente trabalho está dividido em seis capítulos, incluindo esta introdução.

No capítulo II faz-se uma breve revisão bibliográfica sobre as técnicas de resolução do PPT e descreve-se o método heurístico considerado neste trabalho para resolvê-lo.

O capítulo III descreve o PRVFHJT abordado, bem como o modelo heurístico de otimização desenvolvido para resolvê-lo.

No capítulo IV são mostrados detalhes do sistema computacional implementado.

O capítulo V mostra os resultados obtidos pela aplicação do sistema.

No capítulo VI o trabalho é concluído com a análise dos resultados obtidos e sugestões para trabalhos futuros.

2 Revisão Bibliográfica

Nos últimos 40 anos o problema de roteamento de veículos e seus problemas derivados tem sido estudados. Destaque é dado para alguns métodos heurísticos e metaheurísticos.

Golden [2] faz uma descrição detalhada de vários métodos heurísticos dentre os quais se pode citar o de Clark & Wright e Giant Tour.

Ocorre que nos últimos 10 anos os esforços foram voltados para o desenvolvimento das metaheurísticas, utilizando dois princípios básicos: busca local, objetivando a intensificação da busca de melhores soluções em um dado local, e busca na população global, de forma a promover a diversificação e permitir a combinação entre as soluções geradas.[7]

Gendreau, Laport e Potvin [8] descrevem e comparam uma série de métodos metaheurísticos como por exemplo, o *Taburoute* e o SA de Osman.

O problema de roteamento de veículos e algumas de suas variantes são descritas a seguir:

2.1 O Problema de Roteamento de Veículos - PRV

O Problema de Roteamento de Veículos (PRV) é um dos problemas mais estudados dentro da área de otimização combinatória (CORDEAU *et al.* 2002).

A noção de roteamento de veículos já é bastante comum no cotidiano das pessoas. Desde a distribuição de gás à coleta de lixo, da entrega de encomendas normais às de grande porte e urgência, o problema básico de definição de rotas para que os veículos de transporte possam entregar suas encomendas se faz presente.

Um problema típico de roteamento de veículos (PRV) pode ser descrito, conforme Bräysy (2001), como o problema de definição do menor custo das rotas de um depósito a um conjunto de clientes distribuídos em um plano geográfico. As rotas devem ser definidas para que cada cliente seja visitado somente uma única vez por exatamente um único veículo, começando e terminando no depósito, de forma a não exceder a capacidade do veículo. Todas as demandas dos clientes devem ser satisfeitas.

Uma revisão sobre diversos métodos de solução para o PRV, bem como a apresentação de diferentes variantes do problema é encontrada em CORDEAU *et al.* (2002). Neste trabalho os procedimentos de solução existentes na literatura são classificados em quatro categorias, a saber, precisão, velocidade, simplicidade e flexibilidade.

Uma heurística melhorada baseada no método *Simulated Annealing* é, por sua vez, apresentada em Breedam (1995), que compara os resultados a outras heurísticas da literatura. O método proposto consiste de duas fases. Na primeira fase, procura-se minimizar as distâncias e tempos de deslocamento de cada rota pela reordenação da seqüência dos clientes a serem atendidos. Na segunda fase procura-se realocar os clientes entre duas rotas de uma solução viável.

Diversas são as variações do PRV básico, na maioria das vezes apenas compostos por um maior número de restrições, seja de tempo, de distância, de composição de frota, de limitação de frota etc. As variantes do PRV mais comuns são aquelas nas quais a frota é heterogênea e aquelas nas quais existem restrições com respeito a janelas de tempo.

2.2 Problema de Roteamento de Veículos com Frota Heterogênea

Este problema consiste no PRV básico com a restrição de que a frota é composta por tipos diferentes de veículos de transporte. O que diferencia um tipo de veículo de outro é a sua capacidade de transporte. A frota de veículos de cada tipo é normalmente limitada.

Em Gendreau *et al.* (1999), tal problema foi resolvido utilizando o método de Busca Tabu. Os autores adaptaram a heurística denominada GENIUS (GENI: *Generalized Insertion* e US: *Unstringing and Stringing*), originalmente desenvolvida por Gendreau *et al.* (1992), conforme Gendreau *et al.* (1999), para resolver o Problema do Caixeiro Viajante, para a construção da solução inicial do PRV e uma heurística baseada no método das economias, aliado a um método de memória adaptativa, para refinar a solução construída.

Uma outra abordagem para o problema, apresentado por (Renaud & Boctor, 2002), utiliza um método baseado na heurística de varredura (*Sweep Based Algorithm*), auxiliado por cinco outros procedimentos denominados: *Order* (ordena os clientes em ordem crescente de distância a partir do vértice de referência), *1-petal* (seleciona uma rota, minimiza a distância desta e aloca um veículo), *2-petal* (seleciona 2 rotas, minimiza suas distâncias e aloca 2 veículos às mesmas), *petals selection* (seleciona um conjunto composto por uma configuração *1-petal* e uma configuração *2-petal*, combinando os clientes) e *improve* (procedimento de melhora das rotas). Os resultados apresentados são comparados a outros da literatura.

Uma abordagem para o PRV com frota heterogênea e janela de tempo é apresentada por LIU *et al.* (1999). Os autores utilizaram uma versão modificada do método das economias para inserção dos clientes nas rotas, onde um fator N (fator de decremento) determina a variação dos resultados através da sua multiplicação pelo número consecutivo de rotas construídas com pelo menos dois clientes. Utiliza-se desse recurso pelo fato de que, segundo os autores, muitas construções consecutivas compostas por dois ou mais clientes resultam numa solução de péssima qualidade no atendimento a restrições de janelas de tempo.

