

## **SELEÇÃO DE OBJETIVOS NO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELAS DE TEMPO**

**Lucas Carvalho Oliveira Matsueda**

Departamento de Computação - Universidade Federal de Ouro Preto  
Campus Universitário, Morro do Cruzeiro, CEP 35.400-000, Ouro Preto - MG, Brasil  
lucasmatsueda@gmail.com

**Alan Robert Resende de Freitas**

Departamento de Computação - Universidade Federal de Ouro Preto  
Campus Universitário, Morro do Cruzeiro, CEP 35.400-000, Ouro Preto - MG, Brasil  
alandefreitas@gmail.com

**Frederico Gadelha Guimarães**

Departamento de Engenharia Elétrica – Universidade Federal de Minas Gerais  
Av. Antônio Carlos 6627, CEP 31.270-901, Belo Horizonte – MG, Brasil  
fredericoguimaraes@ufmg.br

### **RESUMO**

A maior parte dos problemas práticos encontrados na área de otimização envolve a obtenção de diversas metas que devem ser atingidas simultaneamente. Estes problemas são chamados de problemas de otimização multiobjetivo. Visto que a maioria dos objetivos colocados em questão são conflitantes, esse tipo de problema geralmente não possui uma única solução que otimize todas suas metas. Diante deste contexto, este trabalho estuda os principais objetivos tratados na literatura para o Problema de Roteamento de Veículos com Janelas de Tempo e o conflito existente entre eles. Uma ferramenta denominada de Árvores de Agregação foi utilizada para verificar a relação entre os objetivos. Mediante os resultados, um algoritmo multiobjetivo (NSGA-II) foi utilizado para resolver o problema com aqueles objetivos que apresentaram maior conflito entre si. Os resultados retornados comprovam o conflito indicado pela Árvore de Agregação, gerando diversas soluções conflitantes que formaram os Paretos para as soluções testadas.

**PALAVRAS CHAVE: Problema de Roteamento de Veículos com Janelas de Tempo. Otimização multiobjetivo. Conflito. Árvores de Agregação.**

### **ABSTRACT**

Most real-world problems found in field of optimization involve obtaining various goals to be achieved simultaneously. These problems are called multiobjective optimization problems. Since most goals conflicting, this kind of problem usually does not have a single solution that optimizes all goals. In this context, this paper presents the main objectives addressed in the literature for the Vehicle Routing Problem with Time Windows and the conflict between them. A tool called aggregation tree was used to verify the relationship between objectives. From the results, a multiobjective algorithm (NSGA-II) was used to solve the problem with the most conflicting objectives. The results prove the conflict indicated by aggregation tree, spawning several conflicting solutions that formed the Pareto solutions tested.

**KEYWORDS: Vehicle Routing Problem with Time Windows. Multiobjective optimization. Conflict. Aggregation tree.**

## 1. Introdução

Com a competitividade do mercado atual, o correto gerenciamento dos recursos e investimentos deixou de ser um diferencial entre as organizações para se tornarem um fator de sobrevivência. Assim, a busca pela otimização dos processos, auxiliados por sistemas de informação, tem se tornado cada vez mais corrente em diversas áreas de pesquisa. A logística é uma das áreas que mais sentiu a necessidade de aprimorar seus métodos. Devido ao constante desenvolvimento do mercado global, diversas exigências, relacionadas à logística e ao transporte, surgem a todo instante. Redução de custos, programação das entregas, redução e cumprimento dos prazos de entrega e disponibilidade constante dos produtos, são alguns exemplos de problemas logísticos.

Um dos principais problemas de ampla aplicação prática na logística é o Problema de Roteamento de Veículos (PRV) (Vehicle Routing Problem – VRP), pois abrange desde a produção de mercadorias até a entrega aos clientes. Como exemplo, os trabalhos de Brown e Graves (1981), Fisher et al. (1982), Bell et al. (1983), Evans e Norback (1985) e Golden e Watts (1987) descrevem aplicações do PRV em indústrias de petróleo, químicas, alimentícias e de bebidas.

De modo geral, o PRV consiste no atendimento de um conjunto de clientes (consumidores) por intermédio de uma frota de veículos. Tal problema apresenta uma demanda associada a cada cliente, uma capacidade de carga associada a cada veículo e um depósito central. Assim, a demanda de todos os clientes devem ser atendidas por veículos que iniciam e terminam seu trajeto no depósito (rota). Diversos objetivos podem ser levados em consideração, os mais comuns são minimizar o tempo total de transporte, minimizar a distância total percorrida, minimizar o tempo de espera, minimizar a quantidade de veículos, maximizar o benefício e equilibrar a utilização dos recursos.

Dantzig e Ramser (1959) foram os primeiros a formular o PRV, no qual procuravam resolver um problema real de distribuição de combustível. Desde então, diversas variações do problema surgiram com a finalidade de atender a uma série de problemas reais, estes são formulados adicionando restrições ao PRV. Quando apenas a restrição de capacidade de carga máxima da frota de veículos é definida, o problema é denominado de Problema de Roteamento de Veículos Capacitados (PRVC) (Capacitated Vehicle Routing Problem – CVRP).

Uma importante variação do PRV é o Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT) (Vehicle Routing Problem with Time Windows – VRPTW). Esta variante, além de possuir as restrições de capacidade possui restrições que determinam o horário de atendimento aos clientes. Deste modo, cada cliente está associado a uma janela de tempo  $[a_i, b_i]$  e um tempo de serviço  $s_i$  (descarregamento), onde  $a_i$  é o horário mais cedo e  $b_i$  o horário mais tarde que se pode começar o atendimento. Os veículos devem chegar ao consumidor antes de  $b_i$ . Caso chegue ao consumidor antes do horário definido por  $a_i$ , este deve esperar pela abertura da janela. Contudo, as restrições de tempo aproximam o problema da realidade, ampliando sua utilidade prática e, por consequência, resultando em diversas contribuições para a literatura.

