



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

**UMA NOVA ABORDAGEM PARA O PROBLEMA DE ROTEIRIZAÇÃO E
PROGRAMAÇÃO DE VEÍCULOS**

Geraldo Regis Mauri

Monografia apresentada para o Exame de Qualificação do Curso de Doutorado em
Computação Aplicada, orientada pelo Prof. Dr. Luiz Antonio Nogueira Lorena.

INPE
São José dos Campos
Maio - 2006



RESUMO

Este trabalho apresenta um modelo matemático geral e multi-objetivo para o problema de roteirização e programação de veículos e uma aplicação do *Simulated Annealing* para resolvê-lo. O problema de roteirização e programação de veículos consiste em desenvolver rotas econômicas que atendam às requisições de transporte dos clientes sem violar um conjunto de restrições operacionais e qualitativas. Uma requisição consiste do transporte de um determinado número de clientes entre uma origem e um destino, ambos específicos. O modelo trata a forma estática do problema e abrange vários casos distintos dos modelos mais comuns, tais como frota homogênea e heterogênea, garagens múltiplas ou única, e uma função de minimização multi-objetivo que trata os custos de transporte e a inconveniência dos clientes através de penalizações. A aplicação do *Simulated Annealing* é simples, porém para a geração de novas soluções vizinhas, são utilizados três movimentos de troca selecionados de forma aleatória e uniformemente distribuída, e as rotas são roteirizadas e programadas separadamente por outros métodos heurísticos. Os resultados computacionais são obtidos através de instâncias públicas disponíveis e comparados com outros métodos que apresentam o atual estado-da-arte em que o problema se encontra.¹

¹ Este trabalho conta com apoio financeiro da FAPESP (processo 04/11053-9).

SUMÁRIO

	<u>Pág.</u>
LISTA DE FIGURAS	3
LISTA DE TABELAS	4
CAPÍTULO 1 INTRODUÇÃO	5
1.1 Objetivos do trabalho	6
CAPÍTULO 2 ROTEIRIZAÇÃO E PROGRAMAÇÃO DE VEÍCULOS	7
CAPÍTULO 3 REVISÃO BIBLIOGRÁFICA	10
3.1 Considerações iniciais	10
3.2 Abordagens clássicas	10
3.3 Abordagens recentes	12
3.4 Considerações finais	14
CAPÍTULO 4 METODOLOGIA PROPOSTA	15
4.1 Considerações iniciais	15
4.2 Formulação matemática	17
4.3 Método de solução	19
4.3.1 Geração da solução inicial	20
4.3.2 Estrutura de vizinhança	23
4.4 Considerações finais	26
CAPÍTULO 5 EXPERIÊNCIA COMPUTACIONAL	28
5.1 Considerações iniciais	28
5.2 Resultados	29
5.3 Comportamento do <i>Simulated Annealing</i>	32
5.4 Considerações finais	33
CAPÍTULO 6 CONCLUSÃO	35
6.1 Trabalhos futuros	36
REFERÊNCIAS BIBLIOGRÁFICAS	37

LISTA DE FIGURAS

Figura 4.1: Heurística de distribuição.....	21
Figura 4.2: Heurística de programação.....	22
Figura 4.3: Movimento <i>Re-ordenar rota</i>	24
Figura 4.4: Movimento <i>Re-alocar pontos</i>	24
Figura 4.5: Movimento <i>Trocar pontos</i>	25
Figura 4.6: Algoritmo <i>Simulated Annealing</i> implementado.....	26
Figura 5.1: Início da execução do SA para a instância R10b.....	32
Figura 5.2: Final da execução do SA para a instância R10b.....	33

LISTA DE TABELAS

Tabela 5.1: Resumo dos experimentos realizados.....	29
Tabela 5.2: Melhores soluções obtidas pelo <i>Simulated Annealing</i>	30
Tabela 5.3: Melhores soluções obtidas por Cordeau e Laporte (2003b).....	31
Tabela 5.4: Melhores soluções obtidas por Bergvinsdottir et al. (2004).....	31

CAPÍTULO 1

INTRODUÇÃO

Nos países ocidentais, principalmente na Europa e América do Norte, várias autoridades locais vêm se dedicando à implantação de serviços de coleta e entrega ou à revisão dos sistemas já existentes em resposta à crescente demanda. Este fenômeno pode ser atribuído em parte ao envelhecimento da população, mas também a uma tendência para o desenvolvimento de serviços de cuidado com a saúde (ambulatórios e assistência às pessoas necessitadas) (Cordeau e Laporte, 2003a).