O PRV com restrições de janela de tempo, como o apresentado, é uma das variações mais implementadas na literatura e é apresentada a seguir.

2.3 O Problema de Roteamento de Veículos com Janelas de Tempo

Uma das variações do PRV mais implementadas, o PRV com janelas de tempo, já possui diversas contribuições na literatura.

Também conhecido como Problema de Roteamento de Veículos com Janela de Tempo (PRVJT), ou na literatura inglesa como VRPTW (*Vehicle Routing Problem with Time Windows*), o problema consiste na adição de restrições de intervalo de tempo de atendimento ao problema básico do PRV.

Mais precisamente, dado um tempo de serviço s_i , cada janela de tempo é definida como $[a_i, b_i]$, onde a_i é o horário mais cedo e b_i o horário mais tarde que se pode começar o atendimento, (ALVARENGA *et al.*, 2002). Caso o veículo chegue ao cliente antes do horário definido por a_i , este deve esperar. Caso chegue após b_i , este não poderá ser atendido.

Tal problema foi tratado por Bräysy (2001, 2002), Alvarenga *et al.* (2002), Chaovalitwongse *et al.* (2002), Laporte (1992) e Liu & Shen (1999), apresentando apenas pequenas diferenças entre as abordagens.

Os métodos utilizados na resolução do problema de roteamento de veículos com janela de tempo incluem Algoritmos Genéticos (BRÄYSY, 2001), Método Reativo de Busca em Vizinhaça Variável (BRÄYSY, 2002), Algoritmo Genético Híbrido, combinando o método básico com um algoritmo de melhora (ALVARENGA *et al.*, 2002), GRASP combinado com um novo método de busca local (CHAOVALITWONGSE *et al.*, 2002) e um método melhorado de construção de rotas (LIU & SHEN, 1999).

2.4 Métodos utilizados

2.4.1 Método metaheurístico *Simulated Annealing*

Simulated Annealing é uma classe de metaheurística proposta originalmente por Kirkpatrick *et al.* (1983), sendo uma técnica de busca local probabilística, que se fundamenta em uma analogia com a termodinâmica, ao simular o resfriamento de um conjunto de átomos aquecidos, operação conhecida como recozimento. Esta técnica começa sua busca a partir de uma solução inicial qualquer. O procedimento principal consiste em um *loop* que gera aleatoriamente, em cada iteração, um vizinho s' da solução corrente s .

A cada geração de um vizinho s' de s , é testada a variação Δ do valor da função objetivo, isto é, $\Delta = f(s') - f(s)$. Se $\Delta < 0$, o método aceita a solução e s' passa a ser a nova solução corrente. Caso $\Delta \geq 0$ a solução vizinha candidata também poderá ser aceita, mas neste caso, com uma probabilidade $e^{-\Delta/T}$, onde T é um parâmetro do método, chamado de temperatura e que regula a probabilidade de aceitação de soluções com custo pior.

A temperatura T assume, inicialmente, um valor elevado T_0 . Após um número fixo de iterações (o qual representa o número de iterações necessárias para o sistema atingir o equilíbrio térmico em uma dada temperatura), a temperatura é gradativamente diminuída por

2.4.2 Método dos Quadrados Mínimos

Muitos dos problemas abordados na literatura relativos a roteamento de veículos trabalham com coordenadas cartesianas, o que implica que em muitos casos as distâncias utilizadas no cálculo dos custos são distâncias euclidianas. Isto é um fator que empobrece o problema no que diz respeito à realidade, afinal, as distâncias euclidianas, que são a distância em linha reta de um ponto ao outro, tem uma probabilidade mínima de serem as distâncias reais entre estes mesmos pontos, uma vez que não são considerados as curvas das estradas e nem os desníveis de altitude, que são variáveis presentes no mundo real.

Usar as distâncias reais como parâmetro para o problema é a solução ideal, mas nem sempre possível. Quando o numero de clientes torna-se grande, o conjunto de distâncias de entrada torna-se muito grande uma vez que a distância entre todos os vértices (Clientes e depósito) e necessária. Supondo que existam “n” vértices, o número de arestas necessárias é “ n^2 ”, o que torna a entrada de dados quando o numero de vértices se torna relativamente grande pois, o volume de entrada torna-se muito grande e nem sempre é possível saber a distância real entre todos os vértices.

O que se torna interessante é utilizar mecanismos que utilizem os dois conceitos, utilizando as distâncias reais conhecidas, e quando esta distância não for conhecida, usar as distâncias euclidianas.

Um método interessante neste aspecto é o método dos Quadrados Mínimos. Este método utiliza um conjunto de distâncias reais para gerar dados estatísticos, estes dados por sua vez são utilizados, juntamente com as distâncias euclidianas, para calcular uma estimativa das distâncias reais entre os vértices. Tornando assim, o calculo das distâncias entre os pares de vértices que não foram mencionados no conjunto de distâncias reais de entrada, mais adequado ao modelo real do problema.

[5] Mostremos, inicialmente, como ajustar um conjunto de pontos a uma reta $y = a + bx$, onde a e b são parâmetros a serem determinados.

Neste caso, estamos interessados em minimizar a distância de cada ponto $(x_i; y_i)$ da tabela à cada ponto $(x_i; a + bx_i)$ da reta, conforme ilustra a figura 1.