Solomon (1987), por exemplo, propôs quatro heurísticas para resolver o PRVJT. A heurística de economia, que é uma adaptação da heurística de economia de Clarke e Wriarth (1964) para o problema com janelas de tempo. A heurística do vizinho mais próximo, no qual o próximo cliente a ser visitado é o que possui o menor custo em relação ao último cliente adicionado à rota. A heurística da inserção, neste procedimento a rota é construída levando em consideração a distância, o tempo e a urgência de atendimento. E por fim, a última heurística proposta, é uma adaptação da heurística de varredura apresentada por Gillet e Miller (1974).

Yu, Yang e Yao (2011) propõem uma abordagem híbrida, que consiste em um algoritmo de colônia de formigas e busca tabu, para resolver o PRVJT. Inicialmente, o algoritmo de colônia de formigas constrói as soluções seguindo uma regra probabilística. Para melhorar as soluções geradas, dois métodos de vizinhança são aplicados aos mesmos. O primeiro método faz a troca entre dois clientes de rotas diferentes, os clientes e as rotas são escolhidos de forma aleatória. O segundo método escolhe um cliente aleatoriamente e o insere em outra posição. A busca local 2-Opt é usada para melhorar as soluções de vizinhança. A melhor solução obtida até então é usada

como solução inicial do algoritmo de busca tabu, que irá procurar por melhores soluções vizinhas.

Além destes, diversas técnicas foram propostas para resolver o PRVJT como Simulated Annealing (Li e Lim, 2003), busca tabu (Ho e Haugland, 2004) e Algoritmos Genéticos (Cheng e Wang, 2009). Estes trabalhos apresentam soluções para o PRVJT no qual um único objetivo, relacionado a minimização dos custos de transporte, é tratado. Contudo, diversos problemas reais de logística não limitam-se somente aos aspectos de custo, visto que as soluções para estes problemas podem envolver outros fatores como a equidade de trabalho e o atraso de entrega.

Garcia-Najera e Bullinaria (2009), por exemplo, resolvem o PRVJT cujos objetivos são minimizar o custo de transporte e minimizar o número de veículos utilizados, Garcia-Najera e Bullinaria (2011) ainda incluem a minimização do tempo de entrega. Já Hong e Park (1999) procuraram minimizar o tempo total do trajeto percorrido pelos veículos e o tempo total de espera dos consumidores a serem atendidos.

Sessomboon et al (1998) transforma a restrição da janela de tempo em um objetivo, neste caso o autor propõe resolver o PRVJT considerando os seguintes objetivos: minimizar o número de violações da janela de tempo, o atraso ou adiantamento no atendimento aos clientes, o número de veículos e a distância percorrida pelos veículos. Behan (2007) apresenta um modelo semelhante ao de Sessomboon (1998), no qual busca minimizar os custos de transporte e o grau de violação da janela de tempo.

Gutiérrez (2008) além de transformar a restrição de tempo em objetivo, transforma a restrição de capacidade em objetivo, deste modo o autor considera o PRVJT: minimizando a distância total percorrida, o número de veículos, o número de violações da janela de tempo e o número de violações da capacidade do veículo. De forma semelhante, Rahoual et. al (2001) transforma grande parte das restrições em objetivos, minimizando a distância total percorrida pelos veículos, o número de veículos, o número de violações da restrição de distância máxima, o número de violações da restrição de capacidade do veículo, o número de violações da restrição de tempo máximo de tráfego dos veículos e o número de violações da janela de tempo.

Este trabalho propõe um estudo sobre os principais objetivos tratados na literatura para o PRVJT e o conflito existente entre eles. Para mensurar e visualizar o conflito entre os objetivos foi utilizado uma ferramenta denominada de Árvores de Agregação proposta por Freitas et. al (2014). Posteriormente utilizou-se o *Nondominated Sorting Genetic Algorithm II* (NSGA-II) para resolver o PRVJT, considerando os objetivos que apresentaram maior conflito segundo a Árvore de Agregação.

## 2. Modelo Matemático

Apresenta-se, a seguir, uma formulação de programação matemática adaptada de Larsen (1999), para o Problema de Roteamento de Veículos com Janelas de Tempo.

- Variáveis de decisão:

$s_{iv} \rightarrow$  instante de tempo em que o veículo  $v$  atende a um consumidor  $i$ ;

$x_{ijv} \rightarrow \begin{cases} 1 & \text{se há arco ligando o cliente } i \text{ ao cliente } j \text{ usando o veículo } v \\ 0 & \text{caso contrário} \end{cases}$

- Dados de entrada

$V = \{v_1, v_2, \dots, v_m\} \rightarrow$  conjunto de  $m$  veículos idênticos;

$C = \{c_1, c_2, \dots, c_m\} \rightarrow$  conjunto de  $n$  consumidores;

$N = C \cup \{c_0, c_{n+1}\} \rightarrow$  conjunto de consumidores e o depósito central ( $C_0$  e  $C_{n+1}$ );

$R = \{r_1, r_2, \dots, r_n\} \rightarrow$  conjunto de rotas;

$d_{ij} \rightarrow$  distância do consumidor  $i$  ao  $j$ ;

$t_{ij} \rightarrow$  tempo de viagem do consumidor  $i$  ao  $j$ ;

$q \rightarrow$  capacidade máxima de carga dos veículos;

$w_i \rightarrow$  demanda associada ao consumidor  $i$ ;

$p_i \rightarrow$  tempo de serviço no consumidor  $i$ ;

$a_i \rightarrow$  início da janela de tempo no consumidor  $i$ ;

$b_i \rightarrow$  fim da janela de tempo no consumidor  $i$ ;

$k \rightarrow$  constante de ativação da equação (valor escalar suficientemente grande).

