

Ministério da Ciência e Tecnologia – MCT

Instituto Nacional de Pesquisas Espaciais – INPE

O PROBLEMA DE ROTEAMENTO DE VEÍCULOS E ALGUMAS
METAHEURÍSTICAS

Reinaldo Gen Ichiro Arakaki

Monografia apresentada para o Exame de Qualificação do Curso de Computação Aplicada
– CAP

Setembro de 1998

SUMÁRIO

CAPÍTULO 1	1
1. INTRODUÇÃO	1
1.1 DESCRIÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS	1
CAPÍTULO 2	6
2. EXTENSÕES AO PROBLEMA DE ROTEAMENTO DE VEÍCULOS	6
2.1 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM DIVISÃO DE ENTREGA (PRVDE)	6
2.2 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELA DE TEMPO (PRVJT)	6
2.3 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM MULTI-DEPÓSITO (PRVMD)	7
CAPÍTULO 3	9
3. MÉTODOS EXATOS PARA O PRV	9
CAPÍTULO 4	10
ALGORITMOS GENÉTICOS E O PROBLEMA DE ROTEAMENTO DE VEÍCULOS	10
4.1 INTRODUÇÃO AOS ALGORITMOS GENÉTICOS (AG)	10
4.2 ALGORITMOS GENÉTICOS APLICADOS AO PRV	12
CAPÍTULO 5	18
GRASP E O PROBLEMA DE ROTEAMENTO DE VEÍCULOS	18
5.1. INTRODUÇÃO AO GRASP	18
5.2 APLICAÇÃO DE GRASP AO PRV	18
CAPÍTULO 6	20
BUSCA TABU E PROBLEMA DE ROTEAMENTO DE VEÍCULOS	20
6.1. INTRODUÇÃO A BUSCA TABU	20
6.2 BUSCA TABU APLICADO AO PROBLEMA DE ROTEAMENTO DE VEÍCULOS	21
CONCLUSÕES	24
REFERÊNCIAS BIBLIOGRÁFICAS	25

CAPÍTULO 1

1.Introdução

Há uma série de serviços como entrega bancária, entrega postal, entrega de mercadorias, rotas de ônibus escolar, coleta de lixo industrial, serviço de entrega noturnas, operações de frete etc que são basicamente problemas de distribuição. A solução destes problemas pode diminuir bastante o custo de distribuição, causando uma grande economia tanto para a indústria ou para o governo. No entanto, muitos destes problemas são difíceis de resolver. Estes dois atrativos fazem com que haja muita literatura sobre estes problemas que são conhecidos como problemas de roteamento e planejamento (scheduling).

A utilização de modelagem matemática aliada aos computadores tem-se mostrado satisfatória em termos de diminuição de custos de distribuição. Há uma série de pacotes comerciais que resolvem problemas de roteamento e planejamento, porém estas soluções tendem a serem soluções aproximadas (utilizam heurísticas), porque como foi dito anteriormente, são extremamente difíceis de resolver. Algoritmos exatos frequentemente exigem um tempo computacional não razoável e são limitados a problemas pequenos.

Neste trabalho, procuraremos dar uma visão geral do problema de roteamento de veículos (PRV) e as suas variações, bem como conhecer algumas metaheurísticas como algoritmos genéticos (AG), busca tabu (BT) e Procedimentos de Busca Adaptativa Aleatória Gulosa (GRASP- Greedy Randomized Adaptive Search Procedures) aplicados ao PRV.

No capítulo 1 introduzimos o PRV e a sua formulação matemática. No capítulo 2 é mostrado as diversas variações encontradas para o PRV. No capítulo 3 uma sucinta descrição dos métodos exatos para o PRV. No capítulo 4 abordamos os algoritmos genéticos – uma introdução e aplicação ao PRV. O mesmo é feito para o GRASP no capítulo 5 e no capítulo 6 para a busca tabu.

1.1 Descrição do problema de roteamento de veículos

Considerando n clientes cada um com uma demanda de mercadorias. As mercadorias são entregues a partir de um depósito por uma frota de veículos homogêneos. Cada veículo

realiza um percurso saindo do depósito e entregando as mercadorias para um subconjunto de clientes, satisfazendo as necessidades de demanda de cada um e retornando ao depósito. A rota de cada veículo deve obedecer a algumas restrições como: a quantidade de mercadoria entregue não deve exceder a capacidade do veículo e o tempo limite para realizar uma rota não deve ser ultrapassado. O problema de roteamento de veículos pretende traçar rotas para os veículos, determinando a quais clientes deve-se fornecer a mercadoria, de forma a não violar as restrições e otimizar alguma função objetivo. Normalmente são considerados três funções objetivos:

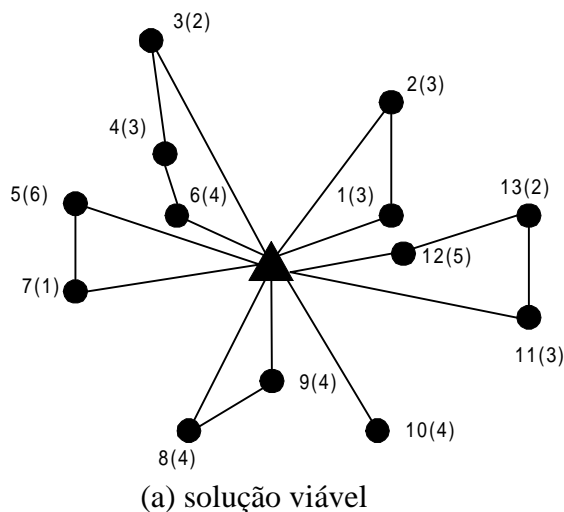
O₁. Minimizar a distância total percorrida (ou tempo gasto) por todos os veículos,

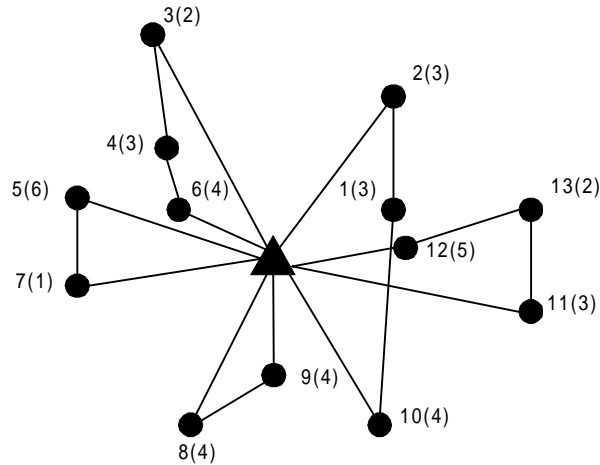
O₂. Minimizar o número de veículos e deste número mínimo, minimizar a distância total percorrida.