De uma forma geral, o problema de roteirização e programação de veículos compartilha várias características de problemas com coleta e entrega, como por exemplo os serviços de mensagens (correios). Porém, em se tratando do transporte de pessoas, o nível do critério de serviço (a “qualidade” do serviço) se torna mais importante e complexo. Assim, pontualidade, redução do tempo de espera e a duração das rotas são mais críticas nesse caso do que em outros problemas correlatos (Cordeau e Laporte, 2003a).

Mesmo dentro desse escopo de transporte de pessoas, esse problema ainda é bem abrangente, podendo ser aplicado em vários contextos, como por exemplo, o transporte porta-a-porta de pessoas anciãs, inválidas ou com dificuldades de mobilidade, o transporte de enfermos para ambulatórios, o transporte de crianças para escola, etc.

No Brasil, apesar de esse tipo de serviço ainda ser pouco comum, nota-se um certo crescimento em sua utilização nos últimos anos. Este fato é evidenciado pela operação em algumas regiões, como por exemplo, nos municípios de Vitória e Vila Velha – ES, São José dos Campos e São Paulo – SP. No município de São Paulo, tal serviço (apenas para deficientes) opera desde 1996, por iniciativa de uma empresa chamada São Paulo Transportes S/A, através de um sistema denominado ATENDE. Embora o sistema utilizado não consiga acompanhar o nível de demanda atual, não se pode negar que tal serviço constitui um significativo avanço na melhoria das condições de mobilidade e integração dos deficientes. Além disso, trata-se de um serviço de caráter social, pois se destina prioritariamente a famílias carentes, que não têm condições de prover o

deslocamento regular dos deficientes, na sua maioria crianças, até os centros especializados de atendimento (Znamensky e Cunha, 1998).

Assim, considerando-se o nível de demanda atual, não só no Brasil, mas como em todo o mundo, bem como a sua esperada tendência de aumento, faz-se imperativo o uso de métodos automatizados de roteirização e programação dos veículos para otimizar esse tipo de serviço.

Esse problema é conhecido por ser *NP-Hard*, e várias metodologias vêm sendo adotadas para resolvê-lo. Como pode ser visto em Cordeau (2004), uma formulação matemática exata para solucionar tal problema é muito complexa, e incapaz de ser resolvida em um tempo aceitável para problemas de “tamanho real”. Porém, o uso de heurísticas e metaheurísticas para solucionar esse problema vem aumentando significativamente, pois tais métodos apesar de não garantirem a obtenção de soluções ótimas, permitem a inserção das inúmeras restrições de uma forma mais amena.

1.1 Objetivos do trabalho

Este trabalho apresenta uma alternativa simples e eficaz para resolver o problema em questão. É proposto um modelo matemático geral e multi-objetivo para representar o problema. Para tratá-lo, é utilizada a metaheurística *Simulated Annealing* juntamente com outras heurísticas. Assim, busca-se encontrar boas soluções para o problema, ou seja, gerar roteiros que sejam econômicos do ponto de vista operacional e que satisfaçam à demanda dos clientes mantendo um “bom” nível de qualidade no serviço prestado.

O restante do trabalho está organizado da seguinte forma. O CAPÍTULO 2 faz uma descrição minuciosa sobre o problema. O CAPÍTULO 3 faz uma breve revisão bibliográfica sobre outras metodologias utilizadas para solucionar o problema. No CAPÍTULO 4 é descrito, em detalhes, o tratamento do problema e as metodologias utilizadas para resolvê-lo. O CAPÍTULO 5 apresenta as experiências computacionais realizadas. Finalmente, o CAPÍTULO 6 apresenta algumas conclusões a respeito do trabalho como um todo, e sugere possíveis extensões.