- Restrições
  - $\sum_{v \in V} \sum_{j \in N} x_{ijv} = 1, \quad \forall i \in C \quad (1)$
  - $\sum_{i \in N} w_i \sum_{j \in N} x_{ijv} \leq q, \quad \forall v \in V \quad (2)$
  - $\sum_{j \in N} x_{0jv} = 1, \quad \forall v \in V \quad (3)$
  - $\sum_{i \in N} x_{i(n+1)v} = 1, \quad \forall v \in V \quad (4)$
  - $\sum_{i \in N} x_{ijv} - \sum_{i \in N} x_{jiv} = 0, \quad \forall j \in C, \forall v \in V \quad (5)$
  - $s_{iv} + p_i + t_{ij} - k(1 - x_{ijv}) \leq s_{jv}, \quad \forall i, j \in N, \forall v \in V \quad (6)$
  - $a_i \leq s_{iv} \leq b_i, \quad \forall i \in N \quad (7)$
  - $x_{ijv} \in \{0,1\}, \quad \forall i, j \in N \quad (8)$
- Objetivos
  - $\min |R| \quad (9)$
  - $\min \sum_{v \in V} \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ijv} \quad (10)$
  - $\min \sum_{i \in N} \sum_{v \in V} \text{Max}(s_{iv} - b_i, 0) \quad (11)$

A equação (1) garante que cada consumidor é visitado somente por um veículo. A equação (2) especifica que os veículos não devem ultrapassar a capacidade máxima de carga. As equações (3) e (4) demonstram que todos os veículos devem partir e retornar ao depósito central. A equação (5) garante que os veículos devem partir de um consumidor para outro (continuidade). Já as equações (6) e (7) definem as janelas de tempo, enquanto a equação (8) indica a bivalência das variáveis de decisão. As equações (9), (10) e (11) representam a minimização do número de rotas, minimização da distância total percorrida e minimização da violação da restrição de tempo, respectivamente.

Considerando as restrições de (1) a (8) a primeira formulação do problema (PRVJT-A) integra as funções (9) e (10) como objetivos. Já a segunda formulação (PRVJT-B) transforma as restrições de tempo em objetivo. Assim, as restrições (6) e (7) são desprezadas nesta abordagem, que adere como objetivo, as funções (10) e (11). Neste problema uma frota ilimitada de veículos idênticos é considerado.

### 3. Árvores de Agregação

A Árvore de Agregação, proposta por Freitas et. al (2014), é uma ferramenta que possibilita visualizar a redundância e o conflito entre objetivos. Assim, esta abordagem se baseia na organização de objetivos em ramos de uma árvore que representa as melhores possibilidades de agregação entre os mesmos.

A ferramenta tem como entrada uma matriz, no qual as colunas representam os objetivos e as linhas representam os pontos no espaço de busca (possíveis soluções para o problema). Por exemplo, para um problema bi-objetivo  $(f_1, f_2)$  as soluções geradas A, B e C tem valores iguais a (300, 4), (286, 5) e (312, 4), respectivamente. Neste caso, a Árvore de Agregação tem como entrada um matriz com duas colunas e três linhas, como mostrado na Tabela 1.

300	4
286	5
312	4

**Tabela 1** - Representação da matriz de entrada para Árvore de Agregação

Assim, a cada iteração do algoritmo, os objetivos mais harmônicos são agrupados em um novo objetivo composto até que haja um único objetivo composto, que representa o somatório simples de todos os objetivos para otimização mono-objetivo.

Como saída a ferramenta apresenta uma árvore com as seguintes especificidades: (I) os nós folhas representam os objetivos na forma  $f_n$ , onde  $n$  é o número do objetivo (II) pais dos nós folhas representam um objetivo composto na forma  $f_{n1} + f_{n2} - c$ , onde  $f_{n1}$  e  $f_{n2}$  são os objetivos agregados e  $c$  é o conflito existente entre eles (III) outros pais representam objetivos compostos de classes mais altas formados de maneira semelhante, os valores de conflito nestes nós consideram somente o conflito entre os objetivos compostos de seus filhos.

Em suma, cada nó da árvore demonstra o conflito entre seus filhos, sendo que quanto mais distantes dois nós folhas primos são, menos harmonia há entre estes objetivos.

#### 4. Metodologia

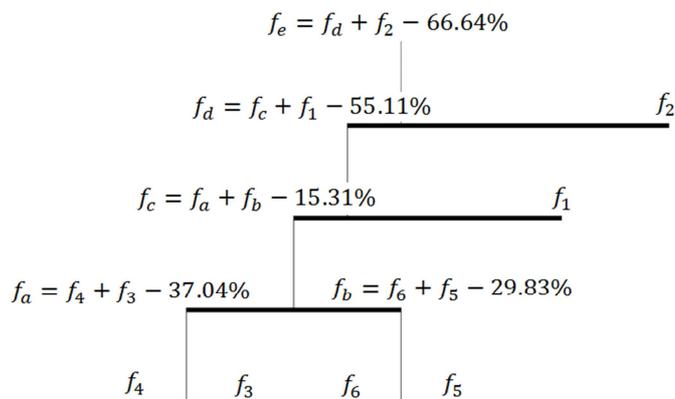
Esta seção descreve a justificativa de escolha das formulações apresentadas na Seção 2 e o algoritmo proposto para resolvê-las.

##### 4.1. Redução/Seleção dos Objetivos

O PRVJT é um dos problemas mais estudados na pesquisa operacional. Deste modo, diversas abordagens são tratadas procurando otimizar objetivos relacionados ao custo de transporte, satisfação do consumidor, satisfação dos motorista, segurança e violação das restrições do problema. Fundamentando-se em tais abordagens e na constância dos objetivos apresentados na literatura, foram escolhidos seis objetivos para o problema tratado neste trabalho. São eles:  $f_1$  = minimizar a distância total percorrida pela frota de veículos;  $f_2$  = minimizar o número de rotas;  $f_3$  = minimizar a violação da restrição de tempo;  $f_4$  = minimizar a espera dos veículos;  $f_5$  = minimizar a diferença entre a maior e a menor rota (balanceamento);  $f_6$  = minimizar a maior rota.

O objetivo  $f_1$  minimiza o custo relacionado a distância total percorrida pela frota de veículos. Este é dado pelo somatório da distância entre os consumidores na ordem em que eles foram visitados. O objetivo  $f_2$  minimiza o número de rotas necessárias para atender aos clientes visto que, uma rota é determinada pela viagem que tem seu início e termino no depósito. O objetivo  $f_3$  é o resultado da transformação da restrição de tempo no próprio objetivo. Assim,  $f_3$  minimiza o somatório dos atrasos dos veículos. Caso um veículo chegue a um consumidor depois do tempo de fim da janela de tempo, o atraso é computado como a diferença entre o tempo de chegada do veículo e o tempo de fim da janela. O objetivo  $f_4$  minimiza o somatório do tempo de espera dos veículos. Caso um veículo chegue a um consumidor antes do tempo de início da janela de tempo, a espera é computada como a diferença entre o tempo de abertura da janela e o tempo de chegada do veículo. O objetivo  $f_5$  minimiza a diferença entre a maior rota e menor rota, tratando-se da distância percorrida nas mesmas. E por fim, o objetivo  $f_6$  minimiza a maior rota existente na solução.