O₃. Minimizar a combinação de custo de veículos e distância percorrida.

Existem outras muitas possibilidades de funções objetivos como Christofides et al. (1979) mostram.

Um exemplo de solução para o PRV é mostrado na figura 1.1, os clientes estão ordenados radialmente e os valores entre parenteses são os pedidos de mercadorias (a capacidade do veículo é 10). A primeira solução utiliza 6 veículos e a melhor solução 5 veículos.





(a) melhor solução viável

figura 1.1- Exemplo de soluções PRV
Fonte: Hjorring (1995)

Uma formulação de programação matemática para o PRV foi dada por Bodin et al. (1983).

A formulação considera o depósito como nó 0 e os clientes são numerados de 1 a n de modo que o problema se torna;

$$\text{Minimize } \sum_{i=0}^n \sum_{j=0}^n \sum_{v=1}^K c_{ij}^v x_{ij}^v \quad (1.1)$$

Suj a

$$\sum_{i=0}^n \sum_{v=1}^K x_{ij}^v = 1 \quad j = 1, \dots, n \quad (1.2)$$

$$\sum_{i=0}^n \sum_{v=1}^K x_{ij}^v = 1 \quad i = 1, \dots, n \quad (1.3)$$

$$\sum_{i=0}^n x_{ip}^v - \sum_{j=0}^n x_{pj}^v = 0 \quad j = 1, \dots, n \quad (1.4)$$

$$\sum_{j=0}^n q_j \left(\sum_{i=0}^n x_{ij}^v \right) \leq Q_v \quad v = 1, \dots, K \quad (1.5)$$

$$\sum_{j=1}^n \delta_j^v \left(\sum_{i=0}^n x_{ij}^v \right) + \sum_{i=0}^n \sum_{j=0}^n t_{ij}^v x_{ij}^v \leq T_v \quad v = 1, \dots, K \quad (1.6)$$

$$\sum_{j=1}^n x_{0j}^v \leq 1 \quad v = 1, \dots, K \quad (1.7)$$

$$\sum_{j=1}^n x_{i0}^v \leq 1 \quad v = 1, \dots, K \quad (1.8)$$

$$X \in S \quad (1.9)$$

$$x_{ij}^v \in \{0,1\} \text{ para todo } i, j, k \quad (1.10)$$

Onde : n = número de clientes

K = número de veículos

c_{ij}^v = custo de viagem do nó i ao nó j para o veículo v

$$x_{ij}^v = \begin{cases} 1, & \text{se veiculo viaja do no } i \text{ ao no } j \\ 0, & \text{caso contrario} \end{cases}$$

q_i = tamanho do pedido do cliente i

Q_v = capacidade do veículo v

δ_i^v = tempo de descarregamento do cliente i para o veículo v

t_{ij}^v = tempo de viagem do nó i para o nó j para o veículo v

T_v = tempo de rota máximo permitido para o veículo v

X = matriz de componentes $x_{ij} \equiv \sum_{v=1}^K x_{ij}^v$

Restrições 1.2 e 1.3 asseguram que cada cliente é servido exatamente uma vez. A continuidade da rota é garantido pela restrição 1.4, onde se um veículo chega no ponto de entrega deve também partir daquele ponto. A restrição 1.5 é a restrição da capacidade do veículo. A restrição 1.6 limita o máximo comprimento da rota. A restrição 1.7 e 1.8 asseguram que cada veículo é usado não mais do que uma vez. A restrição 1.9 representa um restrição de quebra de “subtour”. Isto previne que uma rota de um veículo forme múltiplos “tour” desconectados .

Hjorring (1995) mostrou que o problema de roteamento de veículos é NP-difícil ou seja pertence a uma classe de problemas difíceis de resolver. Para um caso particular do PRV, ele se reduz ao clássico problema do caixeiro viajante, ou seja, encontrar as várias rotas dos veículos é resolver pequenos problemas do caixeiro viajante.

CAPÍTULO 2

2.Extensões ao Problema de Roteamento de Veículos

Muitos pesquisadores tem incluído outras extensões ao problema básico de roteamento de veículos, com o intuito de melhorar a aproximação a aplicações reais. Estes melhoramentos incluem acréscimos de novas restrições, relaxação de algumas outras restrições, tamanho da frota de veículos ou rotas de diferentes formas. Apresentaremos algumas delas como:

O problema de roteamento de veículos com divisão de entrega (PRVDE).

O problema de roteamento de veículos com janela de tempo (PRVJT).

O problema de roteamento de veículos com multi-depósito (PRVMD).

2.1 O problema de roteamento de veículos com divisão de entrega (PRVDE).

O PRV básico não permite que um cliente seja servido por mais de um veículo, podemos relaxar esta restrição permitindo que o cliente seja servido por mais de um veículo, se isto beneficia o custo total (no caso distância), isto pode ocorrer se a demanda do cliente estiver próximo da capacidade do veículo. Dror e Trudeau (1990) apresentam algumas heurísticas para resolver o problema e mostram que apesar da relaxação o problema é NP-completo. Os resultados apresentados por eles mostram a vantagem de permitir a divisão de entregas. Se a demanda dos clientes for próxima da capacidade do veículo a melhoria em distância chega a 10% em relação ao PRV básico.

Drol et al. (1994) descreveram uma restrição de relaxação utilizando branch and bound para PRVDE e concluíram que este é mais difícil de resolver exatamente que o clássico PRV.

2.2 O problema de roteamento de veículos com janela de tempo (PRVJT).

O problema de roteamento de veículos com janela de tempo (PRVJT) implica em satisfazer as necessidades dos clientes em termos de horário de entrega de mercadorias, a competição das distribuidoras (entregadoras) fez com que para atender melhor os clientes, estes determinem em que horários devem ser realizadas as entregas. A formulação do problema costuma estipular dois horários a_i e b_i e neste intervalo de tempo o cliente deve ser servido.