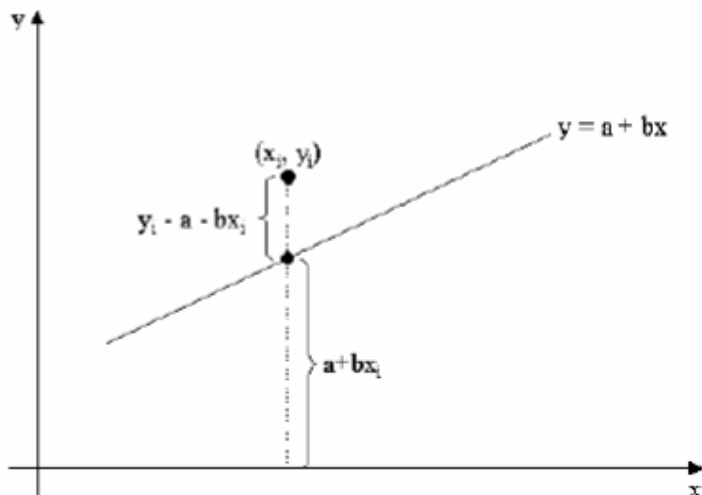


Figura 2.4.2. 1 Distância de um ponto $(x_i; y_i)$ à reta $y = a + bx$

A distância entre esses pontos é $|y_i - a - bx_i|$ e a soma dos quadrados dessas distâncias é:

$$q = \sum_{i=1}^n (y_i - a - bx_i)^2 \quad (2.1)$$

Os candidatos a ponto de mínimo da função 2.1 são aqueles para os quais são nulos as derivadas parciais de q em relação a cada um de seus parâmetros, isto é:

$$\frac{\partial q}{\partial a} = -2 \sum_{i=1}^n (y_i - a - bx_i) = 0 \quad (2.2)$$

$$\frac{\partial q}{\partial b} = -2 \sum_{i=1}^n x_i (y_i - a - bx_i) = 0 \quad (2.3)$$

Tendo em vista que:

$$\begin{aligned} \sum_{i=1}^n (y_i - a - bx_i) &= \sum_{i=1}^n y_i - \sum_{i=1}^n a - \sum_{i=1}^n bx_i \\ &= \sum_{i=1}^n y_i - na - \left(\sum_{i=1}^n x_i \right) b \end{aligned}$$

e que:

$$\sum_{i=1}^n x_i (y_i - a - bx_i) = \sum_{i=1}^n x_i y_i - \left(\sum_{i=1}^n x_i \right) a - \left(\sum_{i=1}^n x_i^2 \right) b$$

Obtemos o seguinte conjunto de equações denominado “equações normais” do problema, cujas incógnitas são os parâmetros “a” e “b” da equação $y = bx + a$.

$$\begin{cases} na + \left(\sum_{i=1}^n x_i \right) b = \sum_{i=1}^n y_i \\ \left(\sum_{i=1}^n x_i \right) a + \left(\sum_{i=1}^n x_i^2 \right) b = \sum_{i=1}^n x_i y_i \end{cases}$$

Tornando “ x_i ” a distância euclidiana e “ y_i ” como a distância real de um dado subconjunto de pontos, este método pode ser aplicado para estimar as distâncias reais de todo o conjunto de pontos de problemas de roteamento. Para isso, basta aplicar a fórmula $y_i = bx_i + a$ para estimar todas as distâncias, inclusive aquelas para as quais se conhece a distância real. Na literatura [Novaes 2002] é comum estimar a distância real através da fórmula: $d_{\text{real}} = k \cdot d_{\text{euclidiana}}$, sendo k um fator de correção que depende da região sob análise. Para obter esse fator k basta dividir a média das distâncias estimadas (pela aplicação do método dos quadrados mínimos) pela média das distâncias euclidianas.

3 Metodologia

3.1 Formulação matemática do Problema de Roteamento de Veículos

Deixe $G = (V, A)$ ser um grafo direto onde $V = \{v_0, v_1, \dots, v_n\}$ é o conjunto de vértices e $A = \{(v_i, v_j): i \neq j\}$ é o conjunto de arcos. O vértice v_0 representa um depósito onde está um conjunto de veículos de diferentes capacidades, enquanto os vértices restantes correspondem às cidades ou consumidores. Cada consumidor v_i tem uma demanda não-negativa q_i e $q_0 = 0$. A cada arco (v_i, v_j) está associada uma distância não-negativa c_{ij} que representa a distância entre os consumidores (nós).

O PRV consiste em determinar o conjunto de rotas que devem ser seguidas pelos veículos minimizando os custos de transporte dado pelas distâncias obedecendo às seguintes restrições:

- Cada rota começa e termina no depósito;
- Cada cidade de $V \setminus \{v_0\}$ é visitada somente uma única vez por somente um veículo;
- A demanda total de cada rota não pode exceder a capacidade do caminhão;
- O tempo total gasto para percorrer a rota não pode ser maior que um limite LT dado como parâmetro;
- A distância total percorrida em uma rota não pode ser maior que um limite LD dado como parâmetro;

Formulação Matemática[10]

Variáveis principais

t_i tempo de chegada no vértice i
 w_i tempo de espera no vértice i
 $x_{ijk} \in \{0,1\}$ 1 se existe um arco dos vértices i para j utilizando um veículo k , 0 caso contrário $i \neq j; i, j \in \{0,1,2,\dots,N\}$

Parâmetros:

K número total de veículos
 N número total de clientes
 c_{ij} custo de viagem do vértice i para o vértice j
 t_{ij} tempo de viagem entre os vértices i e j
 m_i demanda do vértice i
 q_k capacidade do veículo k
 e_i tempo de chegada mais cedo no vértice i
 l_i tempo de chegada mais tarde no vértice i
 f_i tempo de serviço no vértice i
 r_k tempo máximo permitido para a rota do veículo k

$$\text{Minimizar: } \sum_{i=0}^N \sum_{j=0, j \neq i}^N \sum_{k=1}^K c_{ij} x_{ijk} \quad (1)$$

sujeito a:

$$\sum_{k=1}^K \sum_{j=1}^N x_{ijk} \leq K \quad \text{para } i=0 \quad (2)$$

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

$$\sum_{k=1}^K \sum_{j=0, j \neq i}^N x_{ijk} = 1 \quad \text{para } i \in \{1, \dots, N\} \quad (4)$$

$$\sum_{k=1}^K \sum_{i=0, i \neq j}^N x_{ijk} = 1 \quad \text{para } j \in \{1, \dots, N\} \quad (5)$$

$$\sum_{i=0}^N m_i \sum_{j=0, j \neq i}^N x_{ijk} \leq q_k \quad \text{para } k \in \{1, \dots, K\} \quad (6)$$

$$\sum_{i=0}^N \sum_{j=0, j \neq i}^N x_{ijk} (t_{ij} + f_i + w_i) \leq r_k \quad \text{para } k \in \{1, \dots, K\} \quad (7)$$

$$t_0 = w_0 = f_0 = 0 \quad (8)$$

$$\sum_{k=1}^K \sum_{i=0, i \neq j}^N x_{ijk} (t_i + t_{ij} + f_i + w_i) \leq t_j \quad \text{para } j \in \{1, \dots, N\} \quad (9)$$

$$e_i \leq (t_i + w_i) \leq I_i \quad \text{para } i \in \{1, \dots, N\} \quad (10)$$

A fórmula (1) é a função objetivo do problema. (2) especifica que existem no máximo K rotas saindo do depósito. A equação (3) certifica que todas as rotas começam e terminam no depósito central. As equações (4) e (5) definem que cada vértice de cliente pode ser visitado apenas uma vez por um veículo. A equação (6) é a restrição de capacidade. A equação (7) é a restrição do tempo máximo de viagem. As restrições de (8) a (10) definem a janela de tempo.

(4)

4 Modelagem

4.1 Estrutura

O Projeto está dividido em uma série de classes as quais serão mostradas a seguir juntamente com suas funcionalidades.