Contudo, dez mil soluções foram geradas por uma permutação aleatória de inteiros (consumidores). Estas foram avaliadas para os objetivos  $f_1, f_2, f_3, f_4, f_5$  e  $f_6$ , no qual a instância R101 proposta por Solomon (1987) foi utilizada. Logo, a matriz de entrada da Árvore de Agregação, que contém seis colunas e dez mil linhas, foi construída tendo como base tais soluções. O resultado obtido pela ferramenta é apresentado na Figura 1.



**Figura 1** – Árvore de Agregação demonstrando o conflito entre os objetivos  $f_1, f_2, f_3, f_4, f_5$  e  $f_6$ .

Analisando a árvore resultante podemos perceber que os objetivos mais harmônicos são agregados na primeira iteração, sendo estes  $f_6$  e  $f_5$ , e  $f_4$  e  $f_3$  com conflitos de 29.83% e 37.04%, respectivamente. Isso indica que não há muito conflito entre o balanceamento e a maior rota, e o atraso e a espera dos veículos. Já considerando a agregação destes objetivos em novas funções

$f_a = f_6 + f_5$  e  $f_b = f_4 + f_3$ , o próximo objetivo mais harmônico é então a combinação destas duas  $f_c = f_a + f_b$ , com 15.31% de conflito. A terceira iteração indica que os próximos objetivos mais harmônicos são  $f_c$  e  $f_1$ , tendo um conflito de 55.11% e resultando no objetivo agregado  $f_d$  ( $f_c + f_1$ ). E por fim, a última iteração sugere a seguinte agregação,  $f_e = f_d + f_2$  com 66.64% de conflito entre  $f_d$  e  $f_2$ . Assim,  $f_e$  representa a agregação de todos os seis objetivos escolhidos no primeiro instante.

O resultado demonstra que o maior conflito existente é então entre  $f_d$  e  $f_2$ . Visto que existe harmonia entre os objetivos agregados de um grupo, se torna indiferente minimizar sua soma ou qualquer um dos objetivos deste grupo individualmente. Deste modo, a última iteração sugere um problema bi-objetivo que minimize  $f_2$  e qualquer um dos objetivos do grupo  $f_d$ . Baseado no resultado apresentado pela árvore resultante, o PRVJT-A procura minimizar o número de rotas ( $f_2$ ) e a distância total percorrida pela frota de veículos ( $f_1$ ). Dentre os objetivos contidos em  $f_d$ ,  $f_1$  foi escolhido por ser mais constante na literatura, porém, qualquer objetivo de  $f_d$  poderia ser selecionado.

Se o número de rotas ( $f_2$ ) não for considerado, a última iteração da árvore seria a agregação entre  $f_c$  e  $f_1$ , ou seja, o segundo maior conflito apresentado é justamente entre estas duas funções (objetivos). De modo semelhante à última iteração, o resultado obtido sugere outro problema bi-objetivo que minimize  $f_1$  e qualquer um dos objetivos do grupo  $f_c$ . Assim, o PRVJT-B minimiza a distância total percorrida pelos veículos ( $f_1$ ) e a violação da restrição de tempo ( $f_3$ ). Dentre os objetivos contido em  $f_c$ ,  $f_3$  foi escolhido pelos mesmos motivos que  $f_1$  no PRVJT-A.

Portanto, este trabalho resolve o PRVJT para os dois maiores conflitos apresentados pela Árvore de Agregação. A primeira abordagem minimiza o número de veículos e a distância total percorrida pela frota, enquanto a segunda minimiza a distância total percorrida pelos veículos e a violação da restrição de tempo.

## 4.2. Algoritmo Genético Proposto

Para resolver o PRVJT-A e o PRVJT-B foi implementado um Algoritmo Genético multi-objetivo baseado no NSGA-II (Deb et. al, 2002). As seções seguintes apresentam os detalhes deste algoritmo.

### 4.2.1. Codificação e Decodificação de uma Solução

Para representar um indivíduo (solução), o AG utiliza uma lista que armazena os consumidores na sequência em que eles devem ser visitados pela frota de veículos. Ao fazer a leitura do indivíduo o algoritmo inclui um zero (representação do depósito) no início da lista, o que representa que o veículo inicia a primeira rota no depósito. Visto que o problema possui as restrições de capacidade de carga máxima e de janelas de tempo, existe a possibilidade de um determinado veículo, ao atender o próximo consumidor, violar alguma restrição do problema. Caso isso ocorra, o número zero é inserido entre o último consumidor atendido pelo veículo e o consumidor subsequente da lista, representando a volta do veículo para o depósito central. Por fim, o zero também é incluso no final da lista, garantindo que a última rota finalize sua viagem no depósito. Este processo garante que nenhuma restrição seja violada, garantindo a factibilidade das soluções geradas.

### 4.2.2. Inicialização

A população inicial é construída por meio de três métodos diferentes. O primeiro método gera uma solução que, a cada iteração, insere o consumidor mais próximo do inserido anteriormente. Se o consumidor a ser inserido for o primeiro da rota, o depósito central é considerado como o último consumidor inserido na solução. Como este método é determinístico, uma única solução é inserida na população inicial. O segundo método é uma versão aleatória do Push-Forward Insertion Heuristic (PFIH). Este método se diferencia do PFIH, proposto por Solomon (1987), por inicializar as rotas com consumidores selecionados aleatoriamente. E por fim, o terceiro método de inicialização gera soluções completamente aleatórias. Os dois últimos métodos completam a população inicial com a mesma quantidade de soluções.

### 4.2.3. Crossover

Neste trabalho os métodos de *crossover* implementados foram os seguintes: Cruzamento de Mapeamento Parcial (PMX); Cruzamento de ordem (OX). Para cada par de pais selecionado para o cruzamento, um destes dois métodos é escolhido aleatoriamente.