Algumas formulações permitem múltiplas janelas de tempo por cliente. Quando a janela de tempo deve ser estritamente respeitada se diz que é “forte” (hard) e quando a janela pode ser violada se diz que é “fraca” (soft) e sua violação está sujeita a algumas penalidades.

Quando há janela de tempo para entregas, deve-se levar em conta o custo incorrido no tempo de espera ao chegar muito cedo ao local de entrega e o tempo de carregar e descarregar o veículo. Solomon e Desrosiers (1988) apresentam uma revisão bibliográfica de vários problemas de roteamento e gerenciamento (“scheduling”) de veículos com janela de tempo, tais como o problema do caixeiro viajante, problema de múltiplos caixeiros viajantes e problema do caminho mínimo. Solomon e Desrosiers concluem que este problema é mais difícil do que o PRV básico.

Solomon (1987) faz um estudo de várias heurísticas para o PRV com janela de tempo forte como a heurística econômica (savings), heurística do vizinho mais próximo orientado ao tempo, heurística da inserção e heurística de varredura orientada a tempo. Chegando a conclusão após um conjunto de testes que a heurística de inserção aliada a uma abordagem de varredura é a melhor heurística para obter uma excelente solução inicial ao PRVJT.

Kosdosidis et al. (1992) compara para o PRV com janela de tempo fraca uma otimização baseada em heurística e algoritmos de heurísticas locais. Kosdosidis et al. (1992) faz a decomposição do PRVJT-fraco resultando em um problema de atribuição/agrupamento e vários problemas do caixeiro viajante.

2.3 O problema de roteamento de veículos com multi-depósito (PRVMD).

Suponhamos que em vez de apenas um depósito haja vários depósitos que podem servir os clientes. Se estes clientes estão agrupados aos depósitos o problema de distribuição a ser resolvido poderia ser modelado de forma a termos um conjunto independente de PRVs. Porém, se ambos clientes e depósitos estiverem misturados cairemos no problema de roteamento de veículos com multi-depósito.

Wren e Holliday (1972) descrevem um algoritmo que constrói uma solução inicial para o problema de roteamento de veículos com multi-depósito e um número de diferentes refinamentos heurísticos são aplicados a solução obtida pelo primeiro algoritmo. A

construção inicial é realizado com um algoritmo do tipo varredura. Os clientes são atribuídos ao seus mais próximos depósitos, e os ângulos formados entre os clientes e os depósitos são calculados. A partir disto, os clientes são ordenados no sentido do relógio, independente do depósito. As rotas são construídas sequencialmente em cada depósito. Os clientes são inseridos na rota que minimiza o custo.

CAPÍTULO 3

3. Métodos Exatos para o PRV

Os métodos exatos para o PRV produzem resultados eficientes em termos de tempo computacional quando aplicados a problemas pequenos (máximo 50 clientes). Serão abordados dois métodos: o método da busca em árvore direta e a programação inteira pela formulação por particionamento de conjuntos. Ambos apresentados em Christofides et al. (1979).

O método da busca em árvore direta consiste na construção incremental das rotas dos veículos por meio de uma árvore *branch and bound*. Cada nó da árvore corresponde a uma única rota viável para um veículo. Portanto a árvore terá K (número de veículos) níveis. Em cada nó uma lista de rotas passando por um cliente i é gerado para ramificá-lo. O desempenho do algoritmo é diretamente dependente do número de rotas geradas em cada nó. O cliente i deve ser escolhido de maneira que o número de rotas seja pequeno, isto é, o cliente tem um pedido grande ou que seja distante de outros clientes. Ele sonda as ramificações da rotas em busca de violações das restrições, utilizando um limite inferior que mostra que o ramo selecionado não conduz a melhor solução do que a solução.

A abordagem por particionamento de conjuntos para resolver o PRV parte do pressuposto que todas as rotas na qual um único veículo pode operar viavelmente pode ser gerado, então se associa cada veículo a um subconjunto de clientes que ele pode servir e utiliza uma matriz $G=[g_{ij}]$ onde cada linha i corresponde a um cliente x_i e com M blocos de colunas. O k -ésimo bloco corresponde a um veículo v_k e a j_k coluna deste bloco corresponde a uma única rota viável S_{jk} deste veículo. Seja $g_{ik} = 1$ ou 0 dependendo se o cliente x_i pertença ou não a S_{jk} , respectivamente. O PRV torna-se um problema de escolher no máximo, uma coluna de cada bloco de G de maneira que cada linha de G tenha valor 1 sobre uma das colunas escolhidas e custo total de colunas escolhidas seja mínimo. Este problema pode ser facilmente modificado para se tornar um problema de particionamento de conjuntos.

CAPÍTULO 4

Algoritmos Genéticos e o Problema de Roteamento de Veículos

4.1 Introdução aos Algoritmos Genéticos (AG)

Os algoritmos genéticos foram introduzidos por John Holland , na década de 70, com intuito de aplicar as idéias de Darwin ou seja utilizar os conceitos da evolução biológica como genes, cromossomos, cruzamento, mutação e seleção a teoria computacional.

Atualmente, os AG tem se constituído ferramentas poderosas para resolver problemas onde o espaço de busca é muito grande e os métodos convencionais se mostraram ineficientes (Srinivas e Patnaik, 1994).

Diferentemente dos métodos de busca e otimização tradicionais, AG trabalham não com uma única solução, mas com uma população de soluções ; em vez dos parâmetros dos problemas , da codificação desses parâmetros; utilizam ainda informações de custo ou recompensa e regras de transição probabilísticas.

Embora AG sejam, atualmente, uma classe de procedimentos pois existem variações na sua forma, faremos uma explicação sucinta do funcionamento básico destes algoritmos.

Inicialmente, temos a geração de uma população formada por indivíduos (criados aleatoriamente) que podem ser vistos como possíveis soluções do problema. Após o processo de seleção uma porcentagem dos mais adaptados permanecem e os outros são descartados, por isso cada indivíduo é avaliado por uma função de avaliação recebendo uma nota ou índice. Para obter uma boa diversidade de soluções é aplicado ao indivíduos selecionados operadores de cruzamento e mutação gerando a próxima descendência. Este processo chamado de reprodução é continuado até que algum critério seja alcançado (número de iterações ou solução satisfatória alcançada).

