

***Metaheurísticas Híbridas Baseadas em
Redes Elásticas e Algoritmos Evolutivos
Para a Solução de Problemas de
Roteamento de uma Frota de Veículos***

Dissertação de Mestrado

Ana Lúcia Gondim Tavares Ribeiro
Mestranda

Luiz Satoru Ochi
Orientador

Maria Cristina Silva Boeres
Orientadora

Niterói
Maio/1999

Índice

<i>Capítulo 1 – Introdução.....</i>	<i>1</i>
<i>Capítulo 2 – O Problema de Roteamento de Veículos com Frota Heterogênea.....</i>	<i>4</i>
2.1 <i>Introdução.....</i>	<i>4</i>
<i>Capítulo 3 – Metaheurísticas.....</i>	<i>6</i>
3.1 <i>Introdução.....</i>	<i>6</i>
3.2 <i>Algoritmo Elástico.....</i>	<i>6</i>
3.2.1 <i>Descrição do Algoritmo Elástico.....</i>	<i>7</i>
3.2.2 <i>Modelo Escolhido.....</i>	<i>9</i>
3.3 <i>Modelo de Pétalas.....</i>	<i>10</i>
3.4 <i>Busca Tabu.....</i>	<i>11</i>
3.5 <i>Método 2-Optimal.....</i>	<i>13</i>
<i>Capítulo 4 – Algoritmos Propostos.....</i>	<i>16</i>
4.1 <i>Introdução.....</i>	<i>16</i>
4.2 <i>O Algoritmo Pétala-Elástico.....</i>	<i>16</i>
4.3 <i>O Algoritmo Diversificação-Intensificação.....</i>	<i>21</i>

<i>Capítulo 5 – Resultados Obtidos</i>	24
5.1 <i>Introdução</i>	24
5.2 <i>Os Problemas Teste</i>	24
5.3 <i>A Implementação</i>	25
5.3.1 <i>Os Parâmetros do Algoritmo Elástico</i>	25
5.4 <i>Apresentação dos resultados</i>	26
5.4.1 <i>Algoritmo Pétala-Elástico</i>	27
5.4.1.1 <i>Análise das melhores soluções</i>	28
5.4.2 <i>Algoritmo Diversificação-Intensificação</i>	29
5.4.3 <i>Resumo dos resultados encontrados</i>	33
<i>Capítulo 6 – Conclusões</i>	35
<i>Referências Bibliográficas</i>	36

Índice de Figuras

<i>Figura 1 - F_1 e F_2 são, respectivamente, as forças de atração da cidade e dos.....</i>	<i>7</i>
<i>Figura 2 - Pseudocódigo do Algoritmo Elástico.....</i>	<i>9</i>
<i>Figura 3 – Divisão em Pétalas.....</i>	<i>10</i>
<i>Figura 4 – Algoritmo Básico de Busca Tabu.....</i>	<i>12</i>
<i>Figura 5 – Pseudocódigo do Método 2-optimal.....</i>	<i>14</i>
<i>Figura 6 – Exemplificação do método 2-optimal.....</i>	<i>15</i>
<i>Figura 7 – Formação das Pétalas.....</i>	<i>17</i>
<i>Figura 8 - Construção das flores.....</i>	<i>19</i>
<i>Figura 9 - Pseudocódigo Algoritmo Pétala-Elástico.....</i>	<i>20</i>
<i>Figura 10 - Soluções em formato de Flor e Flor Artificial.....</i>	<i>21</i>
<i>Figura 11 - Pseudocódigo Algoritmo Diversificação-Intensificação.....</i>	<i>23</i>
<i>Figura 12 – Evolução do Algoritmo Elástico.....</i>	<i>27</i>

Índice de Tabelas

<i>Tabela 1 – Problemas de Teste</i>	26
<i>Tabela 2 – Parâmetros do Algoritmo Elástico</i>	27
<i>Tabela 3 – Resultados do Algoritmo Pétala-Elástico</i>	28
<i>Tabela 4 – Análise dos custos das soluções finais</i>	29
<i>Tabela 5 – Resultados do Algoritmo Diversificação-Intensificação 1ª Versão</i>	30
<i>Tabela 6 - Resultados do Algoritmo Diversificação-Intensificação 2ª Versão</i>	31
<i>Tabela 7 - Resultados do Algoritmo Diversificação-Intensificação 3ª Versão</i>	32
<i>Tabela 8 - Resumo dos resultados do Algoritmo Diversificação-Intensificação</i>	33
<i>Tabela 9 – Resumo dos resultados encontrados pelos algoritmos propostos</i>	35

Índice dos Gráficos

<i>Gráfico 1 – Resultados do Algoritmo Pétala-Elastico.....</i>	<i>28</i>
<i>Gráfico 2 – Análise dos custos das soluções finais</i>	<i>29</i>
<i>Gráfico 3 - Resultados do Algoritmo Diversificação-Intensificação 1ª Versão.....</i>	<i>31</i>
<i>Gráfico 4 - Resultados do Algoritmo Diversificação-Intensificação 2ª Versão.....</i>	<i>32</i>
<i>Gráfico 5 - Resultados do Algoritmo Diversificação-Intensificação 3ª Versão.....</i>	<i>33</i>
<i>Gráfico 6 - Resumo dos resultados do Algoritmo Diversificação-Intensificação.....</i>	<i>34</i>
<i>Gráfico 7 —Resumo dos resultados encontrados pelos algoritmos propostos.....</i>	<i>35</i>

Índice dos Anexos

<i>Anexo 1 - Método Heurístico de Geração de Colunas.....</i>	<i>i</i>
---	----------

Capítulo 1

Introdução

Um problema de Otimização Combinatória pode ser caracterizado pela minimização ou maximização de uma função f de variáveis independentes x_1, \dots, x_n definidas sobre o domínio D enumerável, quase sempre de cardinalidade finita. Usualmente denominada função-objetivo ou função-custo, f pode ser definida por $f: D_n \rightarrow R$

A busca do mínimo (ou máximo) de f é realizada sobre um subconjunto de D_n , determinado por um conjunto de restrições ou critérios C , especificado para o problema. Esse subconjunto é denominado conjunto dos Pontos Viáveis.

Como o conjunto de Pontos Viáveis pode assumir uma quantidade de valores muito grande, a obtenção de uma solução exata para o problema através da especificação de um algoritmo para resolvê-lo, pode exigir um *tempo* de computação inviável. A medida desse *tempo* determina a *complexidade* do algoritmo. Quando a *complexidade* de um algoritmo pode ser limitada por uma função polinomial no tamanho do problema, diz-se que o algoritmo é *eficiente*. Caso contrário, o algoritmo é dito *ineficiente* ou *exponencial*.

Um problema é dito NP-completo quando não se conhece nenhum algoritmo *eficiente* capaz de resolvê-lo. Muitos problemas de Otimização Combinatória são NP-completos. Em função da importância desses problemas, surgiram heurísticas ou técnicas aproximativas para resolvê-los num tempo de computação viável, porém sem a garantia de obtenção da solução ótima.

O problema em que estamos interessados aqui é o Problema de Roteamento de Veículos — *PRV* que é classificado como um problema NP-Completo. Devido ao grande

número de algoritmos heurísticos existentes para solucioná-lo, este problema é um dos temas de maior sucesso dentro da Pesquisa Operacional e da Otimização Combinatória. No entanto, a maioria absoluta das heurísticas propostas são as chamadas *heurísticas convencionais*, baseadas em conceitos de economias, vizinho mais próximo, inserção, além de técnicas de refinamento tal como o *k-optimal* [20]. Estes procedimentos embora flexíveis possuem limitações que muitas vezes acarretam numa solução final de baixa qualidade. Uma das limitações é a parada prematura em ótimos locais às vezes distantes de um ótimo global em problemas de otimização.

A partir dos anos 80 começaram a surgir na literatura heurísticas mais genéricas que incluem ferramentas para tentar escapar das armadilhas dos ótimos locais. Estes procedimentos passaram a ser conhecidos como *heurísticas inteligentes* ou *Metaheurísticas*. Dentre as Metaheurísticas mais conhecidas temos Algoritmos Genéticos (AG), Redes Neurais Artificiais (Rns), Busca Tabu (BT), Simulated Annealing (SA), Ant System (AS), GRASP, Scatter Search (SS), entre outros (veja [11][23][24][25][26][27][28][29]).

Este trabalho apresenta uma nova proposta para a solução do PRV incluindo uma decomposição em pétalas (*Petal Decomposition*)[28][29] para a seleção de clientes em cada rota e um procedimento baseado em Redes Elásticas — *RE (Elastic Nets)* [24] para o problema de geração das rotas dos veículos. O método de decomposição em pétalas é um procedimento baseado em um processo de varredura para gerar *clusters* que estarão associados a uma rota de um veículo para o PRV. As redes elásticas são uma proposta alternativa a modelos clássicos de Rede Neurais Artificiais, cujo desempenho em problemas altamente combinatorios não tem, em muitas aplicações, obtido soluções competitivas quando confrontadas às melhores Metaheurísticas existentes. Em particular, as redes elásticas tem alcançado bons resultados na solução do Problema do Caixeiro Viajante (*Travelling Salesman Problem*) e algumas de suas generalizações [24][27].

Nosso principal objetivo é apresentar uma metodologia capaz de gerar soluções de qualidade em tempos computacionais pequenos.

Este trabalho está dividido em mais cinco capítulos. No Capítulo 2, o Problema de Roteamento de Veículos a ser tratado é apresentado. No Capítulo 3 são abordados os métodos utilizados no sistema proposto, que é apresentado no Capítulo 4. No Capítulo 5 os testes realizados com os algoritmos são apresentados, analisados e avaliados em relação ao

algoritmo proposto por Éric Taillard [40]. E, finalmente, no Capítulo 6 é realizada uma análise de todo o trabalho através da apresentação das conclusões.

Capítulo 2

O Problema de Roteamento de Veículos com Frota Heterogênea

2.1 Introdução

O Problema de Roteamento de Veículos - *PRV* (*Vehicle Routing Problem* — *VRP*) é um problema NP-árduo, tradicionalmente atacado na área de otimização[3] [4] [10] [15] [26] [28] [31]. Dependendo das características consideradas, como por exemplo, o tipo da frota (homogênea ou heterogênea) e o período de em que o cliente deve ser atendido (*time window*), podemos ter diversas variações deste problema. O Problema de Roteamento modela a situação em que uma companhia necessita determinar um bom escalonamento dos veículos de sua frota para atender a seus clientes e distribuir/coletar suas mercadorias. Visto que, o atendimento a clientes, a redução de custos operacionais e o escoamento de produção são objetivos altamente desejados, o modelo deste problema pode ser aplicado para representar muitas situações de otimização no mundo real.

No *PRV* clássico, veículos de capacidade fixa Q , devem atender a um número n de clientes com demanda $q_i > 0, (i = 1, \dots, n)$, a partir de um depósito ($i = 0$). A distância (custo) entre os clientes i e j ($i, j = 0, \dots, n$) é suposta conhecida na maioria dos casos. O problema consiste em encontrar rotas para os veículos de forma que:

- A distância total percorrida por cada veículo seja minimizada;
- Somente um veículo será escalonado para atender a um mesmo cliente, em uma única visita;

- A quantidade de produto transportada por um veículo não pode exceder sua capacidade Q .

No Problema de Roteamento de Veículos com Frota Heterogênea - (*Vehicle Routing Problem with Heterogeneous Fleet —VRPHE*) existe um conjunto $\Psi = \{1, \dots, K\}$ de K tipos de veículos com capacidades diferentes. O veículo $k \in \Psi$ possui capacidade Q_k e seu número disponível é n_k . O custo da viagem ou distância entre os clientes i e j ($i, j = 0, \dots, n$) com o veículo k é dado por d_{ijk} . A todo veículo k está associado um custo fixo f_k , que representa o custo da utilização deste veículo.

Existe um caso especial do *PRVFH*, também chamado de *Vehicle Fleet Mix - VFM*, onde o objetivo é determinar uma frota de veículos de forma que a soma dos custos fixos de cada veículo e o custo da rota sejam minimizados. Este problema é um caso especial, onde:

- O custo da viagem entre os clientes i e j ($i, j = 0, \dots, n$) é o mesmo para qualquer veículo, ou seja, $d_{ijk} = d_{ijk'}, \forall k, k' \in \Psi$;
- O número de veículos k disponível é ilimitado ($n_k = \infty$).