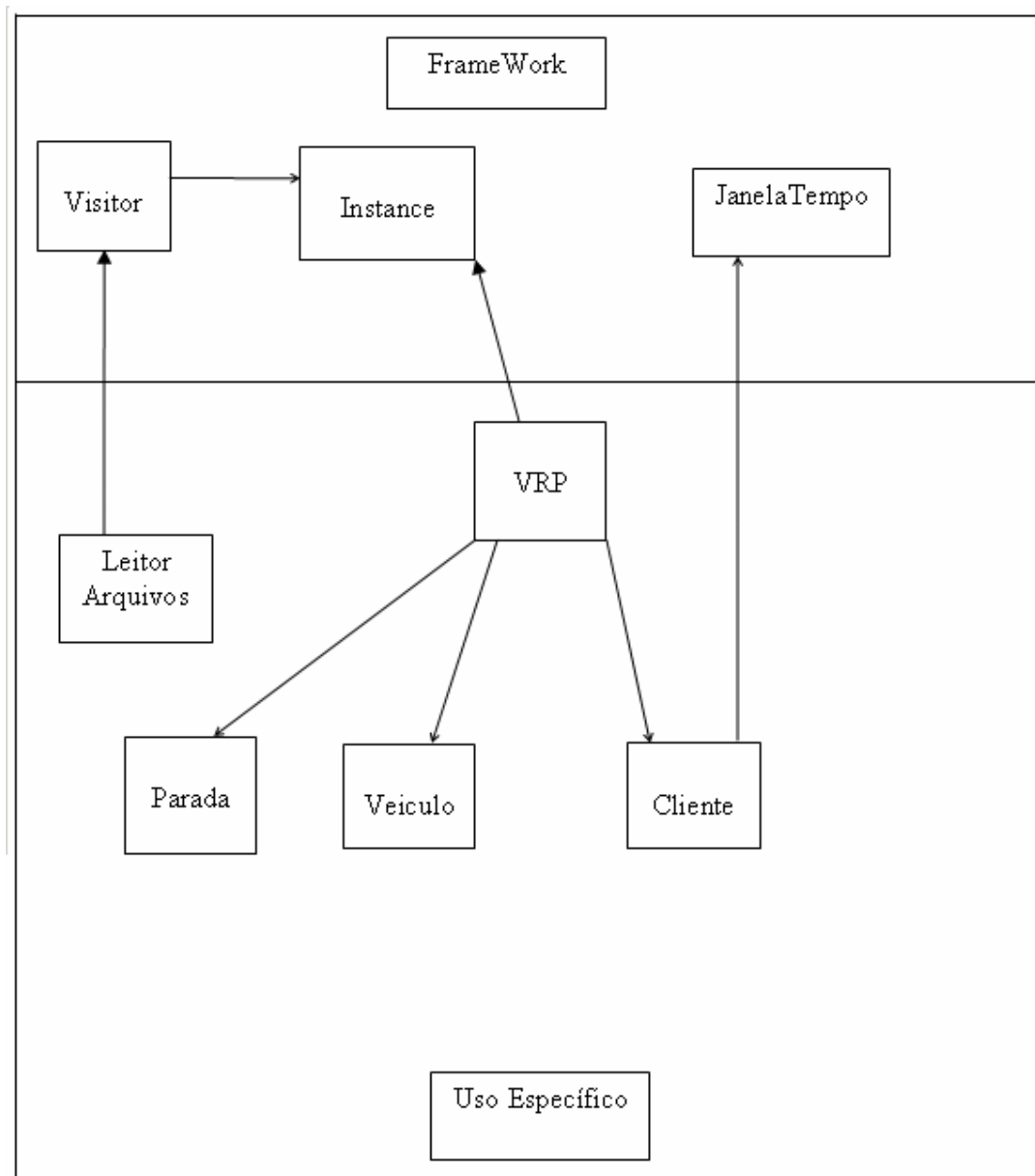


Figura 4.1 Arquitetura do Sistema

Já há algum tempo vem se realizando, pelo autor, a construção de um framework para tratar deste tipo de problemas. A construção deste framework ainda está incompleta, mas isso não impede o uso de algumas de suas partes como foi feito neste trabalho.

As classes deste framework que foram usadas, como mostrado na figura anterior são *Visitor*, *Instance* e *JanelaTempo*. As classes *Visitor* e *Instance* foram usadas de acordo com o padrão de projeto Visitor, este padrão foi usado para inserir os dados lidos dos arquivos na classe a qual deve utilizá-los para o processo de otimização.

A classe *JanelaTempo* é utilizada para modelar os intervalos em que os clientes podem ser atendidos.

As classes de domínio específico são as da parte inferior da figura. A classe *LeitorArquivos* é responsável por ler os dados que estão nos arquivos, transformá-los para o formato desejado e enviá-los para a classe VRP, onde ficarão armazenados para uso futuro.

A classe *Parada* é uma classe auxiliar que é usada pela classe VRP. Sua função é armazenar os dados referentes às paradas feitas pelos motoristas além dos atendimentos (Almoço e jantar, por exemplo).

A classe *Cliente* é responsável por armazenar os dados de um cliente como nome, demanda, janelas de tempo, etc.

A classe *Veiculo* é responsável por guardar as informações de um veículo como rótulo, capacidade, quantidade de carga atual, etc.

A classe VRP é a classe principal do sistema. Nela estão armazenados dados importantes para o processo de otimização como por exemplo, a matriz que contém as distâncias entre cada par de clientes. Ela também é responsável pelo processo de otimização e contém o algoritmo *Simulated Annealing* (SA).

4.2 Base de dados

Os dados para o problema são divididos em uma série matrizes, vetores, e valores que serão descritos adiante.

Os dados relativos aos clientes encontram-se em um vetor de clientes, o qual pode ser referenciado para consulta a qualquer momento, no entanto não sendo modificado durante a resolução do problema. De tal vetor podem ser extraídas informações tais como numero de clientes, as demandas, tempo de atendimento e as janelas de tempo de cada cliente.

Os dados relativos aos veículos se encontram em um vetor de veículos, o qual pode ser referenciado para consulta a qualquer momento da resolução do problema e para modificação durante um período na resolução do problema. Deste vetor podem ser extraídas informações tais como a quantidade de veículos, a capacidade, o custo operacional, o custo quilométrico, e quantidade de mercadoria no veículo em um dado instante para cada um dos veículos.

Os dados relativos às paradas que o motorista pode fazer durante o percurso da rota são armazenadas em um vetor de paradas, o qual contém informações tais como, numero de paradas, tempo de início e duração de cada uma das paradas.

Há duas matrizes na modelagem, uma que contém todas as distâncias entre os pares de clientes, e uma matriz com o tempo de viagem entre os pares de clientes. Ambas as matrizes são preenchidas durante um pré-processamento onde o método dos quadrados mínimos é utilizado para calcular a distância entre os clientes, a partir destas distâncias e da velocidade média, que é um parâmetro de entrada, é preenchida a matriz de tempos.

Alguns dados adicionais são necessários para a resolução do problema, estes dados são os limites de tempo e distância de cada rota, e o limite de tempo das horas extras. Tais dados são armazenados em variáveis.

4.3 Modelagem da solução

A modelagem utilizada para a resolução do problema mostra-se adequada, do ponto de vista da generalidade e da expansividade, pois, trata de maneira conveniente o problema e permite que no futuro, novos elementos sejam incorporados à mesma. Este atributo de um algoritmo, classificado em Cordeau et al. (2002)[9] como flexibilidade, é um fator positivo para um algoritmo.

A solução internamente é representada por um vetor de objetos de um tipo simples chamados “Representações”, que servem para criar uma indireção entre a modelagem da solução e a modelagem do restante dos dados.

Um objeto do tipo Representação contém, neste caso específico, dois dados; um significa o tipo de instância que este objeto representa. No nosso caso 0 (zero) para veículos e 1 (um) para clientes.

Como a solução é um vetor, assim ficaria um exemplo com um veículo e 3 clientes

| | | | | |
|------------------|----|----|----|----|
| índices | 0 | 1 | 2 | 3 |
| | R1 | R2 | R3 | R4 |
| O que representa | 1 | 1 | 0 | 1 |

Figura 4.3. 1 Objetos e o que representam

Um objeto do tipo representação também possui um índice que aponta para qual objeto ele representa. Este índice indica onde na base de dados se encontra o objeto o qual este objeto representa.

| | | | | |
|------------------|----|----|----|----|
| índices | 0 | 1 | 2 | 3 |
| | R1 | R2 | R3 | R4 |
| O que representa | 1 | 1 | 0 | 1 |
| Onde se encontra | 2 | 0 | 0 | 1 |

Figura 4.3. 2 Objetos, o que representam e onde

O esquema acima mostra que o objeto R1 representa um cliente por conter o valor 1 no atributo “oQueRepresenta”, e que este cliente se encontra na posição 2 da base de dados. Por sua vez, o objeto R3 representa um veículo por conter zero no atributo “oQueRepresenta”, sendo que este veículo se encontra na posição zero na base de dados.