A população é composta por um conjunto de cadeias de bits ou caracteres (cromossomos). Cada cromossomo é avaliado por uma função de avaliação (fitness) e atribuído a ele (figura 4.1).

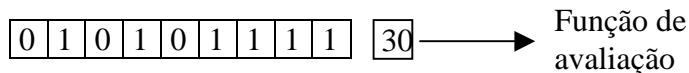


Figura 4.1 – Estrutura de um cromossomo

A seleção é feita para que do conjunto inicial de indivíduos, após muitas gerações, os indivíduos mais aptos permaneçam. Um método de seleção muito utilizado é o Método da Roleta, onde indivíduos de uma geração são escolhidos para fazer parte da próxima geração, através do sorteio de roleta. Neste método, os indivíduos mais aptos possuem uma porcentagem maior na roleta em relação aos indivíduos menos aptos, gira-se a roleta um determinado numero de vezes, dependendo do tamanho da população, e são escolhidos os indivíduos para a próxima geração.

O cruzamento (crossover) é um operador responsável pela recombinação de características dos pais durante a reprodução e é aplicado com probabilidade dada pela taxa de cruzamento.

Este operador pode ser de:

- um ponto: um ponto de cruzamento é escolhido e a partir deste as informações genéticas dos pais são trocadas.
- Multi-pontos: é a generalização da idéia de troca de material genético através de pontos, onde muitos pontos de cruzamento podem ser utilizados.

O operador mutação altera arbitrariamente um ou mais genes de um cromossomo. Desta maneira, assegura-se que a probabilidade de se chegar a qualquer ponto do espaço de busca nunca será zero, além de contornar o problema de mínimos locais, pois com este mecanismo, altera-se levemente a direção de busca.

4.2 Algoritmos Genéticos aplicados ao PRV

A dificuldade inicial para a implantação de AG ao PRV é a codificação de seus parâmetros, pois não é natural para este tipo de problema. Uma representação mais sofisticada deve ser utilizada para codificar as rotas. E esta codificação pode ser feita das mais variadas formas.

Hjorring (1995) fez um estudo de três formas de cruzamento para o PRV. A primeira utiliza cruzamentos padrões e uma codificação que aborda o PRV como um problema de particionamento (os clientes são particionados em diferentes rotas). A codificação feita pelo AG toma uma cadeia, $I = i_1 i_2 \dots i_n$ de n genes. Cada gene é um número de $1 \dots K$, onde o j -ésimo gene, i_j , é o índice do veículo que serve o j -ésimo cliente. De forma que cada cliente, j , é atribuído ao i_j -ésima rota. Este tipo de codificação faz com que soluções levemente diferentes possam ter codificações muito diferentes e ao se aplicar os operadores padrões, freqüentemente resultarão em soluções inviáveis. Para contornar o problema dos indivíduos inviáveis foi feita um reparo nos indivíduos pela criação de rotas “sobrecarregadas”, de forma que os clientes que estavam em rotas inviáveis fossem removidos até a rota se tornar viável.

A segunda forma utiliza um cruzamento feito inicialmente para o problema do caixeiro viajante (PCV) e trabalha com a ordem dos clientes; uma outra heurística, método da pétalas, gera as soluções para o PRV. A abordagem para o PCV utiliza simplesmente a ordem dos clientes a serem visitados, como meio de codificação, ou seja a representação do caminho. Como operador de cruzamento é utilizado o Recombinação de Bordas (Edge Recombination) desenvolvido por Whitley et al. (1989). Este operador constrói uma tabela de bordas formado pelas bordas de ambos os pais. Se uma borda é comum a ambos os pais, ele entra uma vez e é marcado com um sinal negativo. Esta tabela possui os vértices e as ligações existentes entre os vértices, uma vez a tabela montada, o operador dá o início a geração escolhendo um vértice aleatoriamente e verificando se há ligações com sinal negativo ou vértices com menos ligações. O vértice escolhido é eliminado da tabela e o processo se repete até formar um trajeto. Abaixo temos um exemplo:

Pai 1: a b c d e f g h i

Pai 2: a b d f e c g i h

Filho: a b d c e f g i h

Tabela de bordas:	vértice	ligações			
	a	i	-b	h	
	b	-a	c	d	
	c	b	d	e	g
	d	c	e	b	f
	e	d	-f	c	
	f	-e	g	d	
	g	f	h	c	i
	h	g	-i	a	
	i	-h	a	g	

A terceira forma trabalha diretamente sobre as soluções do PRV. Este operador de cruzamento, chamado de rota semente, combina rotas dos pais de forma que um deles é o pai semente da onde provêm a rota semente. O filho é gerado da seguinte forma: se faz uma cópia do outro pai (que não é o pai semente) e se retira a rota semente. Neste subconjunto de clientes ordenados se aplica o método das pétalas, gera-se, então uma solução para o PRV incompleta, após isto se introduz de volta a rota semente completando o filho.

Hjorring (1995) concluiu que os resultados apresentados tanto pela primeira forma (cruzamento padrão e PRV codificado como problema de particionamento) como pela segunda forma (recombinação de bordas) foram desanimadores. Em ambos, a porcentagem média acima do melhor conhecido da população inicial era 14%, utilizando o AG somente foi possível melhorar isto para 13% e 8,4% para a segunda forma. A terceira abordagem executou melhorou o desempenho do AG, apresentando um intervalo de apenas 1,36%.

Thangiah et al.(1991) proporam um sistema de AG para resolver o PRVJT. Seu sistema chamado de GIDEON consiste de dois módulos, um módulo de agrupamento global que atribui clientes aos veículos por um processo chamado setorização genética (GENSECT) e outro módulo que faz uma otimização local das rotas (SWITCH-OPT).