#### 4.2.3.1. Cruzamento de Mapeamento Parcial

O operador PMX transfere informações de ordem e de posição das rotas dos pais para as rotas dos filhos. Uma parte da sequência de um pai é mapeada e uma parte da sequência do outro pai é preservada no filho, o restante das informações é trocado entre os pais (Malaquias, 2006).

Utilizando-se como exemplo a sequência (1 2 3 4 5 6 7 8) como a rota do pai 1 e (3 7 5 1 6 8 2 4) como a rota do pai 2, o operador PMX inicialmente seleciona dois pontos de corte aleatoriamente para a divisão dos pais. Partindo do princípio que o primeiro ponto de corte está entre o terceiro e quarto elemento (gene), e o segundo ponto de corte entre o sexto e sétimo elemento os pais são representados como segue na Figura 2.

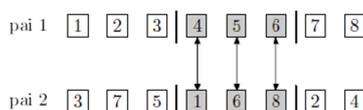


Figura 2 - Exemplo da seção de Mapeamento do PMX (Malaquias, 2006).

No passo seguinte o PMX copia os genes da segunda parte do pai 1 para a segunda parte do filho 2. Da mesma forma, copiam-se os genes da segunda parte do pai 2 para a segunda parte do filho 1. O conjunto de genes pertencentes à segunda parte dos indivíduos é denominado de seção de mapeamento. A Figura 3 ilustra o processo de cópia da seção de mapeamento.

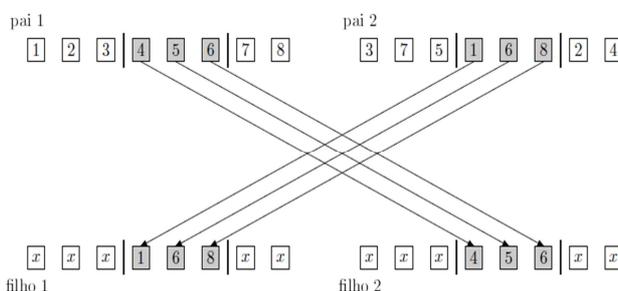


Figura 3 - Cópia dos elementos da seção de mapeamento dos pais para os filhos (Malaquias, 2006).

Por fim, o PMX copia o restante dos genes do pai 1 para as respectivas posições do filho 1, começando pelo primeiro elemento da terceira parte. Caso o pai 1 tente inserir algum gene já existente no filho 1, o PMX verifica a posição deste gene no filho e, tenta inserir outro gene do pai correspondente a essa posição. Isso é feito até que se encontre um gene inédito para o filho. O mesmo processo acontece para o pai 2 e o filho 2. Assim, os filhos são preenchidos como segue na Figura 4.

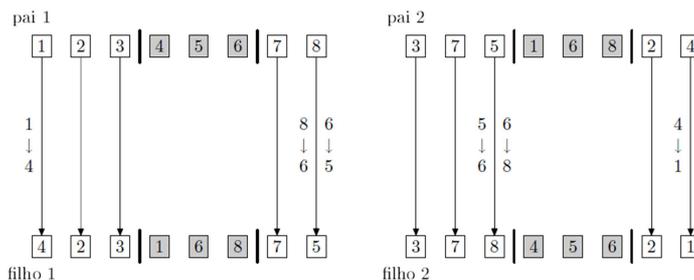
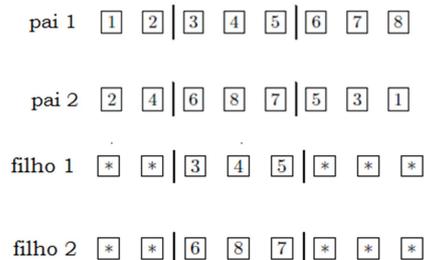


Figura 4 - Cópia dos elementos restantes dos pais para os filhos (Malaquias, 2006).

#### 4.2.3.2. Cruzamento de Ordem

O operador OX explora a propriedade de representação do caminho, em que a ordem é importante e não a posição. Ele escolhe um sub-roteiro de um dos pais preservando a ordem dos elementos do outro pai (Malaquias, 2006).

Assim como no PMX o operador OX inicia gerando dois pontos de corte aleatoriamente e copiando as seções de mapeamento dos pais para os filhos. Este processo é ilustrado nas Figuras 5.



**Figura 5** - Resultado da cópia de seção de mapeamento nos filhos (Malaquias, 2006).

Por fim, o operador OX copia o restante dos elementos dos pais para as respectivas posições dos filhos, começando pelo primeiro elemento da terceira parte. A diferença entre o PMX e o OX é que, caso o pai tente inserir algum elemento já existente no filho, o operador OX busca o próximo elemento a direita contido no pai, até que se encontre um elemento inédito para o filho. Os filhos são representados pela Figura 6.



**Figura 6** - Resultado da execução do OX (Malaquias, 2006).

#### 4.2.4. Mutação

Neste trabalho foram implementadas a mutação por inserção (ISM), mutação por inversão simples (SIM) e a mutação por troca (EM). Para cada indivíduo selecionado para a mutação, um destes três métodos é escolhido aleatoriamente.

O operador ISM escolhe aleatoriamente um elemento na sequência, remove este elemento e o insere em uma posição escolhida aleatoriamente (Malaquias, 2006). Partindo do princípio de que o indivíduo selecionado tenha a seguinte sequência de genes (1 2 3 4 5 6 7 8) e que o quarto gene foi escolhido para preencher a sétima posição, o resultado da mutação é (1 2 3 5 6 7 4 8).

O operador SIM seleciona aleatoriamente dois pontos de corte e inverte os elementos do subconjunto formado a partir dos pontos de corte (Malaquias, 2006). Suponhamos que o indivíduo selecionado tenha a seguinte sequência de genes (1 2 3 4 5 6 7 8) e que o primeiro e o segundo ponto de corte estão entre o primeiro e o segundo gene, e o quinto e o sexto gene, respectivamente. Invertendo o subconjunto (2 3 4 5) o resultado da mutação é (1 5 4 3 2 6 7 8).

O operador de mutação por troca (EM) seleciona aleatoriamente duas posições no indivíduo e troca suas posições (Malaquias, 2006). Considerando a sequência (1 2 3 4 5 6 7 8) como exemplo e que os genes da terceira e da quinta posição foram selecionados, tem-se (1 2 5 4 3 6 7 8) como resultado da mutação.