A posição de cada representação também tem significado. Se um cliente será atendido pelo veículo mais próximo com índice menor que o dele. Em outras palavras, um veículo atenderá todos os clientes que se interpõem entre ele e o próximo veículo na solução. No exemplo anterior, seguindo o que foi dito, o cliente representado por R4 seria atendido pelo veículo representado por R3. Mais um exemplo

| | | | | | |
|------------------|----|----|----|----|----|
| índices | 0 | 1 | 2 | 3 | 4 |
| | R1 | R2 | R3 | R4 | R5 |
| ○ que representa | 0 | 1 | 1 | 0 | 1 |
| Onde se encontra | 1 | 2 | 0 | 0 | 1 |

Figura 4.3. 3 Um Exemplo de solução

No exemplo acima, os clientes representados por R2 e R3 são atendidos pelo veículo representado por R1 enquanto, o cliente representado por R5 é atendido pelo veículo representado por R4.

4.4 Estrutura de vizinhança

A estrutura de vizinhança para o modelo adotado é bem simples, qualquer permutação de duas posições na solução é considerada um vizinho. Exemplo:

| | | | | | |
|---------|----|----|----|----|----|
| índices | 0 | 1 | 2 | 3 | 4 |
| | R1 | R2 | R3 | R4 | R5 |
| índices | 0 | 1 | 2 | 3 | 4 |
| | R1 | R3 | R2 | R4 | R5 |

Figura 4.4 1 Soluções vizinhas

Note que a diferença entre as duas soluções é a troca de posições entre os objetos R2 e R3. Assim modelada, a vizinhança abrange implicitamente 4 tipos diferentes de variação da solução:

Intra-Rota – quando dois clientes da mesma rota são permutados, mudando a ordem de visita dos mesmos.

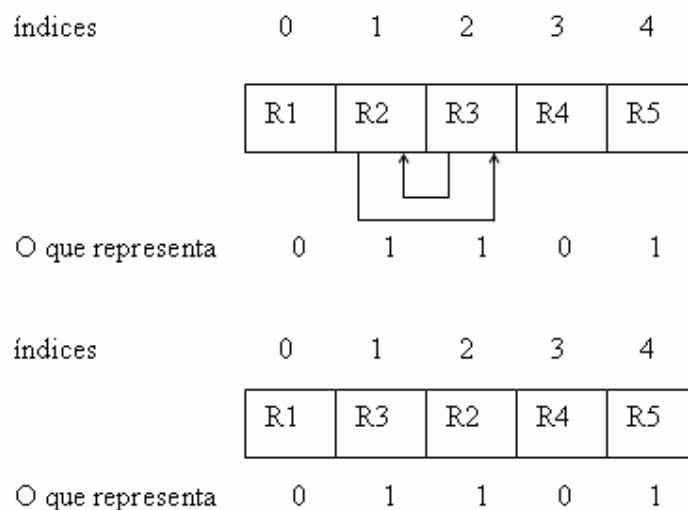


Figura 4.4 2 Movimentação Intra-Rota

Inter-Rota – quando dois clientes de duas rotas diferentes são permutados, mudando assim, duas rotas.

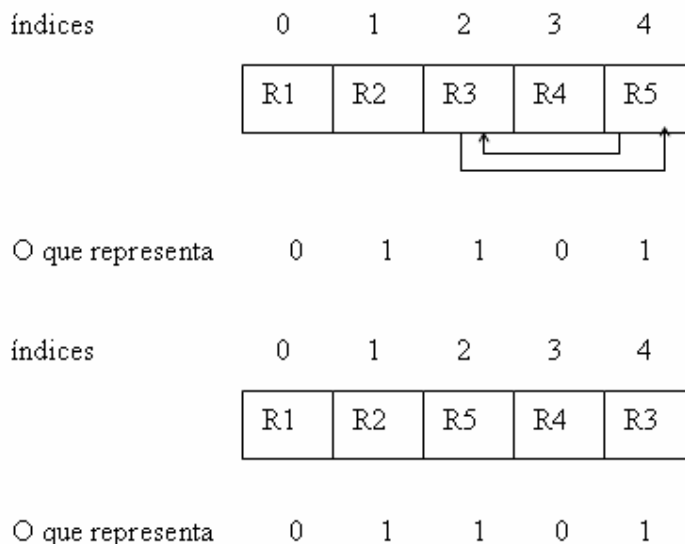


Figura 4.4 3 Movimentação Inter-Rota

Diminuição/Aumento de Rota – quando um cliente e um veículo são permutados, fazendo com que uma rota diminua e outra aumente.