Inicialmente, os clientes são divididos em setores ou agrupamentos (clusters) utilizando um algoritmo de varredura que lembra o de Gillett e Miller (1974). A cada cliente é atribuído

um pseudo ângulo em coordenadas polares, de maneira que a diferença angular entre dois clientes adjacentes seja igual. Os clientes são divididos em K agrupamentos (K é o tamanho da frota de veículos) baseados num conjunto de ângulos “sementes”. Cada cliente é atribuído a um determinado agrupamento de acordo com seu ângulo (figura 4.2)

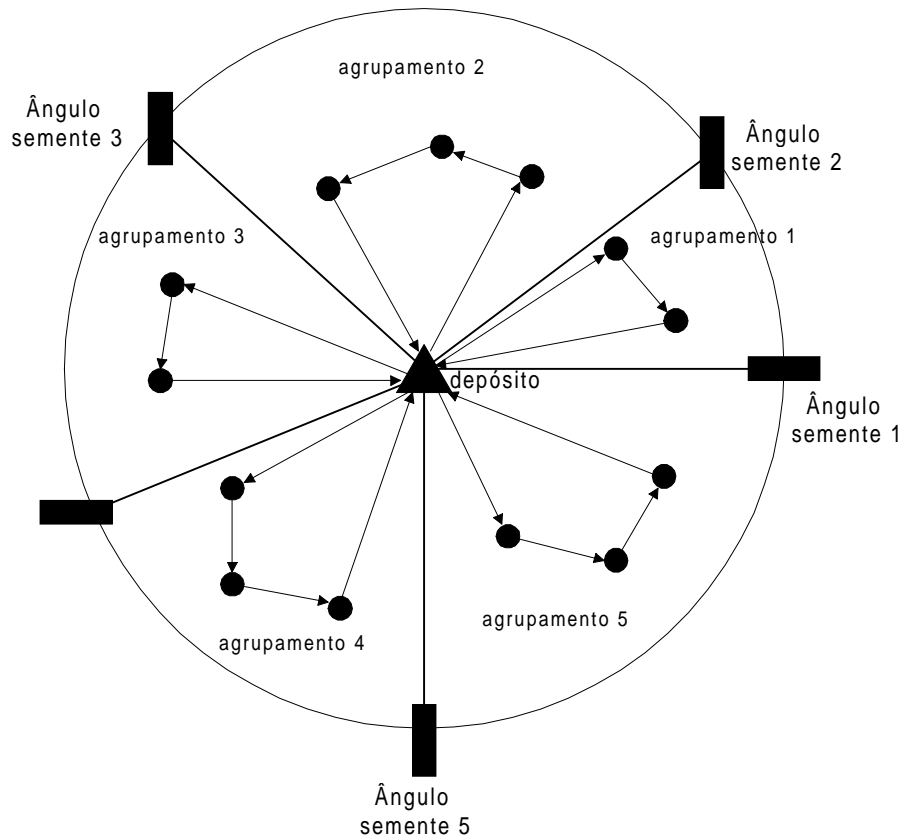


Figura 4.2 – Divisão de clientes utilizando ângulos sementes

Fonte: Thangiah et al. (1991)

O AG é utilizado para encontrar a melhor configuração de ângulos sementes. Durante o processo de setorização genética cada cromossomo representa um conjunto de ângulos sementes. Cada ângulo semente é representado por 3 bits e para um problema de K veículos, $K-1$ ângulos sementes são necessários (o primeiro ângulo semente é igual a 0) e portanto o comprimento de cada cromossomo será $(K-1)*3$. A população inicial é gerada aleatoriamente. Uma conversão de bits para ângulo semente é realizada. A função de

avaliação é o custo total das rotas dos agrupamentos formados a partir do conjunto de ângulos sementes derivados do cromossomo. Foram utilizados operadores de cruzamento tradicionais (2-pontos).

Portanto, os clientes são particionados de acordo com os ângulos sementes e as rotas são determinadas por uma heurística de inserção do mais barato (cheapest insertion).

O módulo de otimização de rota procura melhorar a solução pela movimentação e troca de clientes dentro e entre os agrupamentos. Quatro tipos operações são realizadas pelo módulo: mover 1 cliente, mover 2 clientes, trocar 1 cliente e trocar 2 clientes e em cada uma delas é avaliado o custo da rota. Estas operações de otimização local são realizadas até que não haja mais redução de custo. Esta nova solução pode ser introduzida novamente no primeiro módulo (GENSECT). Thangiah et al. verificou que duas iterações são suficientes para obter boas soluções.

GIDEON apresentou uma redução média de 3,9% no tamanho da frota e 4,4% na distância percorrida pelos veículos para 56 problemas.

Blanton e Wainwright (1993) descrevem um outro esquema de AG para o PRVJT, eles concluíram que os operadores de cruzamento tradicionais não eram eficientes para resolver problemas de otimização com múltiplas restrições e por isso apresentaram dois novos operadores de cruzamento: Combinação em Cruz#1(MX1) (Merge Cross) e Combinação em Cruz #2 (MX2). Eles se baseiam na noção de uma global precedência entre os genes independente de qualquer cromossomo. Isto é, dado um conjunto de N possíveis genes, deve existir uma relação de precedência de modo que há um desejo genérico de um gene i ocorrer antes do gene j dentro de um cromossomo. No PRVJT, cada gene é um cliente e tem associado uma janela de tempo de serviço, portanto há uma natural precedência entre os clientes por causa da sua janela de tempo correspondente. Esta precedência temporal é utilizada para escolher qual dos genes permanece no momento do cruzamento. Um exemplo é dado abaixo:

Vetor de precedência: 0 1 2 3 4 5 6 7 8 9

Cromossomo A: 2 5 6 1 0 7 3 8 4 9

Cromossomo B: 4 1 6 9 3 8 2 0 5 7

O primeiro gene de A (2) é anterior ao correspondente gene de B (4) como podemos notar no vetor de precedência, portanto ele permanece no filho AB e o primeiro gene de B (4) é trocado para manter a viabilidade resultando em:

Cromossomo A: 2 5 6 1 0 7 3 8 4 9

Cromossomo B: **2** 1 6 9 3 8 **4** 0 5 7

Filho AB 2 x x x x x x x x

Os segundos genes são comparados e neste caso o segundo gene de B permanece no filho AB e os genes 5 e 1 são permutados no cromossomo A. Continuando o cruzamento temos:

Filho AB 2 1 6 5 0 7 3 4 8 9

Blanton e Wainwright utilizam uma instância baseado em precedência temporal, porém, não há nenhuma razão que impeça que os operadores possam utilizar um outro tipo de precedência como distância, capacidade ou um outro critério de otimização. O exemplo utilizado se refere a um único veículo. Para múltiplos veículos, um método de codificação foi utilizado onde um cromossomo representa uma permutação de n clientes para ser divididos em k veículos. Uma heurística gulosa atribui os primeiros k clientes a k veículos. O restante n-k clientes são atribuídos individualmente. Uma inserção por tentativa de um cliente para alguma subrota é avaliada para cada veículo e a melhor subrota é selecionada (com menor custo).