Golden *et al.* [15] foi o primeiro a abordar o problema *VFM*. O conjunto de testes proposto por ele foi utilizado por Taillard [40] e também será utilizado neste trabalho.

O problema a ser abordado neste trabalho é o *VFM*. Esta escolha se deve à necessidade do desenvolvimento de um estudo mais aprofundado, no que diz respeito à procura de melhores soluções, já que este problema tem sido menos tratado pela literatura, devido, principalmente, ao fato de ser um problema de resolução mais complexa que o *PRV* clássico.

Apesar dos algoritmos propostos se designarem à solução deste tipo de *PRV*, eles poderão ser adaptados, sem grandes esforços de programação, a outros tipos de *PRV*, como por exemplo, ao *PRV* com frota homogênea.

Capítulo 3

Métodos Utilizados

3.1 Introdução

Neste capítulo serão apresentadas os métodos utilizados para o desenvolvimento dos algoritmos propostos e apresentados no Capítulo 4.

3.2 Algoritmo Elástico

Durbin & Willshaw [8] propuseram uma técnica analógica para resolver o Problema do Caixeiro Viajante (*PCV*). Nesta técnica, denominada **Algoritmo Elástico**, um anel é gradativamente alongado em direção às cidades, que são representadas por pontos distribuídos no plano Euclidiano. Ao final do processamento, o anel passa por todas as cidades, formando uma rota. Este algoritmo possui características essencialmente geométricas, pois a deformação do anel é causada através da atualização das coordenadas de seus pontos. Esta atualização é realizada em função das distâncias dos pontos do anel às cidades no plano.

Na Figura 1 a seguir, podemos ver representadas as duas forças exercidas sobre o ponto Y_j da rede elástica. A força F_1 é a força de atração exercida pela cidade e a força F_2 é a exercida pelos pontos vizinhos (imediatamente anterior e posterior) à Y_j . Os pontos pertencentes à rede elástica estão representados por quadrados e as cidades por pontos.

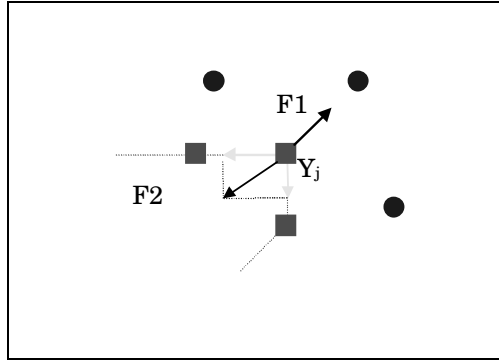


Figura 1 - F1 e F2 são, respectivamente, as forças de atração da cidade e dos vizinhos, sobre o ponto Y_j do anel

3.2.1 Descrição do Algoritmo Elástico

Sejam n cidades do *PCV* representadas por pontos no espaço bidimensional. Sejam m pontos distribuídos em um *anel inicial*, que no algoritmo original, corresponde a uma circunferência, cujo centro coincide com o centróide das cidades. Sejam X_i ($i = 1, \dots, n$) e Y_j ($j = 1, \dots, m$), respectivamente, as coordenadas da cidade i do *PCV* e do ponto j situado sobre o *anel inicial*.

O Algoritmo Elástico deforma gradualmente este *anel inicial*, através da constante atualização de Y_j , obedecendo a uma equação que rege este deslocamento.

$$\Delta Y_j = \alpha \sum_{i=1}^n w_{ij} (X_i - Y_j) + \beta K (Y_{j+1} - 2Y_j + Y_{j-1}) \quad (I)$$

O primeiro termo desta equação representa a força de atração que as cidades exercem sobre o ponto j do anel. Este termo é responsável pela atração do ponto j em direção às cidades. O coeficiente w_{ij} , definido na Equação II, indica a influência que a cidade i exerce sobre o ponto j . Esta influência é função da distância entre o ponto j e a cidade i , e o parâmetro K , responsável, durante toda a evolução do algoritmo, pela maior ou menor deformação do anel. Esse parâmetro possui um comportamento semelhante ao que simula a temperatura no algoritmo “*Simulated Annealing*”[17]. Quanto mais próxima uma cidade estiver do ponto, maior será sua influência sobre ele. Para simular este comportamento utilizou-se uma função exponencial (Equação III, explicada a seguir). O número máximo de pontos distribuídos pelo anel, normalmente corresponde ao dobro de números de cidades. O estabelecimento deste “*upper bound*” é interessante, pois se o número de pontos no anel for pequeno, a flexibilidade na escolha da solução será pequena. Em contrapartida, se este número

for muito grande o tempo de computação aumenta e ainda propiciaremos a ocorrência de possíveis soluções inviáveis. É importante a normalização do coeficiente w_{ij} , para garantir uma distribuição uniforme dos pontos sobre o anel. Assim, tem-se

$$w_{ij} = \frac{\Phi(|X_i - Y_j|, K)}{\sum_{i=1}^n \Phi(|X_i - Y_n|, K)} \quad (\text{II})$$

onde, $\Phi(|X_i - Y_j|, K)$ é uma função positiva, definida como:

$$\Phi(|X_i - Y_j|, K) = e^{-\frac{|X_i - Y_j|^2}{2K^2}} \quad (\text{III})$$

O segundo termo da Equação I de deslocamento representa a força de atração que os vizinhos do ponto j no anel exercem sobre ele. O coeficiente βK é responsável pela intensidade desta atração. É este segundo termo que está relacionado com o comprimento da rota. O parâmetro K é decrementado lentamente durante a execução do algoritmo, logo ao final do processamento ele assumirá um valor muito próximo a zero.

Os parâmetros α e β são responsáveis pelo balanceamento das forças da equação de deslocamento. Assim, é possível ajustar as forças de acordo com as características do problema a ser solucionado.

Para cada valor de K , os pontos do anel sofrem p deslocamentos, sendo p o número de iterações preestabelecido. Na verdade, para cada K o algoritmo calcula os deslocamentos dos pontos do anel até que a função energia E associada ao algoritmo, vista na Equação IV, a seguir, chegue a um mínimo.

$$E(Y_j, K) = -\alpha K \sum_{i=1}^n \ln \sum_{j=1}^m e^{\left(\frac{-|X_i - Y_j|^2}{2K^2}\right)} + \frac{\beta}{2} \sum_{j=1}^m |Y_{j+1} - Y_j|^2 \quad (\text{IV})$$

Essa função possui a propriedade de ser reduzida a cada deslocamento sofrido por cada ponto do anel. Mais formalmente, esta propriedade de redução poderia ser descrita como:

$$\Delta Y_j = -K \frac{\delta E}{\delta Y_j}$$

A função E é limitada inferiormente, logo esta redução ocorre até que a função atinja a um mínimo.

O decréscimo de K propicia a procura de mínimos locais mais próximos do ótimo global. Para uma maior rapidez do algoritmo, o número de iterações para cada K é prefixado.

O pseudocódigo do Algoritmo Elástico é mostrado na Figura 2, onde n representa o número total das cidades, K é o parâmetro do Algoritmo Elástico, x é o número de iterações e m é o número de pontos no anel do elástico.

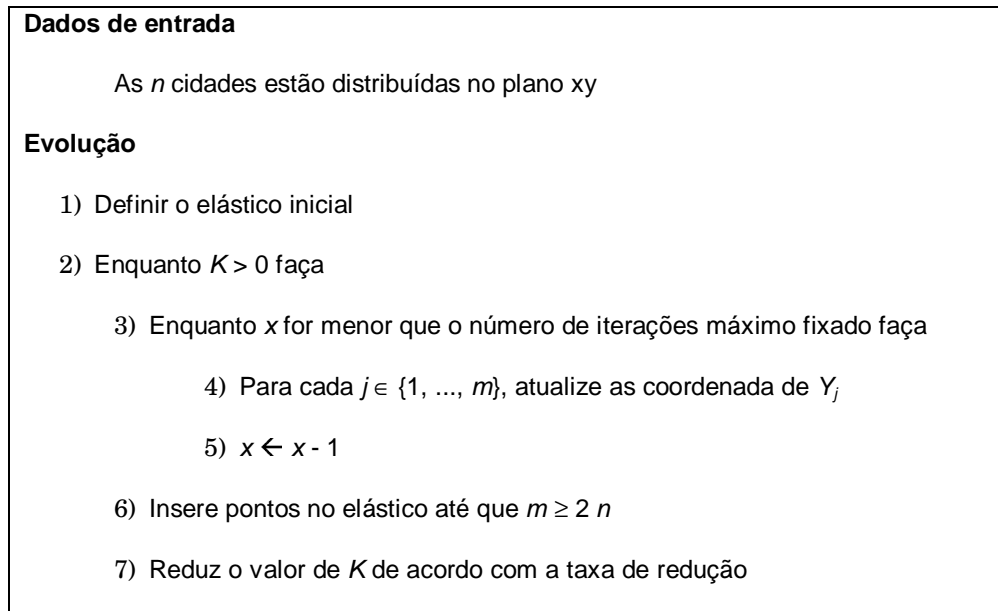


Figura 2 - Pseudocódigo do Algoritmo Elástico

A partir das coordenadas (x,y) das cidades, definimos um anel inicial, que corresponde à circunferência cujo centro coincide com o centróide das cidades, que se deformará de acordo com a equação de deslocamento (definida na Equação I). Um número x iterações é realizado até que novos pontos sejam inseridos no anel e o valor de K seja reduzido (linha 7 da Figura 2). Quando K for menor ou igual a zero, o algoritmo pára e o anel final conterá a solução para o conjunto de cidades.

3.2.2 Modelo Escolhido

O modelo do Algoritmo Elástico a ser utilizado neste trabalho corresponde a uma variação do algoritmo inicialmente proposto por Durbin e Willshaw [8] e descrito na

Figura 2. Nosso *anel inicial*, ao invés de uma circunferência, como no algoritmo de Durbin e Willshaw, é uma envoltória convexa das n cidades. Esta escolha se deve ao fato de que os resultados obtidos em trabalhos realizados por Figueiredo, Ochi e Maculan [27], com esta nova estratégia apresentaram um melhor desempenho.

Os parâmetros α , β e K são ajustados experimentalmente de acordo com cada problema e o número máximo de pontos no anel é atribuído como sendo o dobro do número de cidades.

3.3 Modelo de Pétalas

O Modelo de Pétalas (*Petal Model*) é um método baseado em um processo de varredura com o objetivo de gerar *clusters* (pétalas). Este modelo foi originalmente proposto por Foster e Ryan [10] para solucionar o Problema de Roteamento de Veículos com Frota Homogênea. Posteriormente foi estendido por Ryan, Hjorring e Glover [37]. Renaud *et al.* [34] que propuseram um modelo de pétalas capaz de criar “flores artificiais”, ou seja, *clusters* que se interceptam. Estas “flores artificiais” possibilitam um aumento do espaço de soluções a serem geradas.

Consideramos uma pétala como sendo uma das rotas que compõem uma solução e, analogamente, uma flor como o conjunto das rotas (pétalas) que formam a solução

Na Figura 3 podemos ver exemplo de uma flor (a), conjunto de *clusters* que não se interceptam, e de uma “flor artificial” (b), onde eles se sobrepõem.

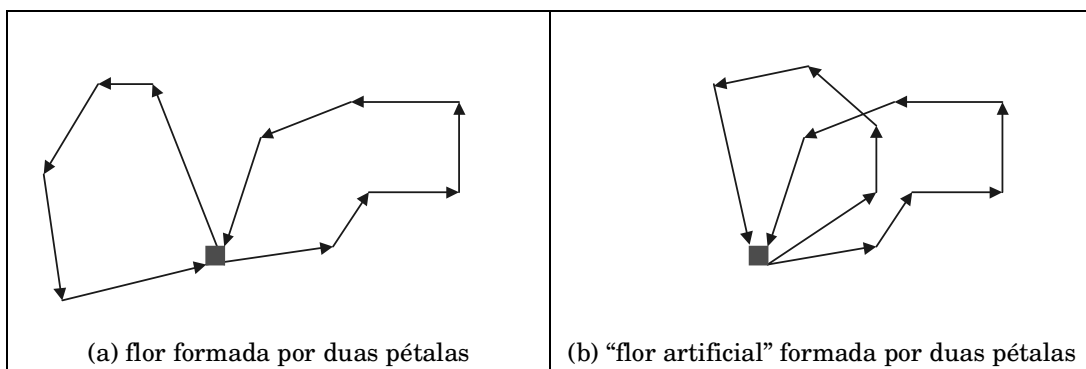


Figura 3 – Divisão em Pétalas

O algoritmo de construção de pétalas é mostrado a seguir.