CAPÍTULO 2

ROTEIRIZAÇÃO E PROGRAMAÇÃO DE VEÍCULOS

Dentre a classificação dos problemas de roteirização e programação de veículos, aqueles que envolvem o transporte de passageiros de seus locais de origem a seus destinos são conhecidos genericamente na literatura como problemas do tipo “*dial-a-ride*” (Znamensky e Cunha, 1998).

Como provado por Baugh jr. et al. (1998), esses problemas são classificados como *NP-Hard*, ou seja, não se conhece um algoritmo com tempo polinomial em função do tamanho da entrada (número de requisições de transporte e veículos disponíveis) capaz de, dado uma solução, verificar se esta é a de custo mínimo (solução ótima).

O *Problema de Roteirização e Programação de Veículos* ou *Dial-a-Ride Problem* – *DARP* (ver Cordeau (2004), Cordeau e Laporte (2003a) e Cordeau e Laporte (2003b)) consiste no desenvolvimento de rotas e escalas de veículos para transportar diversos usuários, os quais especificam requisições de embarque e desembarque (coleta e entrega) entre locais de origem e destino específicos. O objetivo desse processo é planejar um conjunto de rotas para alguns veículos, com “custo mínimo”, capaz de acomodar o maior número possível de usuários, e sempre obedecendo a um conjunto de restrições.

Em se tratando especificamente do caso de transporte de deficientes, um ponto de embarque (coleta) corresponde ao endereço onde um determinado deficiente deve ser buscado, e o ponto de desembarque (entrega) associado corresponde ao endereço onde o mesmo deficiente deve ser entregue. Cada ponto de embarque e seu respectivo ponto de desembarque, juntamente com suas respectivas “janelas de tempo”, formam uma *Requisição de Transporte* de um cliente.

Uma tendência comum nos modelos do DARP é deixar que os usuários determinem uma “janela de tempo” (isto é, janelas ou intervalos de horários para seu atendimento) para sua partida e sua chegada, pois segundo Jaw et al. (1986), os usuários devem estar

aptos a especificar um intervalo de horários para seu embarque e desembarque, ambos em locais específicos, possibilitando assim um melhor planejamento da programação e roteirização dos veículos como um todo.

Cada veículo possui uma capacidade, medida normalmente em número de assentos convencionais e número de cadeiras de rodas, por exemplo. Analogamente, a cada solicitação de atendimento está associada uma ocupação em termos de quantidade de assentos convencionais (eventuais acompanhantes) e em quantidade de cadeiras de rodas (Znamensky e Cunha, 1998).

Segundo Cordeau e Laporte (2003a), e outros diversos autores, o DARP pode ser abordado de forma estática ou dinâmica. No primeiro caso, todas as requisições de transporte são conhecidas a priori, enquanto no segundo caso são consideradas requisições feitas ao longo do dia (normalmente por telefone), e as rotas dos veículos são ajustadas em tempo real de acordo com a demanda. Porém, na prática, raramente existem DARPs dinâmicos “puros”, pois normalmente um subconjunto de requisições é conhecido com antecedência.

A maioria dos estudos acerca do DARP assume a disponibilidade de um conjunto de veículos homogêneos instalados em uma única garagem, entretanto, é importante perceber que na prática existem situações diferentes, como por exemplo: pode haver várias garagens, especialmente em grandes áreas geográficas, e o conjunto de veículos às vezes é heterogêneo, sendo alguns veículos projetados para só transportar cadeiras de rodas, outros só para transportar passageiros para ambulatórios e ainda alguns capazes de acomodar ambos os tipos de passageiro (Cordeau e Laporte, 2003a).

O DARP pode ser considerado com vários objetivos distintos, como por exemplo: minimizar os custos operacionais sujeito à satisfação de toda a demanda; maximizar a satisfação da demanda, sujeito à disponibilidade dos veículos, ou outras combinações destes, que geralmente buscam um equilíbrio entre custo operacional e qualidade do serviço prestado.