Os operadores de cruzamento MX foram comparados com operadores de cruzamento padrões como recombinação de bordas (Edge Recombination), cruzamento cíclico (Cycle Crossover) e PMX para 4 problemas variando de 15 a 99 clientes. Em geral, os operadores MX se mostraram mais adaptados a encontrar soluções viáveis do que os operadores padrões.

Potvin e Bengio (1996) criaram um sistema de roteamento genético para o PRVJT, que é aplicado diretamente as soluções do PRVJT evitando o problema de codificação. Este sistema conhecido como GENEROUS, realiza o cruzamento pela ligação do primeiro cliente da rota do pai 1 ao último cliente da rota do pai 2, como na heurística de troca 2-opt. Um operador de reparos é aplicado para tornar a solução gerada viável. Primeiramente, vértices em duplicata são eliminados. Depois, cada vértice fora de rotas são acrescentados a nova solução utilizando o critério de inserção do mais barato. Se um vértice não pode ser inserido, a solução gerada é descartada. Embora o algoritmo consuma excessivamente os recursos computacionais, ele produz boas soluções aplicados aos problemas de Solomon (1987).

CAPÍTULO 5

GRASP e o Problema de Roteamento de Veículos

5.1. Introdução ao GRASP

Esta metaheurística GRASP (Greedy Random Adaptive Search Procedures) foi desenvolvida por Feo e Resende (1995) como uma tentativa de obter bons resultados para problemas difíceis de otimização combinatória. GRASP é um processo iterativo, onde cada iteração GRASP consiste de duas fases, uma fase de construção e uma fase de busca local. Na fase de construção, a solução viável é iterativamente construída elemento por elemento. Em cada iteração é feita uma escolha numa lista de candidatos ordenados por uma função gulosa que mede o quão bom é cada elemento. A heurística é adaptativa porque os benefícios associados com cada elemento são atualizados em cada iteração da fase de construção para refletir as mudanças ocorridas pela seleção de elementos anteriores. A parte aleatória corresponde a forma de escolha dos melhores candidatos da lista.

A fase de busca local de GRASP aproveita a solução inicial da fase de construção e explora a vizinhança ao redor desta solução. Se um melhoramento é encontrado, a solução corrente é atualizada e novamente a vizinhança ao redor da nova solução é pesquisada. O processo se repete até nenhum melhoramento for encontrado.

5.2 Aplicação de GRASP ao PRV

Hjorring (1995) utiliza GRASP para construir um ordem cíclica inicial de clientes e a partir desta ordem aplica o método das pétalas para gerar soluções para o PRV.

O método das pétalas faz uma varredura radial a partir de um primeiro cliente e o depósito e caminha na direção anti-horária (ou horária) formando subconjuntos contíguos com outros clientes verificando se não há violação de alguma restrição, caso haja termina-se esta rota e inicia-se outra. As rotas tidas como viáveis, respeitando todas as imposições do problema, tornam-se conjunto pétala, nome dado devido ao formato obtido da rota ligando o depósito e os clientes a serem percorridos.

A fase de construção da solução utiliza uma função de distância para avaliar os elementos, esta função é uma combinação da diferença angular utilizada para a ordem radial dos clientes e a distância em linha reta utilizada pela ordem do PCV. Esta fase inicia-se com um cliente arbitrário e forma um caminho pela junção do cliente com o seu vizinho mais próximo (avaliado pela função de distância). Prossegue-se até todos os clientes serem visitados, a partir daí os dois clientes que estão no extremo são ligados para formar uma ordem cíclica.

Na fase de busca local utiliza um algoritmo de reposicionamento de um cliente por vez. Porém, em vez de trocar alguns arcos, troca-se a ordem cíclica e utiliza o método das pétalas para atualizar a solução. Por causa da complexidade computacional do método das pétalas foram considerados avaliadores diferentes e a redução do número de candidatos avaliados em cada iteração da busca local. Os avaliadores são algoritmos que avaliam o efeito de um movimento de um candidato (no caso cliente saindo de uma rota para entrar em outra). Foram testados três avaliadores: avaliador-pétala, avaliador-rápido e avaliador de varredura. Concluiu-se que o avaliador-rápido possui um desempenho melhor do que os outros dois.

Hjorring (1995) obteve resultados mostrando que GRASP é uma heurística útil para resolver o PRV. Ele produz resultados melhores do que simplesmente utilizar a ordem cíclica radial dos clientes.

Não foi encontrada mais nenhuma tentativa de aplicar GRASP ao PRV.

CAPÍTULO 6

Busca Tabu e Problema de Roteamento de Veículos

6.1. Introdução a Busca Tabu

A busca tabu foi desenvolvida por Glover (1986) e esquematizado por Hansen (1990). Uma visão detalhada foi feita por Glover (1989,1990) através de um tutorial.

A busca tabu (BT) parte de uma solução inicial e através de uma sequencia de movimentos caminha para outra solução. O conjunto de movimentos é chamado de lista de candidatos e o método utilizado para comparar os movimentos é conhecido por avaliador.

Em muitos problemas de otimização combinatória há uma infinidade de ótimos locais que são inferiores ao ótimo global. Portanto, uma heurística que realiza movimentos até que um ótimo local seja alcançado, nem sempre alcançará um ótimo global. Consequentemente, seria interessante desenvolver uma heurística que uma vez encontrado um ótimo local continue a busca por ótimos locais melhores e talvez encontrando até o ótimo global. A BT permite isto impondo restrições na lista de candidatos.

Agora, pode ocorrer que na tentativa de buscar outras soluções o movimento realizado conduza diretamente de volta ao ótimo local já visitado. Para prevenir esta volta, um número de atributos relacionados ao movimento são registrados numa lista tabu. Os movimentos que revertem o movimentos já aceitados são considerados tabu, eles não serão aceitos a menos que satisfaçam algum critério de aspiração. A BT escolhe o melhor movimento admissível, cujo movimento não seja tabu ou satisfaça o critério de aspiração.