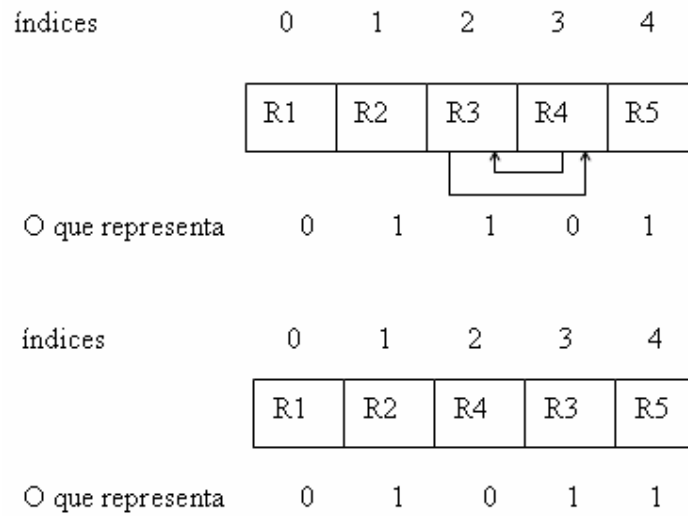


Figura 4.4 4 Movimentação Diminuição/Aumento de Rota

Troca de rota – quando dois veículos são permutados, fazendo que suas rotas anteriores sejam percorridas por outro veículo, possivelmente de capacidade e custos diferentes.

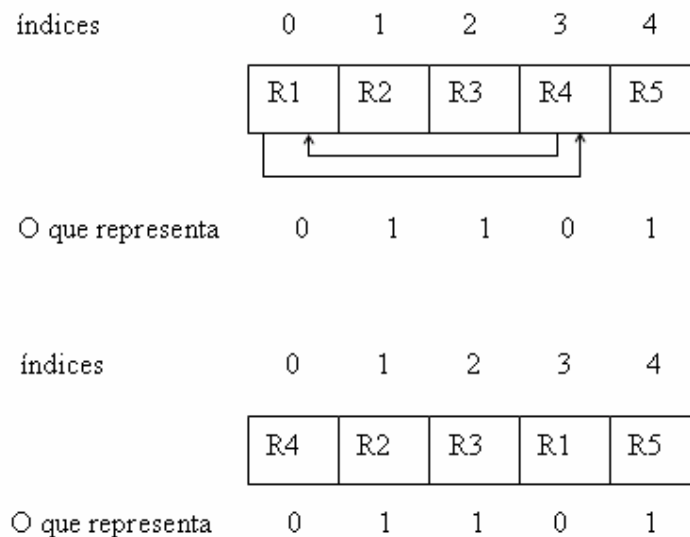


Figura 4.4 5 Movimentação Troca de Rota

4.5 Penalidades

Foram estabelecidas três penalidades diferentes: uma para o caso da rota superar o limite de distância pré-estabelecido, uma para o caso da rota superar o limite de tempo também pré-estabelecido e uma terceira para o caso de demanda da rota superar, em algum momento, a capacidade do veículo.

4.6 Cálculo dos custos

A partir do cálculo da distância para as rotas, é calculado o custo para percorrer as mesmas, em seguida, calcula-se os tempos das mesmas levando em conta o tempo de descarga, o tempo fixo de atendimento e a presença das janelas de tempo. Caso um veículo chegue a um cliente durante um intervalo entre janelas de tempo o veículo deverá permanecer naquele cliente até que a próxima janela comece, atendendo, assim este cliente. As paradas são feitas à medida em que seu tempo de início é atingido, de modo que sua execução não inviabilize a realização da rota. Por exemplo, uma rota já se encontra no tempo 50, o uma parada deveria ser realizada a partir do tempo 45 e esta parada levaria 40 minutos, suponha que a janela de tempo do cliente no qual o veículo se encontra agora, fecharia no tempo 60, se a parada fosse realizada antes do atendimento, o veículo teria que esperar até a próxima janela de tempo, deste modo, o atendimento deve ser feito primeiro. Logo após o atendimento, a parada será realizada.

Após calculados os custos, e estabelecidas as inviabilidades com relação aos limites, os custos totais são calculados por uma soma dos custos com uma multiplicação das inviabilidades por suas penalidades:

$$\text{Custo penalizado} = \sum_{i=1}^3 \text{inviabilidade}_i \times \text{penalidade}_i$$

$$\text{Custos totais} = \sum_{i=1}^3 \text{custos}_i$$

$$\text{FO} = \text{custos totais} + \text{custo penalizado}$$

5 Resultados obtidos

O método proposto foi implementado em Java e testado em um microcomputador Pentium IV 2,8 GHz HT, com 512 MB de RAM. A versão preliminar do método desenvolvido foi comparada com um aplicativo de roteirização disponível no mercado, chamado Logware. Este software possui dois arquivos exemplo, um com frota heterogênea e um com janelas de tempo. O LogWare é um software comercial que usa uma heurística convencional, provavelmente da década de 60 e, portanto, chega sempre ao mesmo resultado. Isto não acontece com o método proposto neste trabalho, o qual, por não ser determinístico, pode chegar a soluções finais diferentes.

A Tabela 1 mostra 20 execuções do método proposto para a instância que não considera janelas de tempo.

| Execução | MP - custo | MP – Tempo (ms) | LogWare - custo | LogWare – tempo (ms) |
|----------|------------|-----------------|-----------------|----------------------|
| 1 | 7844* | 3469 | 8489 | 996 |
| 2 | 8505 | 2962 | 8489* | 996 |
| 3 | 7844* | 3329 | 8489 | 996 |
| 4 | 8031* | 3507 | 8489 | 996 |
| 5 | 8186* | 3094 | 8489 | 996 |
| 6 | 7944* | 3412 | 8489 | 996 |
| 7 | 8011* | 3361 | 8489 | 996 |
| 8 | 8055* | 3246 | 8489 | 996 |
| 9 | 8456* | 3671 | 8489 | 996 |
| 10 | 7844* | 3430 | 8489 | 996 |
| 11 | 7976* | 2971 | 8489 | 996 |
| 12 | 8011* | 3352 | 8489 | 996 |
| 13 | 8037* | 3403 | 8489 | 996 |
| 14 | 7844* | 3458 | 8489 | 996 |
| 15 | 8040* | 3507 | 8489 | 996 |
| 16 | 7965* | 3474 | 8489 | 996 |
| 17 | 8031* | 3562 | 8489 | 996 |
| 18 | 8011* | 3295 | 8489 | 996 |
| 19 | 7844* | 3456 | 8489 | 996 |
| 20 | 7944* | 3521 | 8489 | 996 |

Tabela 5. 1 resultados comparativos para o problema sem janelas de tempo

Observa-se que, em relação à qualidade das soluções, o método proposto em 95% dos casos produz uma solução melhor que a do Logware para o problema exemplo sem janelas de tempo.