1. Ordenar as cidades de acordo com suas coordenadas polares, considerando o depósito como sendo o centro do eixo.
2. Escolher, a cidade que se encontrar mais próxima do depósito, como cidade inicial.
3. A partir daí, um eixo imaginário varre o espaço em um determinado sentido (horário ou anti-horário). A cada cidade encontrada, verifica-se se há espaço no caminhão para atendê-la, se houver, esta cidade é inserida na pétala e o processo de busca prossegue.
4. O passo 3 é repetido até que a capacidade do caminhão seja completada (ou quase completada).
5. Uma pétala é fechada e o processo se repete até que todas as cidades tenham sido alocadas.

Cabe fazer uma ressalva de que, como o método é guloso, existem alguns prejuízos inerentes a esta estratégia. Este mecanismo limita o espaço de soluções, algumas vezes até impossibilita que se chegue a uma solução próxima da ótima. Visto que, uma solução ótima pode ser composta de rotas que se interceptam de forma a não ter um aspecto de pétalas.

Neste trabalho o modelo será utilizado como um mecanismo de divisão do problema inicial em *clusters* que estarão associados a uma rota de um veículo no *PRV*. Cada *cluster* representa uma pétala que possui características similares ao Problema do Caixeiro Viajante (*Travelling Salesman Problem*). O conjunto das soluções de todas as pétalas corresponderá à solução procurada para o problema original.

3.4 Busca Tabu

Busca Tabu (“*Taboo Search*” ou “*Tabu Search*”) [11] é uma metaheurística capaz de guiar uma heurística de busca local para explorar além de um ótimo local. A Busca Tabu baseia-se na premissa de que um procedimento inteligente para solucionar um problema tem que incorporar estruturas flexíveis de memória (*adaptive memory*) e critérios de aspiração (*responsive exploration*). Uma boa analogia para entender o procedimento da Busca Tabu é feita com um alpinista. O alpinista tem que lembrar-se de elementos chaves de seu

caminho executado (*adaptive memory*) e tem que ser capaz de realizar escolhas estratégicas ao longo de sua escalada (*responsive exploration*).

As estruturas flexíveis de memória permitem o armazenamento do conhecimento sobre o espaço de busca. O critério de aspiração permitem guiar a busca através de uma estratégia, pois supõe-se que um má estratégia de escolha pode fornecer maiores informações do que um bom processo aleatório.

A Lista Tabu é a estrutura de memória básica onde as soluções proibidas (tabu) são armazenadas. Esta lista é determinada por informações históricas da busca. Uma solução é proibida por um determinado número de iterações (prazo tabu ou *tabu-tenure*)

O critério de aspiração é uma condição em que uma solução torna-se permitida apesar de estar presente na Lista Tabu.

A Busca Tabu parte de uma solução inicial e, a cada iteração, move para a melhor solução na vizinhança. O algoritmo básico é apresentado na Figura 4 a seguir, onde S é a solução a ser avaliada, S_0 é a solução inicial, S^* é a melhor solução encontrada, $N(S)$ é a vizinhança de S e $f(s)$ é o custo da solução S.

```
s ← s0 ;
s* ← s0 ;
lista-tabu ← ∅
  while critério-de-parada do
    for iterações-ciclo do
      Encontrar a melhor solução s' pertencente a N(s) \ lista-tabu
      If f(s') < f(s*) then
        s* ← s'
      else
        if f(s') ≥ f(s) then
          lista-tabu ← lista-tabu ∪ {s}
        end-if
      s ← s'
      Atualizar lista-tabu (prazos)
    end-for
  end-while
```

Figura 4 – Algoritmo Básico de Busca Tabu

A estratégia da Busca Tabu parte de uma solução inicial s_0 . Enquanto o critério de parada não for satisfeito, uma nova solução vizinha à s_0 é encontrada, sendo que esta não pertence à lista tabu. Se esta nova solução for melhor do que a melhor até então (inicialmente s_0), ela passa a ser a solução escolhida, senão se ela for pior ou igual a anterior, ela passa a pertencer à lista tabu. Os prazos da lista tabu são verificados, os prazos vencidos são eliminados da lista, e este ciclo se repete até que um determinado número de iterações seja satisfeito.

Dois importantes componentes da Busca Tabu são as estratégias de intensificação e de diversificação. A intensificação consiste em concentrar a busca em regiões promissoras do espaço, em torno de boas soluções (soluções de elite) e a diversificação possibilita à busca visitar regiões ainda não exploradas.

Entre as características principais da Busca Tabu, se destacam:

- O uso intensivo de estruturas de memórias adaptativas;
- A produção de soluções de boa qualidade, sendo que em alguns casos representam as melhores soluções encontradas para o problema;
- A pequena dependência da qualidade da solução inicial, pois possui mecanismos eficientes capazes de permitir a fuga de ótimos locais
- Ser robusta.

Os conceitos de Lista Tabu, intensificação e diversificação foram utilizados na implementação do segundo algoritmo a ser descrito posteriormente no Capítulo 4.

3.5 O Método 2-Optimal

O Método 2-optimal é uma instância do método genérico k -optimal, quando $k = 2$. Este método foi inicialmente proposto por Lin e Kernighan [17] como um procedimento para solucionar o problema do caixeiro viajante, e consiste em uma estratégia de troca de aresta de forma a transformar um solução inicial T , com custo $f(T)$, em outra solução T' cujo custo é $f(T') < f(T)$.

Seja S o conjunto de todas as arestas entre as n cidades (i.e., $n(n-1)/2$ arestas) e T uma solução possível, porém não ótima para o problema. Podemos dizer que T é não ótima porque existem k elementos x_1, \dots, x_k pertencentes à T que estão “fora de lugar” (*out of place*). Para tornar T uma solução ótima devemos trocar estes k elementos por y_1, \dots, y_k pertencentes a $S-T$ (ver Figura 5). O método k -optimal apresenta um procedimento para determinar o valor de k e de cada x e y , elemento por elemento.

De uma forma simplificada, podemos descrever o método 2-optimal conforme a Figura 5 a seguir, onde G^* representa o maior ganho, t_i um nó, x_i e y_i arestas, T e T' rotas e S um grafo completo.

- 1 - Rota inicial T .
- 2 - $G^*=0$ (G^* = maior ganho até agora).
- 3 - Escolher um nó t_1 e uma aresta x_1 , que une t_1 a um ponto adjacente t_2 , em T .
- 4 – Repetir
 - 5 - A partir de t_2 , escolher uma aresta y_1 em $S-T$ que une t_2 a um ponto adjacente t_3 em $S-T$.
 - 6 – Repetir
 - 7 – Escolher x_2 , em T , que une t_3 a um ponto adjacente t_4 .
 - 8 – Repetir
 - 9 - Escolher y_2 em $S-T$, onde y_2 é alguma aresta disponível que liga t_4 a um ponto de x_1 , y_2 é diferente de y_1 e o ganho $G = x_1 + x_2 - y_1 - y_2$ é positivo.
 - 10 - Se $G > G^*$
 - 11 - $G^* = G$.
 - 12 – T' é a rota resultante da troca de x_1, x_2 por y_1, y_2
 - 12 - Até que todas as arestas y_2 possíveis tenham sido escolhidas.
 - 13 - Até que todas as arestas x_2 possíveis tenham sido escolhidas.
 - 14 - Até que todas as arestas y_1 possíveis tenham sido escolhidas.
- 15 - Se $G^* > 0$
 - 16 - $T = T'$.
 - 17 – Iniciar o processo de 1.
- 18 – Senão
 - 19 – Escolher outro t_1 , não anteriormente selecionado.
 - 20 - Se existir t_1 a ser escolhido
 - 22 - O processo continua de 1.
 - 23 – Senão
 - 24 - O processo termina.

Figura 5 – Pseudocódigo do Método 2-optimal

Na Figura 6 a seguir podemos ver representada uma rota T , o grafo completo S e o grafo resultante $S-T$. Podemos ainda ver um exemplo da aplicação do método, onde:

- $t_1 = 1, t_2 = 4, x_1$ é a aresta 1-4 em T ;
- $t_3 = 5$ e y_1 é a aresta 4-5, em $S-T$;
- $t_4=2$ e x_2 é a aresta 5-2 em T ;
- y_2 é a aresta 2-1 em $S-T$.

O grafo resultante da troca de x_1 e x_2 por y_1 e y_2 é representado por T' .

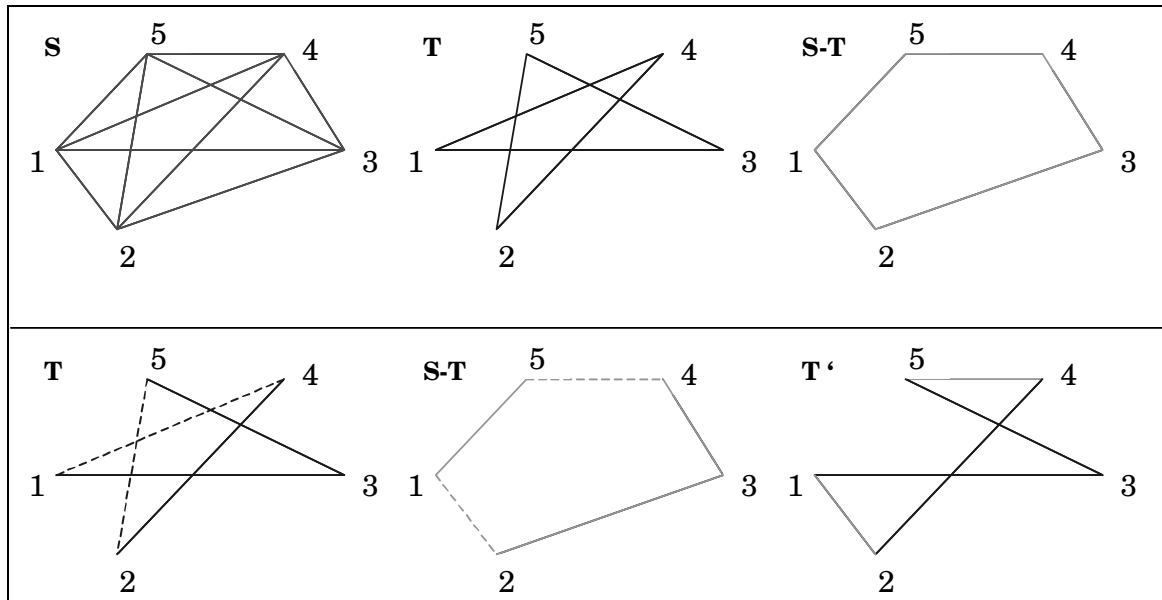


Figura 6 – Exemplificação do método 2-optimal

Escolhemos o Método 2-optimal para ser utilizado como um mecanismo de melhoria da solução final encontrada por novos algoritmos, já que este é um método de otimização aplicado em grafos.

Capítulo 4

Algoritmos Propostos

4.1 Introdução

A idéia inicial foi utilizar a combinação do Modelo de Pétalas e do Algoritmo Elástico para elaborar um algoritmo metaheurístico capaz de solucionar o problema de roteamento de veículos. Surgiu assim o Algoritmo Pétala-Elástico. Variações para este algoritmo também foram implementadas. Um segundo algoritmo também foi elaborado na tentativa de encontrar soluções melhores que o primeiro. Nesta idéia inicial foram incorporados conceitos de Busca Tabu. Este algoritmo foi chamado de Algoritmo Diversificação-Intensificação. Os algoritmos serão apresentados e discutidos mais detalhadamente neste Capítulo.