### 5. Resultados e Discussões

Para testar o NSGA-II desenvolvido, foi utilizado o conjunto de instâncias propostas por Solomon (1987) com 100 consumidores e um depósito central. Estas contêm as informações de localização (coordenadas x e y), demanda, janela de tempo e tempo de serviço de cada consumidor, além de fornecer a capacidade de carga máxima da frota de veículos. As instâncias são divididas em três grupos com características distintas, estas são do tipo C (Clusterizadas), R

(Randomizadas) e RC (união das características de R e C). Deste modo, os testes foram executados com uma instância de cada grupo (C105, R104 e RC101).

Para cada instância testada o NSGA-II foi executado 10 vezes, gerando 10 Paretos com “n” soluções, sendo  $n > 0$ . As soluções resultantes (soluções pertencentes aos Paretos das 10 execuções) passaram pelo processo de não dominância, ou seja, todos os indivíduos que foram retornados pelo algoritmo como parte do resultado foram comparados entre si, sendo que as soluções não dominadas geraram um único Pareto denominado de Pareto-médio. Os parâmetros do algoritmo foram definidos como segue:

Tamanho da população = 100      Taxa de cruzamento = 0.9  
Número de gerações = 1000      Taxa de mutação = 0.1

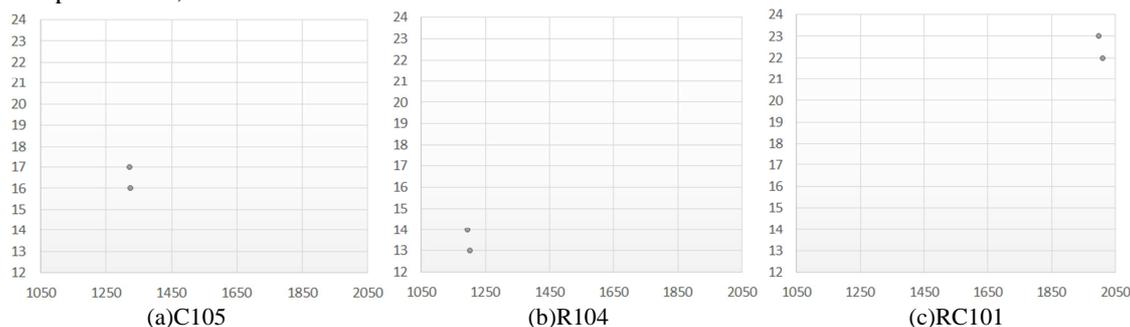
Os testes foram realizados em um computador Pentium i5 com 6GB RAM. As seções 5.1 e 5.2 apresentam os Paretos de uma execução típica (Pareto-típico) do NSGA-II e os Paretos-médio para o PRVJT-A e o PRVJT-B, todas as execuções típicas de Pareto apresentadas foram escolhidas aleatoriamente.

### 5.1. Resultados do PRVJT-A

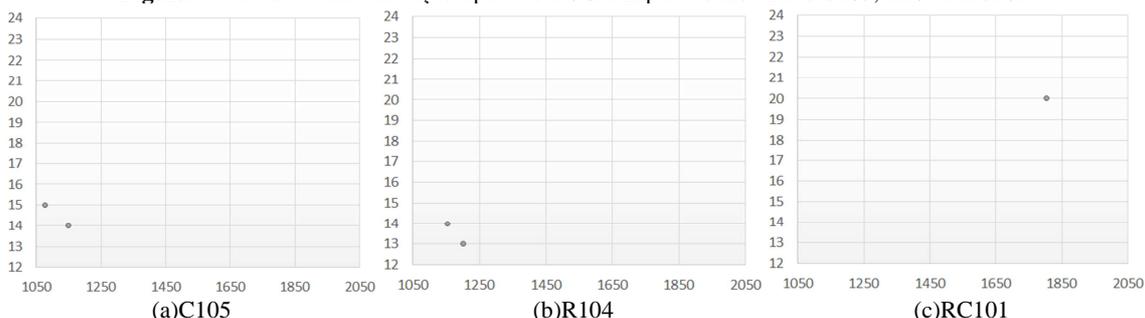
Ao otimizar o número de veículos e a distância total percorrida, o NSGA-II apresentou uma ou duas soluções de Pareto para cada execução do algoritmo. Experimentos realizados por Garcia e Bullinaria (2011) demonstram que para 27 instâncias o resultado apresentou Paretos que continham uma solução, para 24 instâncias os Paretos continham duas soluções, e para as outras 5 instâncias foram retornados Paretos com 3, 4 e 5 soluções, somando um total de 56 instâncias. Deste modo, A natureza das instâncias propostas por Solomon (1987) não possibilita a formação de Paretos com muitas soluções para os objetivos tratados no PRVJT-A.

Outro aspecto que pode influenciar na formação de Paretos com poucas soluções é que o objetivo que minimiza o número de rotas é discreto, e o algoritmo insere o número de rotas de maneira gulosa na solução (seção 4.2.2.). Assim, o algoritmo não gera soluções com a mesma ordem de atendimento aos clientes com número de rotas diferentes, ou seja, se a ordem de atendimento é a mesma, o número de rotas é igual para todas estas soluções. Fazendo que o espaço de busca para este objetivo seja reduzido.

A Figura 7 e a Figura 8 demonstram as soluções de Pareto de uma execução típica do algoritmo e de Pareto-médio para as instâncias testadas. O eixo horizontal apresenta a distância total percorrida, e o eixo vertical o número de rotas.



**Figura 7.** Pareto de uma execução típica do NSGA-II para as instâncias C105, R104 e RC101.



**Figura 8.** Pareto-médio resultante de 10 execuções do NSGA- II para as instâncias C105, R104 e RC101.

Para a instância C105, o Pareto de uma execução apresentou duas soluções com 16 e 17 rotas e distância igual a 1324,10 e 1320,70, respectivamente. O Pareto-médio também apresentou duas soluções, estas contendo 14 e 15 rotas e 1149,37 e 1075,71 de distância percorrida pela frota de veículos. A diferença considerável entre as soluções do Pareto-típico e do Pareto-médio, principalmente com relação ao número de rotas, é que o Pareto-médio foi formado por soluções pertencentes à Paretos que retornaram somente uma solução. Contudo, nenhuma execução típica para a instância C105 retornou um Pareto contendo duas soluções que apresentasse 14 e 15 rotas.