Possuindo os atributos do movimento consegue-se verificar se este é tabu ou não varrendo a lista tabu. Os atributos tabu são armazenados na lista tabu por um número pequeno de iterações e então são removidos. Atributos armazenados na lista intensificam a busca pelo travamento de um número de feições, restringindo o espaço de busca. Porém, a diversificação é conseguida quando se considera movimentos que não envolvem os atributos tabu de forma a direcionar a busca para novas e confiantes áreas do espaço de solução. Este processo de travamento de atributos e depois estrategicamente esquecer-los é chamado de componente de memória de termo curto da busca tabu.

Os critérios de aspiração são utilizados de modo que movimentos interessantes que são tabu não sejam rejeitados completamente.

6.2 Busca Tabu aplicado ao Problema de Roteamento de Veículos

Uma das primeiras tentativas de aplicar a BT ao PRV é devido a Willard (1989). O PRV é transformado, através da replicação dos depósitos, em um PCV e as vizinhanças são definidas como todas as soluções viáveis que possam ser atingidas através da solução corrente utilizando permutação 2-opt ou 3 opt, no entanto os resultados foram muito ruins, sendo superados por heurísticas tais como varredura (Gillet e Miller, 1974) e o algoritmo de duas fases de Christofides et al. (1979).

Pureza e França (1991) apresentam um implementação da BT aplicado diretamente ao PRV. A aplicação é muito básica e não há critério de aspiração, os movimentos considerados são baseados em movimentos de permutação de nós de Dror e Levy (1986) . Duas listas tabus foram utilizadas, uma para bordas deletadas e outra para bordas acrescentadas. Os resultados foram melhores do que a abordagem anterior, porém é inferior a implementações de BT posteriores.

Osman (1993) fez uma implementação cujo os movimentos são similares aos propostos por Pureza e França. A abordagem de Osman utiliza o mecanismo de permutação λ , com $\lambda=2$, isto inclui uma combinação de movimentos 2-opt, reatribuição de vértices a diferentes rotas e permutação de vértices entre duas rotas. O mecanismo 1-permutação foi utilizado para gerar a lista de candidatos e uma estrutura especial de dados foi criada para reduzir o tempo de avaliação da lista de candidatos. A implementação de Osman produziu resultados levemente melhores para problemas com restrições de tempo e capacidade.

Outra aplicação de BT é de Taillard (1993), em alguns aspectos é similar ao de Osman (vizinhança 1-permutação, restrições tabu). Utiliza um esquema de BT robusta com uma lista tabu de comprimento variando de $0,4n$ a $0,6n$. Para problemas Euclidianos, onde os clientes estão uniformemente distribuídos em volta do depósito central, Taillard sugere que os problemas sejam particionados em setores. Cada subproblema é então tratado como um independente PRV e resolvido utilizando BT para um pequeno número de iterações. Este

método de particionamento produziu resultados expressivos, encontrando para os testes de Taillard novas melhores soluções ou as melhores soluções conhecidas.

Semet e Taillard (1993) utilizaram a BT para resolver um PRV real, o problema envolve uma cadeia de supermercados na Suíça, possui janela de tempo para cada pedido de entrega, além de um frota heterogênea (21 caminhões e 7 *trailers*), outras restrições são impostas: muitos supermercados não podem ser acessados pelo trailer e outros somente podem aceitar alguns caminhões. Semet e Taillard descrevem uma técnica para atribuir otimamente veículos às rotas, por ser excessivamente dispendioso computacionalmente ele é somente executado no fim de uma iteração, quando se quer atualizar a solução e não para avaliar cada tentativa de movimento. Utiliza técnicas de redução de tempo de CPU, como por exemplo: reduzir o tamanho da vizinhança. Outra forma é agregar os pedidos, múltiplos pedidos de uma dada loja podem ser agregados para um único pedido, quando o volume total é menor do que a capacidade do menor caminhão que pode alcançar a loja. Esta agregação pode dificultar a descoberta de uma solução ótima para o problema, mas reduz, além do tempo de CPU, o espaço de soluções. Duas listas tabus foram estudadas. A primeira proíbe um pedido de retornar a alguma rota particular e a segunda proíbe lojas de receber entregas por alguma rota particular. Os resultados indicaram que procedimento da BT obteve melhores soluções do que aquelas geradas pelo método anteriormente utilizado pela companhia. Houve redução de custos de 10 a 15 por cento.

TABUROUTE é uma algoritmo desenvolvido por Gendreau et al. (1994), sua implementação possui características próprias. O movimento básico é feito através de permutação de vértices, utilizando GENI (procedimento de inserção generalizada feito por Gendreau et al. (1992)). Este procedimento consiste em inserir um vértice entre dois dos seus p vizinhos mais próximos na rota enquanto realiza um reotimização local da rota.

Outra característica é a busca estar oscilando entre soluções viáveis e inviáveis. Isto é útil para problemas altamente restritos onde é difícil mover diretamente de uma solução viável para outra. TABUROUTE não utiliza lista tabu mas etiquetas (tags) tabu aleatórias. Sempre que um vértice for removido de uma rota r para outra rota s numa iteração t , o

retorno deste vértice a rota r é proibida até a iteração $t + \theta$, onde θ é um inteiro aleatório de intervalo 5 a 10.

TABUROUTE possibilita a utilização de uma estratégia de diversificação através da penalização de vértices que se movem frequentemente com a finalidade de aumentar a probabilidade dos vértices de movimento lento. Processo geral do algoritmo começa com uma inicialização (geração de soluções iniciais) passando para um melhoramento da solução (executa a BT) e uma fase de intensificação da solução.

Os resultados de TABUROUTE são expressivos. O método produz resultados superiores a outras heurísticas conhecidas (Willard, Pureza e França, Osman) para 10 dos 14 problemas testados.

CAPÍTULO 7

CONCLUSÕES

Através deste estudo, pudemos perceber o quão interessante é o PRV e porque muitos pesquisadores tem gasto tempo em tentar resolvê-lo. Tanto pelo seu aspecto extremamente prático (muitos problemas reais de distribuição se reportam ao PRV) como pelo seu aspecto teórico (um número grande de heurísticas aplicadas e construídas) mostram o atrativo que é este problema.