4.2 O Algoritmo Pétala-Elástico

Este algoritmo foi implementado com o objetivo de verificar a viabilidade da utilização do algoritmo elástico para a solução de problemas de roteamento de veículos. O Método das Pétalas foi utilizado como um mecanismo de subdivisão do problema de roteamento em subproblemas, onde a cada subproblema encontra-se associado um subconjunto de cidades e um caminhão. O conjunto de todos os subproblemas representa o problema inicialmente proposto. Para solucionar cada subproblema foi utilizado o algoritmo elástico.

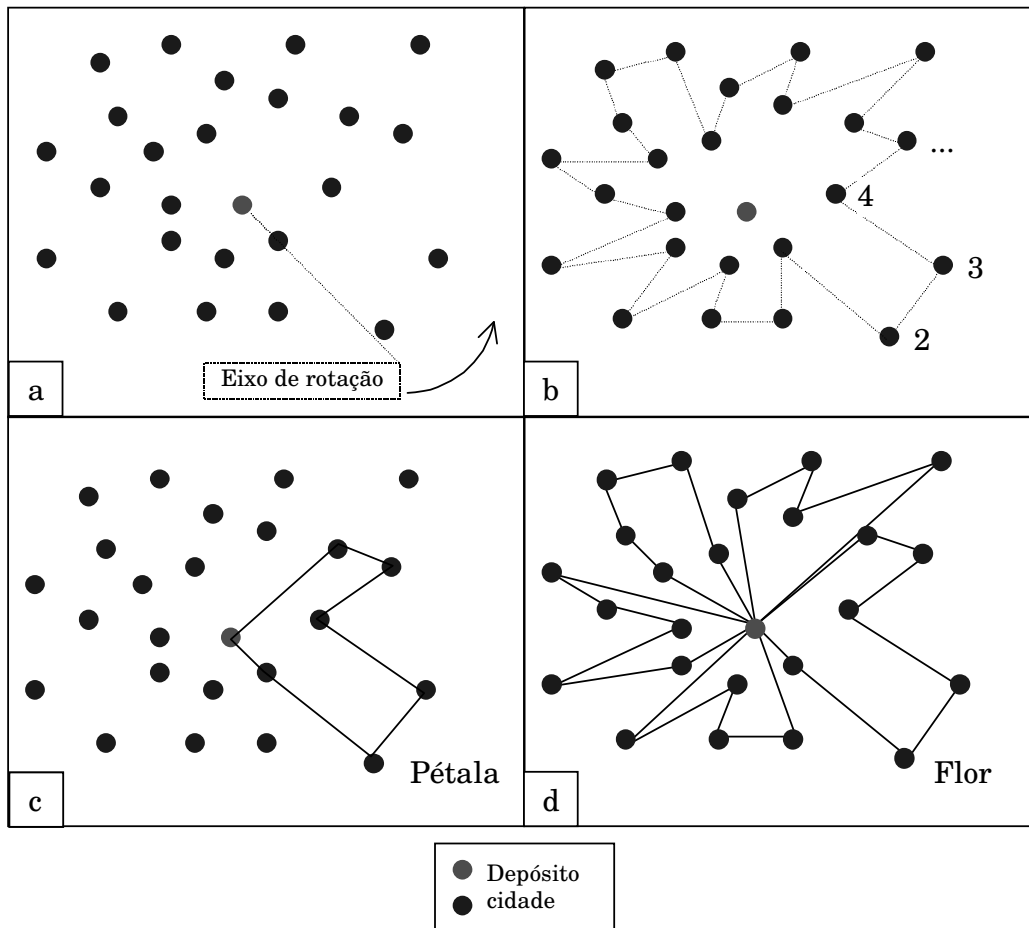


Figura 7 – Formação das Pétalas

Na Figura 7, podemos visualizar o processo de ordenação das cidades, que se inicia com a escolha de uma cidade inicial e a determinação do eixo de rotação no sentido anti-horário (Figura 7a). A partir da rotação do eixo, a ordenação das cidades é determinada de acordo com suas coordenadas polares (Figura 7b). Para a formação das pétalas, escolhemos uma cidade inicial e a partir da ordenação anterior, escolhemos uma segunda cidade, uma terceira, e quantas cidades forem necessárias para a formação de uma pétala (Figura 7c). O que determina a quantidade de cidades em uma pétala são as restrições do problema tratado, que no nosso caso, a soma das demandas das cidades não pode ultrapassar à capacidade do caminhão escolhido para a rota. O conjunto de todas as pétalas geradas representa uma flor (Figura 7d). Cada subproblema será referenciado como uma pétala e o conjunto das pétalas, que formam um problema, como uma flor.

Para a subdivisão do problema em pétalas, é necessário escolher qual caminhão será responsável por um subconjunto de cidades. Essa escolha é feita através da utilização de uma estratégia gulosa, onde é escolhido aquele caminhão que obtiver um menor custo de

abastecimento por unidade. Para determinar este custo de abastecimento, somamos o custo fixo do caminhão associado à rota a uma penalização que chamamos de *custo-desperdício*, definido como sendo o custo relativo à quantidade de unidades que um caminhão deixa de transportar. O cálculo do desperdício, sua conversão em custo e o cálculo do custo final são descritos mais detalhadamente a seguir.

O custo do transporte por unidade em uma rota é o resultado da divisão da capacidade total do caminhão pelo seu custo fixo de utilização. Obtemos assim quanto custa transportar uma unidade com este caminhão.

$$\boxed{\text{custo_por_unidade} = \left[\frac{\text{capacidade_caminhão}}{\text{custo_fixo_caminhão}} \right]} \quad (\text{ I })$$

O desperdício é o resultado da subtração entre a capacidade total do caminhão e a soma das demandas das cidades (unidades efetivamente transportadas por este caminhão neste conjunto de cidades). Obtemos assim, quantas unidades deixam de ser transportadas por este caminhão para este conjunto de cidades

$$\text{desperdício} = \text{capacidade_do_caminhão} - \text{demanda_das_cidades} \quad (\text{ II })$$

Para converter este desperdício em custo, multiplicamos o custo do transporte de cada unidade (descrito na Equação I) pela quantidade de unidades que deixam de ser transportadas. Este custo será utilizado como uma forma de penalização ao caminhão que não utilizar sua capacidade total de transporte.

$$\boxed{\text{custo_desperdício} = \text{custo_por_unidade} \times \text{desperdício}} \quad (\text{ III })$$

O custo final da utilização de um caminhão em um determinado conjunto de cidades é a soma do custo fixo associado ao caminhão com o custo do desperdício, calculado anteriormente.

$$\boxed{\text{custo_final} = \text{custo_desperdício} + \text{custo_fixo_caminhão}} \quad (\text{ IV })$$

Para cada problema é gerado um conjunto de flores, onde a cada flor está associado um custo fixo, que é a soma dos custos fixos de todas as suas pétalas. O custo fixo de uma pétala é o valor do custo do caminhão c responsável por sua rota. O custo final de

uma flor é a soma do custo fixo desta flor mais a soma dos custos das rotas de cada pétala. O custo da rota de uma pétala é obtido através do algoritmo elástico.

A construção das flores associadas a um problema é apresentada pelo pseudocódigo a seguir, onde F é o conjunto das flores geradas, f_i uma flor, e i o número de flores geradas;

1. $F = \emptyset$ { F = conjunto de flores}
2. Escolhe uma cidade inicial;
3. $i = 0$
- 4. Repetir**
 5. Aplicar o método das pétalas à partir da cidade inicial escolhida gerando assim a flor f_i .
 6. $F = F \cup \{f_i\}$
 7. $i = i + 1$
 8. Escolher uma nova cidade inicial, não anteriormente escolhida;
- 9. até que todas as cidades tenham sido cidades iniciais.**

Figura 8 - Construção das flores

Escolhe-se uma cidade como sendo cidade inicial (linha 2, Figura 8) e aplica-se o método de divisão por pétalas até que a cidade final seja atingida (linha 5, Figura 8). Escolhe-se uma nova cidade inicial, não anteriormente escolhida como tal (linha 8, Figura 8). O mesmo processo é repetido até que todas as cidades tenham sido cidades iniciais (linha 9, Figura 8).

A partir do conjunto de flores gerados, foram desenvolvidas três versões para o algoritmo. Na primeira versão o conjunto das flores é ordenado de forma crescente em relação ao seu custo fixo. Seleciona-se 10% das flores cujos custos fixos são os menores. A segunda versão ordena de forma crescente em relação ao desperdício total de cada flor, onde o desperdício total de uma flor é a soma dos desperdícios de todas as suas pétalas. Seleciona

10% das flores cujos desperdícios são os menores. Finalmente uma terceira versão, escolhe a flor que possuir o menor custo fixo.

Para todas as versões o que se faz a seguir é solucionar a rota de cada pétala e calcular o custo final de cada flor. A solução de cada rota é feita através da utilização do Algoritmo Elástico. O ajustes dos parâmetros do Algoritmo Elástico foram feitos através de testes.

Podemos resumir o algoritmo Pétala-Elástico com o seguinte pseudo-código:

```
1 — Leitura dos arquivos de dados:
    cidade[i] = (xi, yi, demanda) ;
    caminhao[i] = (capacidade, custo_fixo);

2 — Ordenação das cidades

3 — Geração das flores.

4 — Aplicar o método de seleção da flores de acordo com cada versão.

5 — Para cada flor selecionada faça:
    5.1 — Para cada pétala, é aplicado o algoritmo elástico:
        i. O elástico inicial é a cobertura convexa das cidades;
        ii. Os parâmetros do algoritmo elástico são definidos;
            K, α, β, número de iterações e a taxa de redução de K;
        iii. Para cada ponto da rede elástica é calculada a fórmula do
            deslocamento e suas coordenadas são atualizadas, até que o
            número de iterações tenha sido atingido ou não houver melhora
            na função de deslocamento;
        iv. Novos pontos são inserido na rede elástica e o valor de k é
            atualizado de acordo com sua taxa de redução;
        v. Se K > 0, retorna-se ao passo iii.
    5.2 Armazenar a flor com menor custo final.

6 - Retornar a flor com o menor custo final entre todas as flores
selecionadas.
```

Figura 9 - Pseudocódigo Algoritmo Pétala-Elástico

A partir das cidades e dos caminhões associados à um problema, utiliza-se o método das pétalas para gerar todas as flores associadas a este problema. Aplica-se então um

método de seleção desta flores (diferente para cada versão do algoritmo como descrito anteriormente). Para cada flor selecionada, em todas as suas pétalas é aplicado o Algoritmo Elástico para encontrar a rota e seu custo correspondente. No final, a flor que possuir o menor custo é a solução encontrada pelo algoritmo para o problema proposto.

4.3 O Algoritmo Diversificação-Intensificação

Este algoritmo foi proposto com o objetivo de encontrar soluções melhores do que o Algoritmo Pétala-Elástico. A principal idéia é gerar rotas que possibilitem a formação de flores artificiais (vide Figura 9). A partir de um conjunto de pétalas, geramos através de um processo de diversificação novas “pétalas artificiais”. E da combinação entre as pétalas tentamos construir soluções viáveis ao problema proposto. Apresentaremos a seguir três variações deste algoritmo.

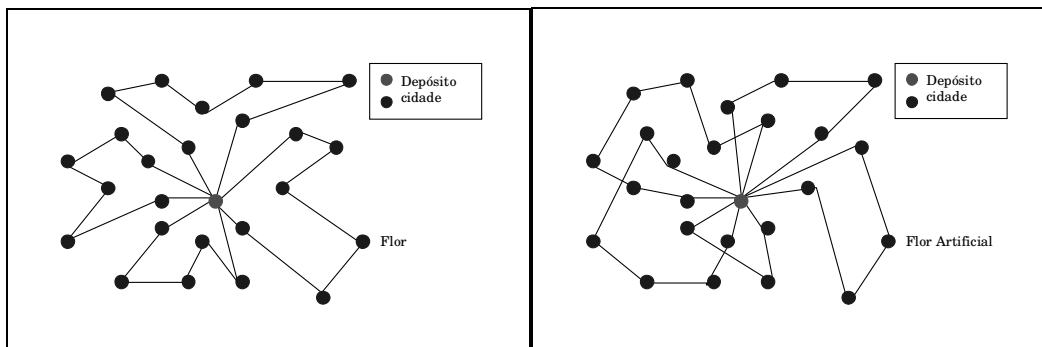


Figura 10 - Soluções em formato de Flor e Flor Artificial

Inicialmente utilizamos o Método de Pétalas para gerar rotas. Utilizou-se para as duas primeiras versões o critério de que um melhor caminhão para uma rota é aquele que apresentar um menor desperdício, igualmente ao Algoritmo Pétala-Elástico, descrito anteriormente no Capítulo 3. Já na terceira versão, nenhum critério de seleção de caminhões foi adotado, uma vez que para cada caminhão foram geradas todas rotas possíveis. Passamos a ter assim uma “população” de rotas. Uma estratégia foi proposta para a partir deste conjunto de rotas, gerarmos novas rotas. Inicialmente, duas rotas são selecionadas aleatoriamente. Em cada rota, também aleatoriamente, uma cidade é selecionada. Se estas duas cidades forem consideradas próximas (distância menor do que a distância média das cidades) elas passam a ser candidatas à permutação. Se esta troca não tiver sido recentemente realizada (presente na

lista tabu) e se não resultar em soluções inviáveis (capacidade dos caminhões excedidas), duas novas rotas são geradas e inseridas na população.