Para a instância R104, o Pareto-típico e o Pareto-médio apresentaram duas soluções com o número de rotas igual a 13 e 14. Diferente dos resultados retornados para a instância C105, os resultados obtidos para R104 demonstram que o NSGA-II retornou soluções com o mesmo número de rotas do Pareto-médio. Sendo que a única diferença entre este e o Pareto-típico foi a distância total percorrida.

Os resultados apresentados pelo NSGA-II para a instância RC101 demonstram que, diferente dos dois casos anteriores, o Pareto-médio foi formado por somente uma solução. Isto porque de todas as execuções para esta instância, um Pareto-típico foi gerado contendo uma solução com 20 rotas e 1804,02 de distância, dominando todos os outros Paretos-típicos gerados.

A melhor solução conhecida para as instâncias testadas são apresentadas na Tabela 2 (www.sintef.no).

Instâncias	Rotas	Distância
C105	10	828,94
R104	9	1007,31
RC101	14	1696,94

Tabela 2. Melhores resultados conhecidos para as instâncias C105, R104 e RC101.

## 5.2. Resultados do PRVJT-B

Ao otimizar a distância total percorrida e a violação da restrição de tempo (atraso total), o NSGA-II apresentou, em média, oitenta soluções de Pareto para cada execução do algoritmo. Isto se deve porque o PRVJT-B procura minimizar duas funções contínuas, sendo que, qualquer mudança realizada em uma rota altera o valor de fitness para os dois objetivos. Isso não ocorre no PRVJT-A, pois caso uma troca seja feita dentro de uma mesma rota, o valor de fitness para a distância total percorrida é alterada, porém, dificilmente o número de rotas será modificado.

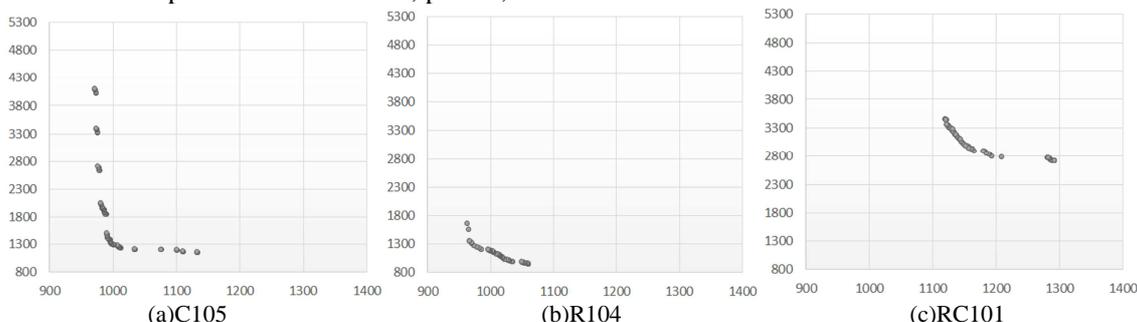


Figura 9. Pareto de uma execução típica do NSGA-II para as instâncias C105, R104 e RC101.

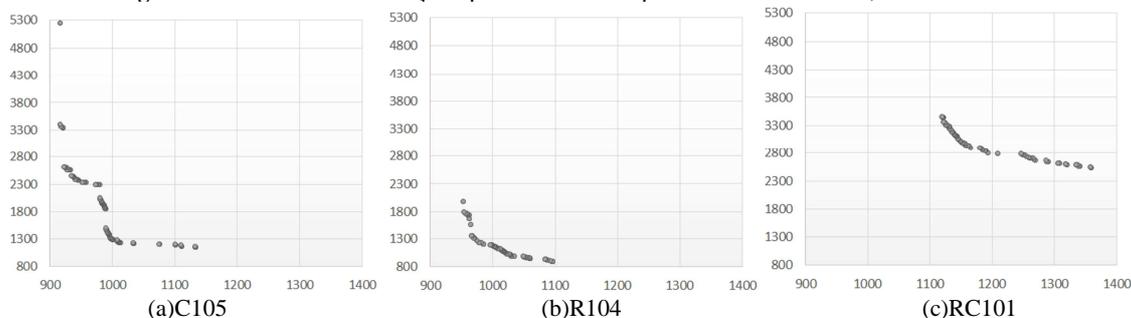


Figura 10. Pareto-médio resultante de 10 execuções do NSGA-II para as instâncias C105, R104 e RC101.

A Figura 9 e a Figura 10 demonstram as soluções de Pareto de uma execução típica do algoritmo e de Pareto-médio para as instâncias testadas. O eixo horizontal apresenta a distância

total percorrida, e o eixo vertical o atraso total. Para as instâncias testadas o valor do tempo é igual ao valor de distância, ou seja, se a distância entre os consumidores A e B é igual a 200, gastasse 200 unidades de tempo para percorrer o trajeto do consumidor A ao consumidor B.

Para a instância C105 o Pareto-típico apresentou resultados que variaram entre 970,95 e 1133,61 para o eixo horizontal (distância), e entre 1149,37 e 4097,90 para o eixo vertical (atraso). Já o Pareto-médio retornou soluções que variaram entre 916,34 e 1133,61 para valores de distância, e entre 1149,37 e 5242,9 para valores de atraso total.

Os resultados obtidos pelo NSGA-II para a instância R104 apresentam o mesmo comportamento de Pareto da instância C105. Sendo que o Pareto-típico obteve soluções no intervalo de 962,56 e 1059,61 para os valores de distância, e entre 951,84 e 1671,13 para valores de atraso. O Pareto-médio apresentou soluções parecidas, com intervalos entre 953,11 e 1096,18 para a distância e 899,88 e 1980,5 para o atraso.

O Pareto-típico referente à instância RC101 demonstra que as soluções variaram entre 1119,05 e 1291,98 no eixo horizontal, e entre 2726,95 e 3456,36 no eixo vertical. O Pareto-médio apresenta os resultados variando entre 1119,05 e 1359,42 para o eixo horizontal, e entre 2542,02 e 3456,36 para o eixo vertical.