Normalmente, o critério de avaliação utilizado na resolução do DARP inclui a distância total percorrida pelos veículos, a duração das rotas, o tempo médio de espera dos

veículos nos locais de embarque e desembarque e o tempo médio de viagem dos clientes (i.e., tempo de permanência dos clientes dentro dos veículos), sendo dado a cada um desses, um maior ou menor grau de importância em relação aos outros.

Em termos gerais, o DARP generaliza vários problemas de roteamento de veículos como o *Pick-up and Delivery Vehicle Routing Problem (PDVRP)* e o *Vehicle Routing Problem with Time Windows (VRPTW)*. O que faz o DARP diferente da maioria de tais problemas é a perspectiva humana, ou seja, no transporte de passageiros, a redução da “inconveniência” dos usuários (qualidade do serviço) deve ser equilibrada com a minimização dos custos operacionais (Calvo et al., 2004, Cordeau e Laporte, 2003a, Xu et al., 2003).

CAPÍTULO 3

REVISÃO BIBLIOGRÁFICA

3.1 Considerações iniciais

O Problema de Roteirização e Programação de Veículos (DARP) é um problema muito comum, porém relativamente pouco difundido dentro da Pesquisa Operacional e da Otimização Combinatória devido a sua alta complexidade. Entretanto, alguns pesquisadores vêm o estudando, e conseqüentemente apresentando métodos variados de solução e trabalhos realizados segundo seus preceitos e aplicações.

3.2 Abordagens clássicas

Psaraftis (1980) desenvolveu um algoritmo exato de *Programação Dinâmica* para a resolução de um problema de coleta e entrega para um único veículo, sem restrições de janelas de tempo, considerando os casos estático e dinâmico do problema. A função objetivo adotada foi a minimização de uma ponderação entre o tempo total de operação do veículo e a “insatisfação” total dos usuários, a qual é medida por uma função linear dos tempos de espera e viagem de cada usuário. A extensão do problema para a inclusão de restrições de janelas de tempo também foi analisada por Psaraftis (1983), e uma melhoria na solução obtida nessa extensão, também por meio de programação dinâmica, foi publicada por Desrosiers et al. (1986).

O trabalho proposto por Jaw et al. (1986) trata o problema de uma forma mais próxima a sua essência, ou seja, com múltiplos veículos e com restrições de janelas de tempo. Eles apresentam uma heurística denominada “*Heurística de Inserção Paralela*”, cuja idéia central consiste em ordenar as solicitações de atendimento por um critério de horário, e inseri-las, segundo a ordenação adotada, nas rotas que resultem em menor acréscimo de custo. Essa heurística foi testada com sucesso para uma base de dados de aproximadamente 2600 clientes e 20 veículos. É interessante destacar que essas instâncias representavam o problema para vários dias de operação, ou seja, nem todos os clientes solicitavam atendimento no mesmo dia.

Dumas et al. (1991) desenvolveram um algoritmo exato para o problema com frota homogênea e restrições de janelas de tempo, baseado no método de *Geração de Colunas*. Segundo os autores, o algoritmo foi desenvolvido tendo em vista instâncias do problema nas quais as restrições de capacidade dos veículos fossem fortemente restritivas e atuantes. O algoritmo foi aplicado com sucesso a problemas de até 55 solicitações.

Uma modificação da *Heurística de Inserção Paralela* de Jaw et al. (1986) foi desenvolvida por Madsen et al. (1995). A diferença central é a adoção de um critério de “dificuldade” para ordenação das solicitações. Este critério procura priorizar a inserção de solicitações com restrições mais rígidas, tornando a inserção mais eficiente, pois nos estágios iniciais da heurística os veículos nas rotas estão mais vazios e há um número maior de alternativas de inserções viáveis.

Ioachim et al. (1995) desenvolveram um método de solução que consiste na criação de “mini-clusters” (segmentos viáveis do itinerário em que o veículo não fica vazio), e na resolução do problema resultante dos m -caixeiros viajantes por meio do método de geração de colunas. O método foi aplicado com sucesso pelos autores a um problema real com cerca de 2500 solicitações.