Utilizamos a idéia da Lista Tabu, descrita no Capítulo 2, para armazenar as trocas mais recentes e evitar repetições, ou seja, em uma lista armazenamos as últimas trocas, a quantidade de trocas que é armazenada para cada problema é um parâmetro que deve ser determinado experimentalmente. As trocas são realizadas até que um número máximo de iterações seja atingido. Este número de iterações também é atribuído experimentalmente.

Neste estágio uma população heterogênea de rotas é estabelecida. A partir desta nova população, a cada rota é aplicado o algoritmo elástico para encontrar uma solução (seqüência de cidades) e um custo associado. Inserimos todas as rota em um procedimento de escolha conhecido como “*roleta*” [21], onde uma rota de melhor qualidade (menor custo associado) possui maior probabilidade de ser escolhida do que uma de qualidade inferior. Uma rota é então selecionada e todas as outras rotas que possuem cidades coincidentes com as da rota escolhida são eliminadas da população de rotas. Este procedimento de escolha é realizado até que não existam mais rotas na população. Se ao final alguma cidade ficar descoberta, ou seja, não pertencer a nenhuma das rotas selecionadas, utilizamos o algoritmo pétala-elástico para gerar estas rotas complementares, que não são inseridas na população de rotas para um uso posterior, evitando assim um aumento exagerado da população de rotas. Esta solução é armazenada e a população inicial é restabelecida. O procedimento é repetido para a formulação de outras soluções até que em um número de iterações definidos experimentalmente, não haja melhora na solução encontrada. Assim uma solução para o problema é encontrada. Na segunda e terceira versões foi aplicado o método 2-optimal para tentar melhorar esta solução final.

O procedimento de diversificação consiste na troca entre cidades de rotas diferentes e o de intensificação consiste em uma mesma população tentar extrair soluções diferenciadas até que não se encontre uma solução melhor.

Este nosso algoritmo pode ser descrito mais formalmente com o seguinte pseudocódigo, onde T e T' representam populações de rotas, S uma solução, S^* a melhor solução, x o número de iterações, t uma rota e $f(S)$ o custo da solução S :

- 1 — Leitura dos arquivos de dados:
 - $cidade[i] = (x_i, y_i, demanda)$;
 - $caminhão[i] = (capacidade, custo_fixo)$;
- 2 — Ordenação das cidades de acordo com o ângulo formado com o depósito.
- 3 — Utilização do Modelo de Pétalas para gerar todas as rotas possíveis para este conjunto de cidades e de caminhões.
- 4 — Procedimento de diversificação:
 - 4.1 — Faça até que o número de iterações seja máximo:
 - 4.1.2 — Escolher aleatoriamente duas rotas.
 - 4.1.3 — Escolher aleatoriamente uma cidade em cada rota.
 - 4.1.4 — Se a distância entre as cidades for menor que a distância média
 - 4.1.5 — Se a troca não foi recentemente realizada.
 - 4.1.5.1 — Realiza a troca gerando duas novas rotas
 - 4.1.5.2 — Insere as rotas na população.
- 5 — Procedimento de intensificação.
 - 5.1 — $T =$ população de rotas.
 - 5.2 — $S^* = \emptyset$.
 - 5.3 — Repetir até que em x iterações a solução final não seja melhorada.
 - 5.3.1 — $T' = T$.
 - 5.3.2 — $S = \emptyset$.
 - 5.3.3 — Repetir até que $T' = \emptyset$
 - 5.3.3.1 — Escolher, utilizando o critério da roleta russa, uma $tour t \in T'$.
 - 5.3.3.2 — $S = S \cup t$.
 - 5.3.3.3 — Remover de T' todas as rotas que possuem cidades coincidentes com as cidades de t .
 - 5.3.4 — Se houver alguma cidade descoberta, utiliza o algoritmo pétala-elástico para encontra as rotas complementares.
 - 5.3.5 — Se ($S^* = \emptyset$ ou $f(S) < f(S^*)$) então $S^* = S$
6. — S^* é a melhor solução encontrada

Figura 11 - Pseudocódigo Algoritmo Diversificação-Intensificação

Numa terceira versão, não foi utilizado nenhum critério para a escolha do caminhão associado a uma rota. Ao invés disto, geramos todas as rotas possíveis para todos os caminhões, ocasionando assim em uma maior “população” de rotas.

Capítulo 5

Resultados Obtidos

6.1 Introdução

Apresentaremos neste capítulo uma análise dos resultados obtidos a partir de testes realizados com os dois algoritmos implementados e suas variações, utilizando como instância do Problema de Roteamento de Veículos com Frota Heterogênea o conjunto de teste propostos por Golden et al. [15]. Este teste foram utilizados por Taillard que propôs um algoritmo [40] (vide Anexo 1) que servirá como referência para a comparação dos nossos resultados obtidos. Este algoritmo apresenta até então os melhores resultados conhecidos para o Problema de Roteamento de Veículos com Frota Heterogênea, sendo assim considerado um método robusto e eficiente.

Os problemas testes são descritos a seguir.

6.2 Os Problemas Teste

Na Tabela 1 a seguir estão descritas as características dos problemas testes utilizados para medir a eficiência e eficácia dos algoritmo apresentados.

Problema	n	A		B		C		D		E		F	
		Q_A	F_A	Q_B	f_B	Q_C	f_C	Q_D	f_D	Q_E	f_E	Q_F	F_F
13	50	20	20	30	35	40	50	70	120	120	225	200	400
14	50	120	100	160	1500	300	3500						
15	50	50	100	100	250	160	450						
16	50	40	100	80	200	140	400						
17	75	50	25	120	80	200	150	350	320				
18	75	20	10	50	35	100	100	150	180	250	400	400	800
19	100	100	500	200	1200	300	2100						
20	100	60	100	140	300	200	500						

Tabela 1 – Problemas de Teste

Na tabela anterior, n representa o número de cidades do problema e as letras **A** até **F** representam os caminhões da frota, onde Q_i é a capacidade de transporte do caminhão i e f_i é o custo fixo de utilização do caminhão i , $i \in \{A, \dots, F\}$.

6.3 A Implementação

Todos os algoritmos foram implementados em ANSI C, em ambiente operacional UNIX.

6.3.1 Os Parâmetros do Algoritmo Elástico

Os parâmetros do algoritmo elástico foram ajustados experimentalmente de acordo com cada conjunto de problemas (50, 75 e 100 cidades). Uma vez ajustados, os mesmos foram utilizados em todas as versões de todos os algoritmos implementados.

Observou-se no Algoritmo Diversificação-Intensificação a ocorrência de rotas cujo elástico não conseguiu encontrar a solução, ou seja, gerou soluções inviáveis que não continham todas as cidades. Mas como essas ocorrências foram escassas, uma ou duas rotas em cada execução do algoritmo para uma determinada instância num universo de mais de 3.000 rotas solucionadas, não houve uma preocupação em realizar novos ajustes destes parâmetros. A solução adotada foi remover estas rotas da instância em execução. Estas rotas

possuem características próprias, uma vez que foram geradas através do processo de diversificação, apresentando assim uma distribuição não uniforme, onde algumas cidades se encontravam isoladas de um grupo maior totalmente concentrado.

Durante o processo de ajuste dos parâmetros do Algoritmo Elástico (Tabela 2) uma ferramenta gráfica foi implementada juntamente com os algoritmos propostos viabilizando a visualização de todo o processo evolutivo do anel inicial até a convergência para uma solução final. A Figura 12 a seguir, apresenta uma das imagens geradas por esta implementação:

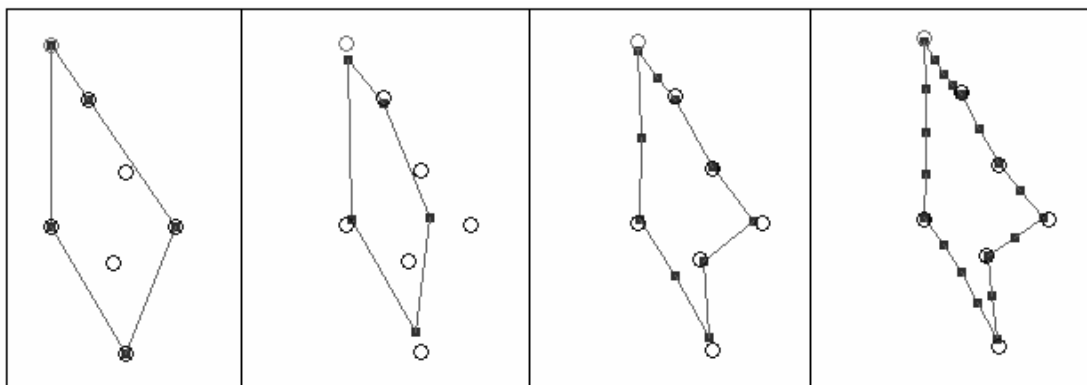


Figura 12 – Evolução do Algoritmo Elástico

50 CIDADES	75 CIDADES	100 CIDADES
$K = 0,35$	$K = 0,35$	$K = 0,35$
$\alpha = 0,25$	$\alpha = 0,4$	$\alpha = 0,6$
$\beta = 0,1$	$\beta = 0,05$	$\beta = 0,05$
Iterações = 10	Iterações = 10	Iterações = 10
Taxa de redução = 10%	Taxa de redução = 10%	Taxa de redução = 10%

Tabela 2 – Parâmetros do Algoritmo Elástico

6.4 Apresentação dos resultados

Serão apresentados os resultados encontrados pelas implementações do Algoritmo Pétala-Elástico e Algoritmo Diversificação-Intensificação.

6.4.1 Algoritmo Pétala-Elástico

Na Tabela 3 e no Gráfico 1 a seguir, os resultados encontrados pelas três versões do Algoritmo Pétala-Elástico são apresentadas.