Contudo, todas as instâncias retornaram diversas soluções de Pareto, tendo em média 80 soluções para Paretos-típicos e 120 para Paretos-médios. Os resultados apontam ainda, que o intervalo apresentado pelo eixo vertical (atraso) sempre foi maior que o intervalo do eixo horizontal (distância), isto quer dizer que, à medida que a distância total percorrida é minimizada o atraso para atender os consumidores cresce em uma proporção maior. Assim, esta abordagem possibilita a escolha de soluções com valores de distância consideravelmente menores do que da abordagem anterior (PRVJT-A), porém, estas soluções apresentam atrasos no atendimento aos clientes, deixando a critério do decisor a escolha de tal solução. Caso este atraso seja inaceitável, somente as soluções do PRVJT-A devem ser consideradas.

## 6. Considerações Finais

Este trabalho propôs um estudo sobre os principais objetivos tratados na literatura para o Problema de Roteamento de Veículos com Janelas de Tempo. Assim, uma ferramenta denominada de Árvores de Agregação foi utilizada para verificar a harmonia e o conflito existente entre estes objetivos. O resultado demonstrou que a distância total percorrida pelos veículos e o número de rotas foram os objetivos que apresentam maior conflito, seguido da distância total percorrida pelos veículos e a violação da restrição de janelas de tempo.

A partir destes resultados um algoritmo genético multiobjetivo (NSGA-II) foi implementado para verificar os resultados para ambas as abordagens. Os testes realizados utilizaram as instâncias de Solomon (1987). A primeira abordagem (PRVJT-A) apresentou poucos pontos de Pareto, enquanto os Paretos da segunda abordagem (PRVJT-B) foram formados por diversas soluções.

Contudo, os resultados apresentados pelo PRVJT-A foram razoáveis quando comparados com os melhores resultados conhecidos na literatura. Já o PRVJT-B apresentou resultados bons com relação à distância percorrida pelos veículos, porém, as soluções desta abordagem aceitam atrasos no atendimento dos consumidores. Deste modo, pode-se concluir que os testes confirmaram o conflito indicado pela Árvore de Agregação, pois à medida que o algoritmo minimizava os valores de um determinado objetivo, resultados piores eram encontrados para o outro objetivo.

## 7. Referências

- Beham, A.** (2007). Parallel tabu search and the multiobjective capacitated vehicle routing problem with soft time windows. *Computer Aided Systems Theory – EUROCAST 2007*, volume 4739 of *Lecture Notes in Computer Science*, pp. 829–836. Springer Berlin Heidelberg.
- Cheng, C. B. & Wang, K. P.** (2009). Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm, *Expert Systems with Applications*, v. 36(4), p. 7758–7763.

- Clark, G. & Wright, J. W.** (1964). Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research*, v. 12, p. 568-581.
- Dantzig, G. B. & Ramser, J. H.** (1959). The truck dispatching problem, *Management Science*, v. 6, pp. 80-91.
- Debian, K., Pratap, A., Agarwal, S. e Meyarivan, T.** (2002), A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, v. 6(2), pp. 181–197.
- Freitas, A. R. R., Pedrosa-Silva, R. C., Guimarães, F. G.** (2014). On the Visualization of Trade-offs and Reducibility in Many-Objective Optimization. Genetic and Evolutionary Computing Conference, 2014, Vancouver. Proceedings of the Genetic and Evolutionary Computing Conference.
- Garcia-Najera, A. & Bullinaria, J. A.** (2009). Bi-objective optimization for the vehicle routing problem with time windows: Using route similarity to enhance performance. *International Conference on Evolutionary Multi-Criterion Optimization*, volume 5467 of *Lecture Notes in Computer Science*, pp. 275–289. Springer.
- Garcia-Najera, A. & Bullinaria, J. A.** (2011). An improved multi-objective evolutionary algorithm for the vehicle routing problem with time Windows, *Computers & Operations Research*, v. 38, pp. 287–300.
- Gutiérrez, J. P. C., Landa-Silva, D. & Moreno-Pérez, J. A.** (2008). Exploring feasible and infeasible regions in the vehicle routing problem with time windows using a multi-objective particle swarm optimization approach, *International Workshop on Nature Inspired Cooperative Strategies for Optimization*.
- Ho, S. C. & Haugland, D.** (2004). A Tabu search heuristic for the vehicle routing problem with time windows and split deliveries, *Computers & Operations Research*, v. 31(12), p. 1947–1964.
- Hong, S.-C. & Park, Y.-B.** (1999). A heuristic for bi-objective vehicle routing with time window constraints, *International Journal of Production Economics*, 62(3):249 – 258.
- Larsen, J.** (1999). Parallelization of the Vehicle Routing Problem with Time Windows. Tese (Institute of Mathematical Modelling), Technical University of Denmark, Lyngby, Denmark.
- Li, H. & Lim, A.** (2003). Local search with annealing-like restarts to solve the VRPTW, *European Journal of Operational Research*, v. 150(1), p. 115–127.
- Malaquias, N. G. L.** (2006). Uso dos algoritmos genéticos para a otimização de rotas de distribuição. Dissertação (Mestrado em Ciências), Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia.
- Matsueda, L. C. & Martins, F. V. C.** (2013). Abordagem mono-objetivo e multiobjetivo para o Problema de Roteamento de Veículos, *Simpósio Brasileiro de Pesquisa Operacional*.
- Rahoual, M., Kitoun, B., Mabed, M., Bachelet, V. & Benameur, F.** (2001). Multicriteria genetic algorithms for the vehicle routing problem with time Windows, *Metaheuristic International Conference*.
- Sessomboon, W., Watanabe, K., Irohara, T. & Yoshimoto, K.** (1998). A study on multiobjective vehicle routing problem considering customer satisfaction with due-time (the creation of Pareto optimal solutions by hybrid genetic algorithm), *Transaction of the Japan Society of Mechanical Engineering*.
- Solomon, M.M.** (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints, *Operations Research*, v. 35, pp. 254–265.
- Yu, B., Yang, Z. Z. & Yao, B. Z.** (2011). A hybrid algorithm for vehicle routing problem with time Windows, *Expert Systems with Applications*, v. 38, p. 435-441.