Toth e Vigo (1996) descrevem uma *Heurística de Inserção Paralela* para o problema com múltiplos veículos e restrições de janelas de tempo. Também é apresentado um método de melhoria da solução obtida, por meio de procedimentos de busca local. Tais procedimentos são divididos em duas categorias: trocas intra-rotas e trocas inter-rotas. A primeira categoria corresponde a tentativas de melhoria das rotas pela modificação da seqüência dos locais visitados dentro de cada rota. Já a segunda categoria considera a possibilidade de troca de requisições de transporte entre rotas distintas. A heurística foi testada com sucesso pelos autores em um conjunto de cerca de 1500 solicitações provenientes de uma situação real. Um aprimoramento desse método de melhoria da solução foi desenvolvido por Toth e Vigo (1997), com a inclusão de um método de busca local derivado da busca tabu. Esse refinamento provou ser capaz de melhorar os resultados obtidos pela versão original do processo de melhoria.

Baugh jr. et al. (1998) utilizam o *Simulated Annealing* para resolver o DARP. Nesse trabalho, é utilizada uma abordagem de “agrupar-primeiro” e “rotear-depois”. Nessa abordagem, os clientes são inicialmente divididos em grupos (*clusters*), e depois para cada grupo é determinada a ordem de atendimento de seus clientes. O agrupamento é feito pelo *Simulated Annealing* e o roteamento por uma heurística do vizinho mais próximo. Os resultados foram obtidos para instâncias com até 300 clientes, e os tempos de processamento para obtenção desses resultados não são apresentados no artigo.

Znamensky e Cunha (1998) tratam o DARP através de uma adaptação da *Heurística de Inserção Paralela* proposta por Madsen et al. (1995), seguida de uma etapa de melhoria das rotas obtida similar à apresentada por Toth e Vigo (1996). Nesse trabalho, é tratado especificamente o atendimento de idosos e deficientes por veículos de pequena capacidade (peruas ou vans), e as instâncias utilizadas representam um caso real extraído da operação do ATENDE, serviço de transporte de deficientes gerido pela São Paulo Transporte S/A. Essas instâncias correspondem a um dia típico do mês de novembro de 1998, e perfazem 349 requisições de transporte (clientes) e 84 veículos disponíveis, localizados em 47 garagens distribuídas no município de São Paulo.

3.3 Abordagens recentes

Jih et al. (2002) utilizam um *Algoritmo Genético* para resolver um problema de coleta e entrega com janelas de tempo e apenas um veículo. Nesse algoritmo, a representação de uma rota é definida através de um cromossomo que representa a seqüência de locais a serem atendido pelo veículo. O algoritmo ainda permite a exploração de soluções inválidas durante a busca. A função objetivo trata apenas os custos de viagem do veículo e as violações nas restrições. Nesse trabalho, são utilizadas instâncias geradas aleatoriamente com até 100 clientes, sendo o tempo de processamento necessário para resolvê-las de aproximadamente 38 minutos.

Cordeau e Laporte (2003b) apresentam uma abordagem que utiliza a metaheurística *Busca Tabu* para resolver o DARP. Nessa abordagem o problema é tratado na sua forma estática, com múltiplos veículos, frota homogênea e garagem única. Além disso, são comparados três métodos heurísticos, *P1*, *P2* e *P3*, que, juntamente com a Busca Tabu,

são utilizados para resolver o problema. O primeiro (*P1*) visa apenas à minimização das violações nas janelas de tempo, enquanto o segundo (*P2*), além de tentar minimizar essas violações, visa também à minimização da duração rotas. Já o terceiro (*P3*) tem esses mesmos objetivos, porém visando também a redução nos tempos de viagem dos clientes. Os resultados obtidos mostram que *P1* e *P2*, por serem mais “simples”, são mais rápidos que *P3*, porém as soluções obtidas por *P3* são expressivamente melhores. Todos os resultados são obtidos a partir de instâncias geradas aleatoriamente baseadas em informações cedidas pela *Montreal Transit Commission (MTC)* de Montreal – Canadá. Essas instâncias variam de 24 a 144 clientes. Além disso, eles apresentam outros resultados obtidos para instâncias de problemas reais (com 200 e 295 clientes) cedidas por uma transportadora dinamarquesa.