<i>Problema</i>	<i>Versão 1</i>	<i>Versão 2</i>	<i>Versão 3</i>	<i>Taillard</i>	<i>Diferença</i>	
13	<u>2127,01</u>	2251,47	2155,43	2413,78	-286,77	-11,88%
14	<u>1321,40</u>	1348,41	1348,41	9119,03	-7797,63	-85,51%
15	<u>2734,98</u>	2824,25	2757,64	<u>2586,37</u>	148,61	5,75%
16	<u>2969,35</u>	3039,24	2975,17	<u>2741,50</u>	227,85	8,31%
17	<u>1975,14</u>	1987,44	2112,79	<u>1747,24</u>	227,9	13,04%
18	<u>2961,12</u>	3269,94	3015,85	<u>2373,63</u>	587,49	24,75%
19	<u>9732,9</u>	9747,43	9785,36	<u>8661,81</u>	1071,09	12,37%
20	<u>4430,28</u>	4769,20	4580,13	<u>4047,55</u>	382,73	9,46%

Tabela 3 – Resultados do Algoritmo Pétala-Elástico

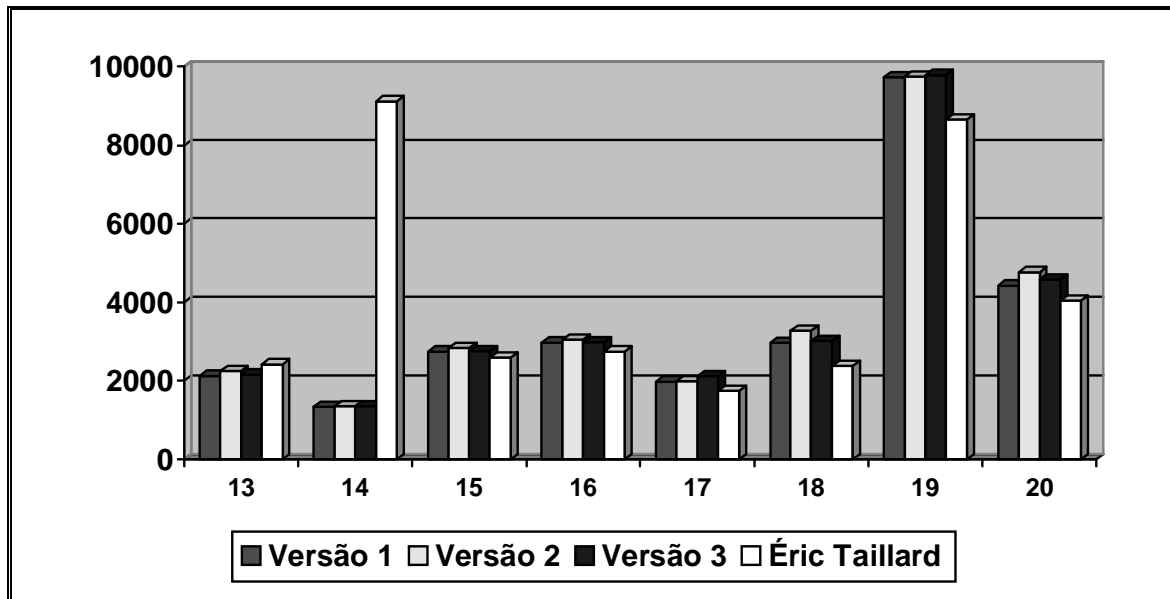


Gráfico 1 – Resultados do Algoritmo Pétala-Elástico

Podemos observar nos dados tabulados acima, que a primeira versão apresentou melhores resultados em todos os problemas de teste. A segunda e terceira versões se alternaram entre as piores soluções, nenhuma apresentou um comportamento estritamente pior ou melhor que a outra.

Em 25% dos casos o algoritmo apresentou soluções melhores do que a encontrada pelo Método de Geração de Colunas. Nas 75% restantes os resultados encontrados foram em média 12,28% piores.

6.4.1.1 Análise das melhores soluções

Realizou-se uma análise da composição dos custos nas soluções finais, com o objetivo de definir uma boa estratégia em busca de melhores soluções.

A seguir na Tabela 4 e Gráfico 2 apresentam esta análise.

Problema	Custo Total	Custo Tour (%)	Custo Caminhão (%)
13	2127,01	31,83%	68,17%
14	1321,40	47,03%	52,97%
15	2734,98	30,53%	69,47%
16	2969,35	29,28%	70,72%
17	1975,14	51,14%	48,86%
18	2961,12	34,82%	65,18%
19	9732,9	10,61%	89,39%
20	4430,28	27,77%	72,23%

Tabela 4 – Análise dos custos das soluções finais

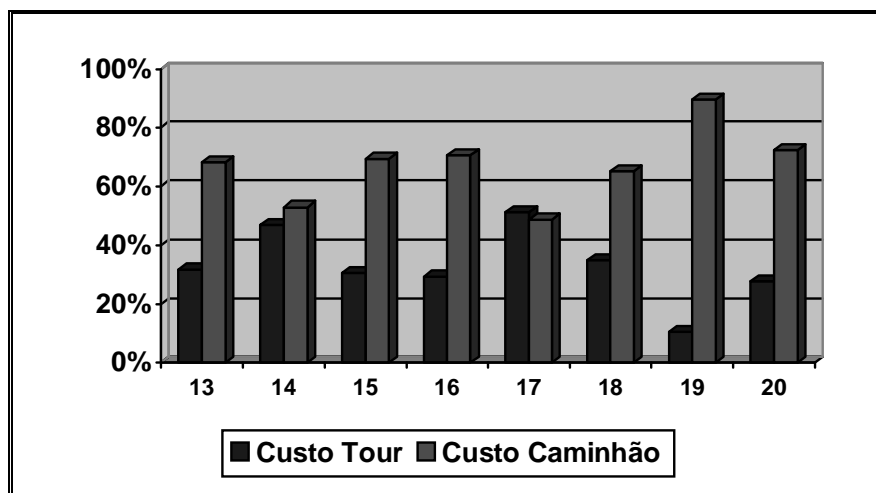


Gráfico 2 – Análise dos custos das soluções finais

As melhores soluções para os problemas propostos apresentaram uma distribuição não balanceada em relação à formação do custo final. Em média o custo associado a utilização dos caminhões equivale a 67,12% do valor total.

Esta predominância do custo do caminhão sobre o custo da rota não era esperada, uma vez que nossa estratégia inicial manteve seu foco principal na otimização do custo das rotas. Sabendo disto, o algoritmo diversificação-intensificação visou suprir esta deficiência inicial.

Alguns teste foram realizados forçando a utilização dos caminhões de menores custos, porém nenhum resultado satisfatório foi obtido. A combinação dos caminhões não obedece a regra de que se usarmos sempre os mais baratos, teremos um custo final menor.

6.4.2 Algoritmo Diversificação-Intensificação

A seguir, os resultados encontrados pelas três versões do Algoritmo Diversificação-Intensificação são apresentados, onde Melhor Solução é a melhor solução encontrada pelo algoritmo e Taillard é a melhor solução encontrada pelo algoritmo proposto por Éric Taillard[40] (vide Anexo 1).

<i>Problema</i>	<i>Melhor Solução</i>	<i>Taillard</i>	<i>Diferença</i>	
<i>13</i>	<u>2002,94</u>	2413,78	-410,84	-17,02%
<i>14</i>	<u>1303,42</u>	9119,03	-7815,61	-85,71%
<i>15</i>	2681,62	<u>2536,37</u>	145,25	5,73%
<i>16</i>	2866,89	<u>2741,5</u>	125,39	4,57%
<i>17</i>	1871,73	<u>1747,24</u>	124,49	7,12%
<i>18</i>	2513,68	<u>2373,63</u>	140,05	5,90%
<i>19</i>	8892,58	<u>8661,81</u>	230,77	2,66%
<i>20</i>	4402,27	<u>4047,55</u>	354,72	8,76%

Tabela 5 – Resultados do Algoritmo Diversificação-Intensificação 1ª Versão

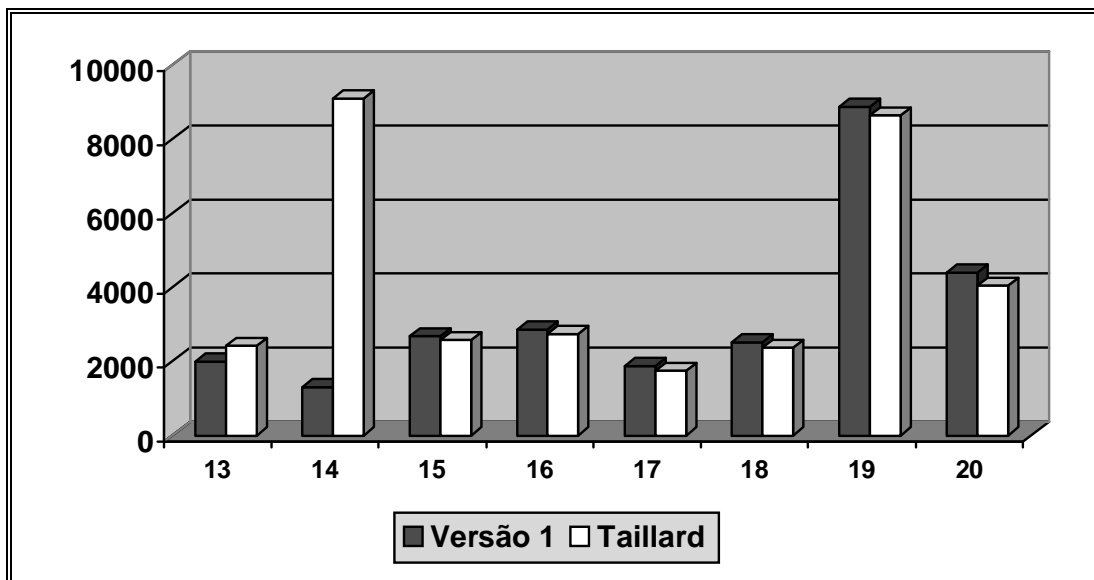


Gráfico 3 - Resultados do Algoritmo Diversificação-Intensificação 1ª Versão

Esta primeira versão do algoritmo apresentou resultados melhores do que os obtidos pelo algoritmo pétala-elástico, porém não conseguiu alcançar, nem melhorar, as melhores soluções conhecidas para o problema.

Encontramos em 25% dos casos soluções melhores do que as conhecidas para o problema. Os 75% restantes foram em média 5,79% piores.

<i>Problema</i>	<i>Melhor Solução</i>	<i>Taillard</i>	<i>Diferença</i>	
<i>13</i>	<u>1990,01</u>	2413,78	-423,77	-17,56%
<i>14</i>	<u>1330,57</u>	9119,03	-7788,46	-85,41%
<i>15</i>	2687,49	<u>2536,37</u>	151,12	5,96%
<i>16</i>	2861,08	<u>2741,5</u>	119,58	4,36%
<i>17</i>	1948,25	<u>1747,24</u>	201,01	11,50%
<i>18</i>	2478,97	<u>2373,63</u>	105,34	4,44%
<i>19</i>	8999,9	<u>8661,81</u>	338,09	3,90%
<i>20</i>	4381,59	<u>4047,55</u>	334,04	8,25%

Tabela 6 - Resultados do Algoritmo Diversificação-Intensificação 2ª Versão

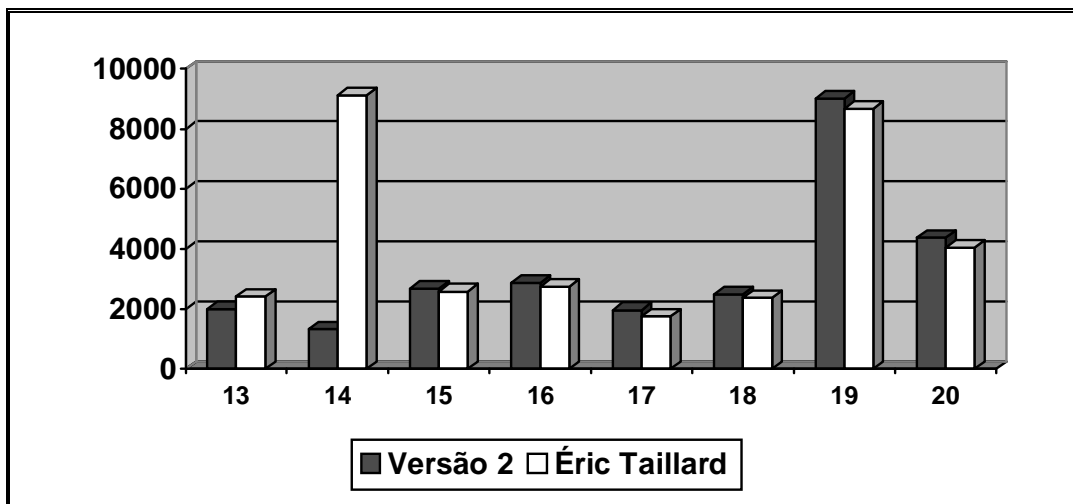


Gráfico 4 - Resultados do Algoritmo Diversificação-Intensificação 2ª Versão

Esta segunda versão diferencia-se da primeira apenas no método 2-optimal aplicado nas melhores soluções encontradas. Apesar deste método de otimização ter sido aplicado, os resultados encontrados não foram necessariamente melhores do que os da versão anterior. Ou seja, a qualidade das rotas geradas não foram substancialmente melhoradas.

Em 75% dos casos as soluções encontradas foram 4,80% piores do que as melhores conhecidas para o problema.