A Tabela 2 mostra 10 execuções do método proposto para a instância que considera janelas de tempo.

| Execução | MP - custo | MP – Tempo (ms) | LogWare - custo | LogWare – tempo (ms) |
|----------|------------|-----------------|-----------------|----------------------|
| 1 | 4223* | 5546 | 5095 | 996 |
| 2 | 4417* | 5609 | 5095 | 996 |
| 3 | 4235* | 5485 | 5095 | 996 |
| 4 | 4263* | 5484 | 5095 | 996 |
| 5 | 4533* | 5516 | 5095 | 996 |
| 6 | 4223* | 5406 | 5095 | 996 |
| 7 | 4147* | 5375 | 5095 | 996 |
| 8 | 4293* | 5516 | 5095 | 996 |
| 9 | 4223* | 5547 | 5095 | 996 |
| 10 | 4333* | 5562 | 5095 | 996 |

Tabela 5. 2 resultados comparativos para o problema com janelas de tempo

Observa-se que, em relação à qualidade das soluções, o método proposto em 100% dos casos produz uma solução melhor que a do Logware para o problema exemplo com janelas de tempo.

| Instancia do Problema | Custo Médio MP | Custo médio Logware | % Melhora | Tempo Médio MP | Tempo médio Logware |
|-----------------------|----------------|---------------------|-----------|----------------|---------------------|
| Sem JT | 8021 | 8489 | 5,5 | 3374 | 996 |
| Com JT | 4289 | 5095 | 15,9 | 5504 | 996 |

Tabela 5. 3 comparativo das médias dos resultados

Obs :1 – o LogWare não possui mecanismo para medir o tempo de execução, então foi utilizado um relógio cronômetro para fazer a medição.

6 Conclusões

Através de um método metaheurístico foi possível resolver um dos problemas mais complexos da literatura. Pode-se confirmar através deste estudo, a afirmação de Cordeau[9], que diz que os softwares de mercado são baseados em metodologias ultrapassadas.

A parte do problema que trata do sequenciamento dos veículos para melhor utilização da frota é também de natureza complexa.. Pelo fato de se tratar de um outro problema de natureza NP-Difícil, decidiu-se cuidar dele em um trabalho posterior, usando os resultados produzidos pelo método proposto neste trabalho como entrada para sua resolução.

6.1 Sugestões para trabalhos futuros

- Refinar o método proposto
- Terminar a implementação do Framework
- Fazer novos testes do uso de engenharia de software em otimização.

7 Referências Bibliográficas

- [1] COSTA Jr., Aloísio Carlos T., O Problema de Roteamento Periódico de Veículos: Uma abordagem via Metaheurística GRASP, Dissertação de mestrado em Ciência da Computação, Universidade Federal Fluminense (UFF), Niterói, Rio de Janeiro (2003).
- [2] GOLDEN, B.; ASSAD, A.; LEVY, L. & GHEYSENS, F., The fleet size and mix vehicle routing problem, *Computers & Operations Research*, vol. 11, p. 49-66 (1984).
- [3] RENAUD, Jacques & BOCTOR, F., Discrete Optimization, A sweep-based algorithm for the fleet size and mix vehicle routing problem, *European Journal of Operational Research*, vol. 140, p. 618-628 (2002).
- [4] GENDREAU, Michel; LAPORTE, Gilbert; MUSARAGANYI, Christophe & TAILLARD, EDRIC D., A tabu search heuristic for the heterogeneous fleet vehicle routing problem, *Computers & Operations Research*, vol. 26, p. 1153-1173 (1999).
- [5] SOUZA, Marcone Jamilson Freitas, Notas de aula para métodos numéricos, Departamento de computação/iceb/ufop. <http://www.decom.ufop.br/prof/marcone/Disciplinas/MetodosNumericoseEstatisticos/QuadradosMinimos.pdf>
- [6] NOVAES, Antônio G. A tabu search heuristic for the heterogeneous fleet vehicle routing problem, *Computers & Operations Research*, vol. 26, p. 1153-1173 (2002).
- [7] MINE, Otávio Massashi. Abordagem Metaheurística para o Problema de Roteamento Periódico de Veículos Relatório Técnico Final DECOM(2003)
- [8] GENDREAU, Michel; LAPORTE, Gilbert; POTVIN, Jean-Yves. Metaheuristics for the Vehicle Routing Problem, Centre de Recherche sur les transports – publication #963 (1994)
- [9] CORDEAU, JF; GENDREAU, Michel; LAPORTE, Gilbert; POTVIN, Jean-Yves; SEMET, F. A guide to vehicle routing heuristics. *Journal of the Operational Research Society* vol: 53, p.512-522 (2002)
- [10] TAN, K.C.; LEE, L.H.; ZHU, Q.L.; OU, K. ; Heuristic Methods for Vehicle Routing Problem with Time Windows. *Artificial intelligence in Engineering* vol. 15 p. 281-295 (2001)