Embora fácil de se descrever, o PRV é difícil de se resolver. Como foi visto anteriormente, este problema é NP- difícil. Embora muitas vezes relaxadas as extensões do PRV como o PRVDE, ou PRVJT, ou ainda o PRVMD tendem a ser NP-difícil. O que mostra que quanto mais perto de aplicações reais o problema torna-se mais complicado ainda.

Surge a idéia de aplicar heurísticas mais modernas para o PRV. Interessante notar que tanto GRASP, BT e AG trabalham em duas etapas, inicialmente utilizam uma heurística mais tradicional para encontrar uma solução inicial viável e a partir disto utilizando cada uma delas, as características próprias procuram melhorar a solução.

GRASP, BT e AG utilizam o conceito de vizinhança ou lista de movimentos candidatos, para buscar o espaço de soluções. Portanto, a estrutura de vizinhança para as melhores implementações consideram para cada iteração vizinhanças razoavelmente grandes que contenham soluções que sejam diferentes da solução corrente.

Enquanto as abordagens das metaheurísticas ao PRV, a BT se mostrou mais eficiente para as inúmeras versões do PRV. O uso de AG esbarra no problema da codificação do problema para forma cromossômica, quando aplicado diretamente ao PRVJT de Potvin e Bengio (1996) mostrou ser bastante promissor, devido as seus bons resultados. GRASP é uma heurística que necessita ser mais explorado ao PRV, estudos de Hjorring mostraram, numa comparação com BT e AG, resultado bem piores. Gendreau et al. (1997) analisam a aplicação de outras duas heurísticas ao PRV: *simulated annealing* e redes neurais. Seria interessante para um trabalho futuro a inclusão destas.

REFERÊNCIAS BIBLIOGRÁFICAS

Bodin, L; Golden, B.; Assad, A; Ball, M;. Routing and scheduling of vehicles and crews: The state of the art. *Computers and Operations Research*, 10:10, p. 200-211. 1983.

Blanton, J.L.; Wainwright, R. L. Multiple vehicle routing with time and capacity constraints using genetic algorithms. In *Proceeding of the 5th International Conference on Genetic Algorithms*. Forrest, S. editor. p. 452-459. 1993.

Christofides, N; Mingozzi, A. ; Toth, P. The vehicle routing problem. In Christofides, Mingozzi, Toth, Sandi, editors. *Combinatorial Optimization*, p. 315-338. John Wiley & Sons, 1979.

Dror, M; Trudeau, P. Split delivery routing. *Naval Research Logistics*, 37: p.383-402. 1990.

Dror, M; Laporte, G.;Trudeau, P. Vehicle routing with split deliveries. *Discrete Applied Mathematics*, 50. p. 239-254. 1994.

Dror, M. ; Levy, L. A vehicle routing improvement algorithm comparison of a “greedy” and a matching implementation for inventory routing. *Computers and Operations Research*, 13. p. 33-45.1986.

Feo, T. A. ; Resende, M.G.C. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, 6: p. 109-133. 1995.

Gendreau, M; Hertz, A; Laporte, G. A tabu search heuristic for the vehicle routing problem. *Management Science*, 40, p. 1276-1290. 1994.

Gendreau, M; Hertz, A; Laporte, G. New insertion and post optimization procedures for the traveling salesman problem. *Operations Research*, 40, p. 1086-1094. 1992.

Gendreau, M; Laporte, G.; Potvin, J-Y. Vehicle routing: modern heuristics. In *Local Search in Combinatorial Optimization*. Aarts, E.; Lenstra, J.K. editors. John Wiley & Sons Ltda. p. 311-336. 1997.

Gillet, B.E.; Miller, L.R. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):p 340-349. 1974.

Glover, F. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13. p. 533- 549. 1986.

Glove, F. Tabu Search – part I. *ORSA Journal on Computing*, 1(3): p. 190-206. 1989.

Glove, F. Tabu Search –part II. *ORSA Journal on Computing*, 2(1): p. 4-32. 1990.

Hjorring, C. A. The vehicle routing problem and local search metaheuristics. Phd Thesis. The University of Aucland. 1995.

Koskosidis, Y.A., Powell, W.B.; Solomon, M. M. An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints. *Transportation Science*, 26(2). p.69-85. 1992.

Osman, I. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41 p. 421-451. 1993.

Potvin, J.Y.; Bengio, S. The vehicle routing problem with time windows; part II: genetic search. *INFORMS Journal on Computing* 8 (2): p. 165-172.

Pureza, V.M.; França, P.M. Vehicle routing problems via tabu seach metaheuristic. Technical Report CRT-747, Centre de Recherche sur les transport, Montreal, 1991.

Semet, F; Taillard, E. Solving real-life vehicle routing problems efficiently using tabu search. *Annals of Operations Research*, 41, p. 469-488.

Solomon, M.M.; Desrosiers, J. Time window constrained routing and scheduling. *Transportation Science*, 22. p. 1-13, 1988.

Solomon, M. M.. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2): p. 254-265. 1987.

Srivinas, M.; Patnaik, L.M. Genetic algorithmics: A survey. IEEE Computer Society, p. 17-26. 1994.

Taillard, E. Parallel iterative search methods for vehicle routing problems. Networks, 23(8), p. 661-674. 1993.

Thangiah, S. R.; Nygard, K.E.; Juell, P. L.. GIDEON : A Genetic algorithm System for vehicle routing with time windows. In: Proceedings of the 7th IEEE Conference on Artificial Intelligence Applications, p. 322-328, Miami Beach, FL. 1991

Whitley, D.; Starkweather, Fuquay, D. Scheduling problems and traveling salesman: The genetic edge recombination operator. In Proceedings of the 3th International Conference on Genetic Algorithms. Schaffer, S. editor. 1989.

Willard, J.A. G. Vehicle routing using r-optimal tabu search. Master's thesis. The Management School. Imperial College. London. 1989.

Wren, A. ; Holliday, A. Computer scheduling of vehicles from one or more depots to a number of delivery points. Operational Research Quaterly, 23:p. 333-344, 1972.