<i>Problema</i>	<i>Melhor Solução</i>	<i>Taillard</i>	<i>Diferença</i>	
<i>13</i>	<u>1676,14</u>	2413,78	-737,64	-30,56%
<i>14</i>	<u>1306,24</u>	9119,03	-7812,79	-85,68%
<i>15</i>	<u>2406,84</u>	2536,37	-129,53	-5,11%
<i>16</i>	<u>2639,52</u>	2741,5	-101,98	-3,72%
<i>17</i>	1928,77	<u>1747,24</u>	181,53	10,39%
<i>18</i>	<u>2187,85</u>	2373,63	-185,78	-7,83%
<i>19</i>	9486,57	<u>8661,81</u>	824,76	9,52%
<i>20</i>	4421,21	<u>4047,55</u>	373,66	9,23%

Tabela 7 - Resultados do Algoritmo Diversificação-Intensificação 3ª Versão

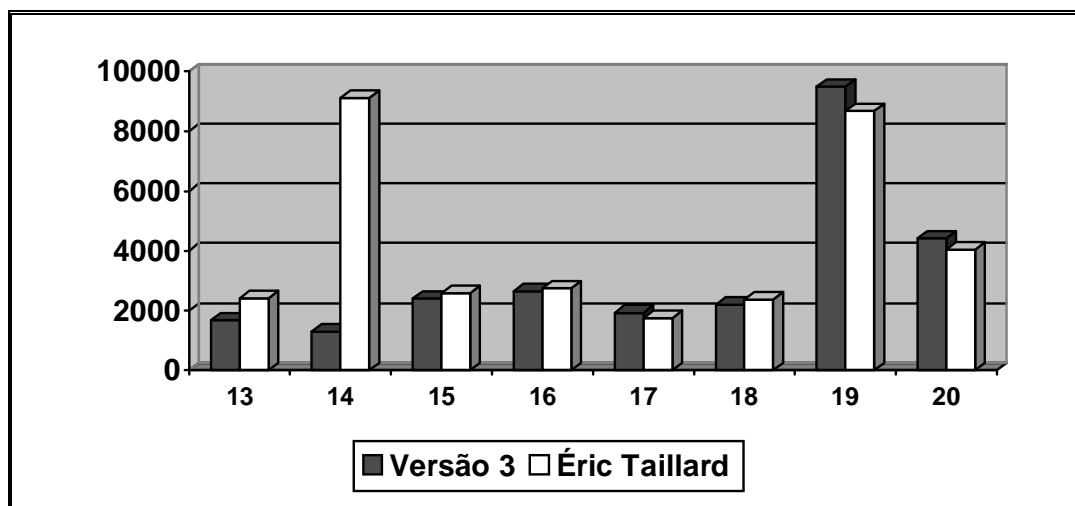


Gráfico 5 - Resultados do Algoritmo Diversificação-Intensificação 3ª Versão

Os resultados da terceira versão superaram os obtidos por todos os algoritmos anteriores.

Na Tabela 8 e Gráfico 6, a seguir, podemos ver um resumo dos resultados encontrados por todas as versões do Algoritmo Diversificação-Intensificação.

<i>Problema</i>	<i>Versão 1</i>	<i>Versão 2</i>	<i>Versão 3</i>	<i>Éric Taillard</i>	<i>Diferença</i>	
<i>13</i>	2002,94	1990,01	<u>1676,14</u>	2413,78	-737,64	-30,56%
<i>14</i>	<u>1303,42</u>	1330,57	1306,24	9119,03	-7815,61	-85,71%
<i>15</i>	2681,62	2687,49	<u>2406,84</u>	2536,37	-129,53	-5,11%
<i>16</i>	2866,89	2861,08	<u>2639,52</u>	2741,50	-101,98	-3,72%
<i>17</i>	<u>1871,73</u>	1948,25	1928,77	<u>1747,24</u>	124,49	7,12%
<i>18</i>	2513,68	2478,97	<u>2187,85</u>	2373,63	-185,78	-7,83%
<i>19</i>	<u>8892,58</u>	8999,9	9486,57	<u>8661,81</u>	230,77	2,66%
<i>20</i>	4402,27	<u>4381,59</u>	4421,21	<u>4047,55</u>	334,04	8,25%

Tabela 8 - Resumo dos resultados do Algoritmo Diversificação-Intensificação

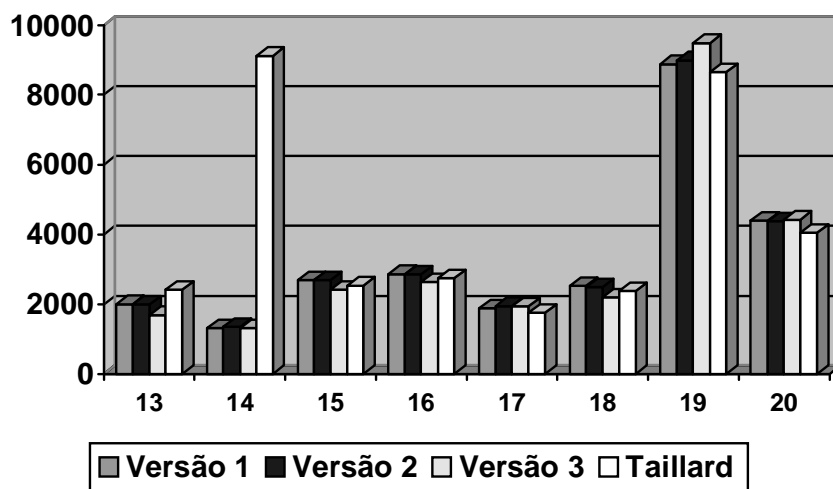


Gráfico 6 - Resumo dos resultados do Algoritmo Diversificação-Intensificação

Como já visto anteriormente, a versão 3 do algoritmo se mostrou mais eficiente do que a versão 1 e 2. Em 62,5% dos testes ela conseguiu superar e estabelecer soluções melhores do que as anteriormente conhecidas para o problema. As soluções de menor qualidade foram em média 6,03% piores.

6.4.3 Resumo dos resultados encontrados

Na Tabela 9 e Gráfico 7, um resumo de todos os resultados encontrados por todas as versões, de todos os algoritmos implementados são mostrados.

Problema	Pétala-Elástico			Diversificação-Intensificação			Taillard
	Versão 1	Versão 2	Versão 3	Versão 1	Versão 2	Versão 3	
13	2127,01	2251,47	2155,43	2002,94	1990,01	1676,14	2413,78
14	1321,40	1348,41	1348,41	1303,42	1330,57	1306,24	9119,03
15	2734,98	2824,25	2757,64	2681,62	2687,49	2406,84	2536,37
16	2969,35	3039,24	2975,17	2866,89	2861,08	2639,52	2741,50
17	1975,14	1987,44	2112,79	1871,73	1948,25	1928,77	<u>1747,24</u>
18	2961,12	3269,94	3015,85	2513,68	2478,97	2187,85	2373,63
19	9732,9	9747,43	9785,36	8892,58	8999,9	9486,57	<u>8661,81</u>
20	4430,28	4769,20	4580,13	4402,27	4381,59	4421,21	<u>4047,55</u>

Tabela 9 – Resumo dos resultados encontrados pelos algoritmos propostos

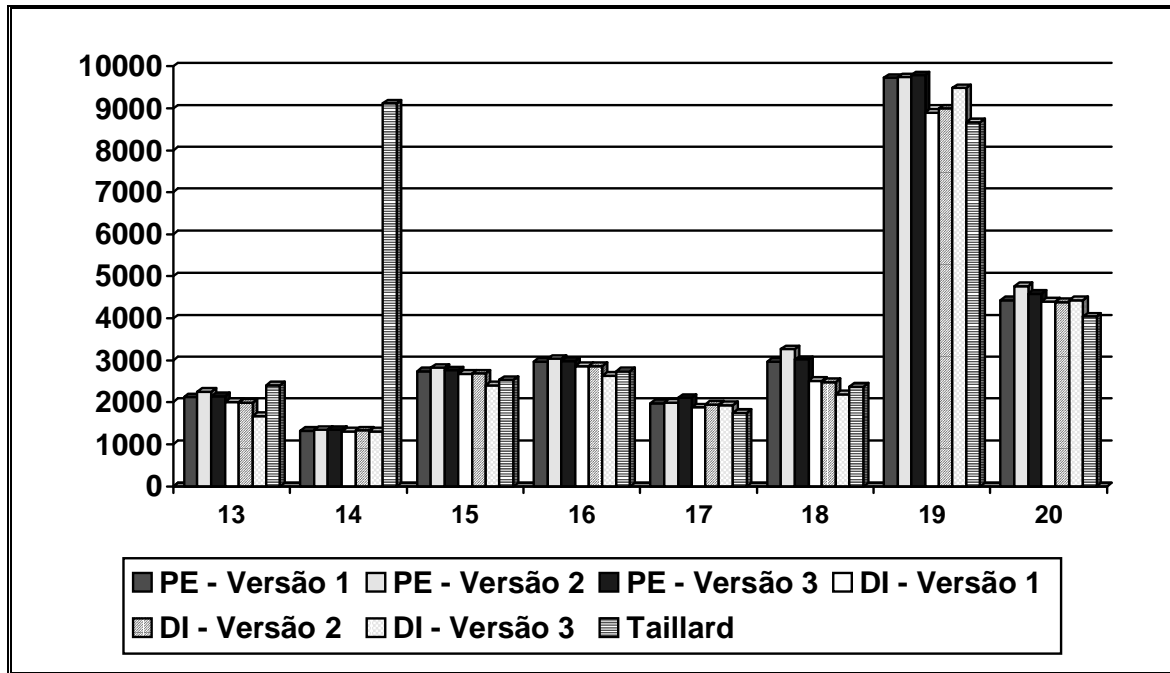


Gráfico 7 — Resumo dos resultados encontrados pelos algoritmos propostos

O algoritmo Diversificação-Intensificação apresentou em todos os problemas testados soluções de melhor qualidade do que o algoritmo Pétala-Elástico. As três versões do algoritmo Diversificação-Intensificação se alternaram entre as melhores e piores soluções encontradas.

Capítulo 6

Conclusões

Neste trabalho foram apresentados dois algoritmos heurísticos híbridos como mecanismo de busca para a solução do Problema de Roteamento de Veículos com Frota Heterogênea. Ao compararmos nossos resultados com o Método de Geração de Colunas, apresentado no Capítulo 5, observamos que eles não apresentam um comportamento nem estritamente pior, nem estritamente melhor. Em alguns teste nossas soluções são melhores e em outros são piores.

O algoritmo Pétala-Elástico apresentou soluções em média 12,28% piores, porém seu tempo de processamento pode ser considerado excelente. Com isso, podemos dizer que o algoritmo é viável já que com pequeno esforço computacional atingiu soluções de níveis razoáveis.

O algoritmo Diversificação-Intensificação apresentou um rendimento superior ao Pétala-Elástico. Em sua terceira versão o algoritmo conseguiu estabelecer melhores soluções em 62,5% dos casos, apesar de exigir mais tempo de processamento

As Redes Elásticas se mostraram eficientes no que tange a resolução das rotas determinadas pela divisão em *clusters* uma vez que o número de rotas não resolvidas pelo algoritmo foi insignificante. Porém, o tempo gasto na resolução das rotas no algoritmo Diversificação-Intensificação foi determinante no tempo final de execução. Ou seja, em média 70% do tempo gasto foi durante a resolução das rotas.

O problema de roteamento de veículos com frota heterogênea tem sido menos tratado pela literatura, porém ele é um problema mais realístico do que o com frota homogênea. Neste sentido, nosso método pode ser adaptado à problemas da vida real com uma certa facilidade.

O método apresentado neste trabalho possui características inerentemente paralelas, o que sugere a possibilidade da implementação de uma versão distribuída do algoritmo na tentativa de obter soluções melhores em bons tempos computacionais.

Referências Bibliográficas

- [1] Back, T. (1996), *Evolutionary Algorithms in Theory and Practice*, Oxford University Press.
- [2] Back, T.; D.Fogel and Z. Michalewicz (Editors) (1997), *Handbook of Evolutionary Computation*, Oxford University Press.
- [3] Badeau, P.; Gendreau, M.; Potvin, J. -Y; Taillard, É, D. (1995), *A parallel tabu search heuristic for the vehicle routing problem with time windows*, Publication CRT-95-84, Centre for research on transportation, University of Montreal.
- [4] Bodin, L. D.(1990), *Twenty years of routing and scheduling*, Oper.Res.,vol.38 (4), pp: 571-579.
- [5] Bodin, L.D. & B. L. Golden(1981), *Classification in Vehicle Routing and Scheduling, Networks*, vol.11, pp: 97-108.
- [6] D. J. Burr (1988), *An Improved Elastic Net Method for the Traveling Salesman Problem*, Proc. Int. Conf. on Neural Network, I-69-76.
- [7] De Jong, K .A.(Ed.)(1993), *Evolutionary Computation*, MIT Press.
- [8] Durbin, R. and Willshaw D. (1987), *An Analogue approach to the Travelling Salesman Problem using an Elastic Net Method*, Nature 326, pp: 689-691.
- [9] Durbin, R.; Szeliski, R. and Yulle (1989) , A., *An Analysis of the Elastic Net Approach to the Taveling Salesman Problem*, Neural Computation 1, pp: 348-358.
- [10] Foster, B. A. and Ryan, D. M. (1976), *An Integer Programming Approach To The Vehicle Scheduling Problem*, Opl. Res. Q. 27, pp: 307-384.
- [11] Glover, F; Laguna, M. ; Taillard, E. and Werra, D. de (editors) (1993), *Tabu Search* □ *Annals of Operations Research*, vol. 41, J. C.Baltzer AG.
- [12] Glover, F.(1989), *Tabu Search - Part I*, ORSA Journal on Computing, 1(3), 190-206.
- [13] Glover, F.(1990), *Tabu Search - Part II*, ORSA Journal on Computing, 2, pp:4-32.
- [14] Glover, F. (1986), *Future Paths for Integer Programming and Links to Artificial Intelligence*, Computers and Operations Research, 13, 533-549.

- [15] Golden, B.; Assad, A.; Levy, L. and Gheysens, F. G. (1984), *The fleet size and mix vehicle routing problem*, Computers and Operations Research 11, 49-66.
- [16] Golden, B. L. & A A .Assad (Ed)(1988), *Vehicle Routing: Methods and Studies*, Studies in Manag. Science and Systems 16, North-Holland Publ.
- [17] Kasahara,H. and S. Narita(1984), *Practical multiprocessor scheduling algorithms for efficient parallel processing*, IEEE J.Robotics and Automation RAI - 2- pp: 104-113.
- [18] Kirkpatrick, S.; Gellat, C. D. and Vecchi, M. P. (1983), *Optimization by Simulated Annealing*, Science, vol 220, n° 4598, pp. 71-680.
- [19] Lenat,D.(1982), *The nature of heuristics*, Artificial Intelligence 19, pp: 189-249.
- [20] Lin, S. and Kernighan, B. W. (1973), *Na Effective Heuristic Algorithm for the Traveling Salesman Problema*, Operations Research 21, 498-516.
- [21] Michalewicz, Z.(1996), *Genetic Algorithm + Data Structures = Evolution Programs*, Third Edition, Springer.
- [22] Moscato, P.(1993), *An introduction to population approaches for optimization and hierarchical objective functions: A discussion on the role of Tabu Search*, Annals of Operations Research, vol.41, pp:85-122.
- [23] Ochi, L.S.; Maculan, N. and Santos, E. M. (1995), *A Genetic Algorithm for the TPP*, Proceeding of the Metaheuristic Int. Conf. (MIC95), USA. Editors: Osaman, H & Kelly, J. P., pp. 52-57.
- [24] Ochi, L.S.; R. M. V. Figueiredo and N. Maculan(1996), *A hybrid metaheuristic using Elastic Net for the TSP*, (submetido revista Heuristics - Kluwer Academic Press).
- [25] Ochi, L.S.; R. M. V. Figueiredo and L. M. A. Drummond(1997), *Design and Implementation of a Parallel Genetic Algorithm for the Traveling Purchaser Problem*, APPLIED COMPUTING'97/ACM-SAC- (Eds: B. Bryant; J.Carroll; D.Oppenheim & K.George), Association for Comp. Machinery, Inc.
- [26] Ochi, L.S.; R. M .V. Figueiredo and N. Maculan(1997), *Using Multicommodity flow formulations for the vehicle routing problems*, Proc. of the INFORMS XXXIV - Barcelona (full paper submitted Issue of the European Journal of Opr. Res.)
- [27] Ochi, L.S.; Figueiredo, R. M. V. and Maculan, N. (1998), *A New Self-Organizing Strategy based on Elastic Nets for solving the Euclidean Traveling Salesman Problem*, in Proceed. Of 11th Int. Conf. on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems (IEA-AIE98), in Lecture Notes in Artificial Intelligence 1416, 479-487, Springer-Verlag.

- [28] Ochi, L. S.; Vianna, D.S. and Drummond, L. M. A. (1998), *Na Evolutionary Hybrid Metaheuristic for Solving the Vehicle Routing Problems with Heterogeneous Fleet*, In Lecture Notes in Computer Science 1391, Springer-Verlag, pp: 187-195.
- [29] Ochi, L. S.; Vianna, D. S. and Drummond, L. M. A.(1998), *A Parallel Genetic Algorithm for the Vehicle Routing Problemas*, In Lecture Notes in Computer Science, Springer-Verlag.
- [30] Pearl,J.(1984), *Heuristics: Intelligence search strategies for computer problem solving*, Addison Wesley, Reading, Mass.
- [31] Potvin, J. Y. and Bengio, S. (1994), *A Genetic Approach to the Vehicle Routing Problem with Time Windows*, Publication CRT-953, Centre for research on transportation, University of Montreal.
- [32] Psaraftis,H.N. (1995), *Dynamic vehicle routing: Status and prospects*, Annals of Oper.Res., vol. 61, pp: 143-164.
- [33] Rego, C. and Roucairol, C. (1994), *Na Efficient Implementation of Ejection Chain Procedures for the Vehicle Routing Problem*, Publication 44, PRISM Laboratory, University of Versailles, France.
- [34] Renaud, J.; Boctor F. F. and Laporte, G. (1996), *An Improved Petal Heuristic for the Vehicle Routing Problem*, Journal of the Operational Research Society 47, pp:329-336.
- [35] Ribeiro, C. C. and F. Soumis (1994), *A column generation approach to the multiple depot vehicle scheduling problem*, In Oper. Res. 42, pp: 41-52.
- [36] Rochat, Y. and E. Taillard (1995), *Probabilistic Diversification and Intensification in Local Search for Vehicle Routing*, Centre de Recherche sur les Transports, Universite de Montreal (To appear in Journal of Heuristics).
- [37] Ryan, D. M., Hjorring and F. Glover(1993), *Extensions of the Petal Method for Vehicle Routing*, J.Op.Res., vol.44(3), pp: 289-296.
- [38] Simmen, M. W. (1991), *Parameter Sensitivity Of The Elastic Net Approach To The Traveling Salesman Problem*, Neural Computation 3, pp: 363-374.
- [39] Taillard, É. D. (1993), *Parallel Iterative Search Methods for Vehicle Routing Problems*, Networks 23, 661-676.
- [40] Taillard, É. D. (1996), *A Heuristic Column Generation Method for the Heterogeneous Fleet VRP*, CRT-96-03.

Anexos

Anexo 1

Método Heurístico de Geração de Colunas

Introdução

A escolha do Método da Heurística de Geração de Colunas (*Heuristic Column Generation Method*) para servir de parâmetro para a análise da qualidade de nossos algoritmos propostos se deve ao fato deste apresentar os melhores resultados encontrados até o momento e de sua extrema robustez. O Método da Heurística de Geração de Colunas (*Heuristic Column Generation Method*) foi proposto por Éric Taillard [40] para solucionar o Problema de Roteamento de Veículos com Frota Heterogênea. Ele é baseado no “*Adaptative Memory Procedure*” (AMP) proposto por Rochat e Taillard [36] para solucionar o Problema de Roteamento de Veículos com Frota Homogênea. A Geração de Colunas, segundo Taillard, pode solucionar todos os outros tipos de problemas de roteamento de veículos, apresentando boas soluções. Os resultados numéricos mostraram que este método é robusto e eficiente, particularmente para instâncias de tamanhos médio e grande.

Descrição Geral do Método

Como os mecanismos para solucionar os Problemas de Roteamento de Veículos (VRP) com Frota Homogênea se tornaram cada vez mais eficientes, este método propõe o tratamento do Problema de Roteamento de Veículos com Frota Heterogênea (VFM - *Vehicle Fleet Mix*) como uma sucessão de problemas homogêneos. Para cada veículo é resolvido o problema de frota homogênea (sem limitação no número de caminhão disponíveis) utilizando o

AMP. As rotas da solução do problema homogêneo são então combinadas para gerar soluções para o VFM.

O AMP inicialmente gera um conjunto de boas soluções utilizando *Busca Tabu* [39]. Depois então, ele extrai as rotas com um só veículo do seu conjunto de soluções e combina algumas destas rotas para determinar uma solução parcial para uma outra aplicação da *Busca Tabu*. Este processo é repetido um número de vezes e as rotas são memorizadas como candidatas para uma solução final do VFM. Uma vez que os problemas homogêneos são resolvidos para cada tipo de veículo, teremos um conjunto T de rotas que terão sido memorizadas. As rotas consideradas inúteis são removidas de T : apenas uma cópia de cada rota é mantida em T ; as rotas “dominadas” são eliminadas (uma rota é considerada “dominada” se existir uma outra rota cobrindo os mesmos clientes mas com um custo menor).

Seja $m=|T|$ e c_{jk} ($j=1,\dots,m;k=1,\dots,K$) o custo da j -ésima rota de T realizada por um veículo do tipo k ($c_{jk}=\infty$ se a j -ésima rota de T não pode ser realizada pelo veículo do tipo k). Define-se a matriz $A=(a_{ij})$, onde:

$$a_{ij} = \begin{cases} 1, & \text{se o cliente } i \in \text{rota } j \text{ de } T \\ 0, & \text{caso contrario} \end{cases}$$

A j -ésima rota de T define uma coluna de A . Conseqüentemente, o AMP é um procedimento heurístico de geração de colunas no sentido de que somente um número muito limitado de rotas entre todas as que podem ser construídas para o VRP homogêneo é gerado. A melhor solução para o VFM que pode ser construída a partir das rotas de T , pode ser encontrada através da solução do seguinte problema linear:

$$\begin{aligned} \min & \sum_{j=1}^m c_j x_j \\ \text{s.a} & \sum_{j=1}^m a_{ij} x_j = 1 \quad i=1,\dots,n \\ & x_j \in \{0,1\} \quad j=1,\dots,m \end{aligned}$$

Temos que $c_j = \min_{k \in \Psi} (c_{jk})$, ($j=1,\dots,m$). Este é um problema NP-Árduo. Um *software* geral, como o CPLEX, é capaz de encontrar uma solução ótima para o problema com muitas centenas de linhas e colunas [Chu and Beasley, 1995].

O Algoritmo

Mais formalmente o algoritmo pode ser descrito da seguinte maneira:

1. Para cada tipo de veículo $k \in \psi$, repetir: (AMP)
 - 1.1. Com o algoritmo de *Busca Tabu* [39], gerar I soluções iniciais para o VRP homogêneo com veículos do tipo k
 - 1.2. Armazenar as rotas destas soluções em um conjunto M_k .
 - 1.3. Repetir R vezes
 - 1.3.1. $M_k' = M_k$
 - 1.3.2. Repetir, até que $M_k' = \emptyset$ (construção de uma nova solução inicial)
 - 1.3.2.1. Selecionar aleatoriamente um rota de M_k'
 - 1.3.2.2. Remover de M_k' todas as rotas que possuem os mesmos clientes em comum com a rota selecionada
 - 1.3.3. Utilizar as rotas selecionadas como ponto de partida e utilizar *Busca Tabu* para gerar uma nova solução do VRP.
 - 1.3.4. Adicionar cada rota desta solução em M_k .
2.
$$T = \bigcup_{k \in \psi} M_k$$
3. Para cada rota t_j de T , calcular o custo c_{jk} (custo de t_j quando realizada pelo caminhão do tipo k)
4. Remover as rotas "inúteis"
5. Utilizar o CPLEX para resolver o problema de programação linear inteira (**A**) associado com T

Algoritmo de Geração de Colunas