Bergvinsdottir et al. (2004) apresentam uma abordagem que utiliza um *Algoritmo Genético* para resolver o DARP. Nessa abordagem o problema é tratado na sua forma estática, com múltiplos veículos, frota heterogênea e garagens múltiplas. A resolução do problema é obtida através de uma abordagem clássica de “agrupar primeiro” e “rotear depois”. O Algoritmo Genético é utilizado para agrupar (distribuir) os clientes nos veículos, ou seja, determinar quais serão os clientes atendidos por cada veículo. Já o roteamento (seqüência de atendimento) e a programação (determinação dos horários) são determinados, independentemente para cada veículo, por uma heurística específica. Os resultados são obtidos a partir das instâncias geradas aleatoriamente por Cordeau e Laporte (2003b). Essa mesma abordagem é apresentada com maiores detalhes em Bergvinsdottir (2004).

Cordeau (2004) apresenta uma abordagem que utiliza o algoritmo *Branch-and-Cut* para resolver o problema. Nessa abordagem o problema é tratado na sua forma estática, com múltiplos veículos, frota homogênea e garagem única. A resolução do problema é dada através de uma abordagem exata, garantindo assim a obtenção da solução ótima para o problema. Os resultados são obtidos a partir de instâncias geradas aleatoriamente com no máximo 32 clientes.

3.4 Considerações finais

Analisando os trabalhos descritos neste capítulo, nota-se que, além dos diferentes métodos para resolução do problema, existem inúmeras variações na formulação do DARP em si. Algumas dessas variações dizem respeito à definição de frota múltipla ou veículo único, de frota homogênea ou heterogênea, de múltiplas garagens ou garagem única, e principalmente de diversas formulações para a função objetivo que se deseja otimizar.

Além disso, pode-se notar que, assim como pode ser visto em Cordeau (2004), os métodos exatos só foram capazes de resolver problemas pequenos (com menos de 100 clientes), o que é bem distante das situações reais, onde a demanda por este tipo de serviço pode chegar facilmente a algumas centenas de clientes. Tal comportamento não é observado nos métodos heurísticos, que se mostram capazes de resolver com certa rapidez problemas maiores, sendo inclusive utilizados em sistemas de roteirização e programação em tempo real, como é o caso apresentado por Madsen et al. (1995).

Nota-se também um grande número de heurísticas desenvolvidas especificamente para o problema de roteirização e programação de veículos, e que são freqüentemente utilizadas, na maioria das vezes com sucesso, em problemas correlatos.

Quanto às instâncias utilizadas nos trabalhos descritos neste capítulo, vale ressaltar que a maioria é obtida através de empresas que atuam no setor, que na maioria dos casos não as disponibilizam publicamente. Quanto aos trabalhos descritos na Seção 3.3 especificamente, nota-se o atual estado-da-arte em que o problema se encontra, sendo esses tomados assim como base para o desenvolvimento deste trabalho.

Por fim, algumas dessas abordagens descritas na Seção 3.3 ainda são utilizadas em comparação com a abordagem proposta neste trabalho (ver CAPÍTULO 5).

CAPÍTULO 4

METODOLOGIA PROPOSTA

4.1 Considerações iniciais

O modelo proposto neste trabalho trata o DARP de forma estática (todas as requisições são conhecidas a priori), com múltiplos veículos, com frota heterogênea (cada veículo possui sua uma capacidade distinta) e garagens múltiplas (cada veículo começa e termina sua rota em garagens específicas). Esta é uma abordagem que abrange diversas outras encontradas na literatura (ver Bergvinsdottir (2004), Bergvinsdottir et al. (2004), Cordeau e Laporte (2003a), Cordeau e Laporte (2003b), etc...), e é bem próxima às situações reais do problema.

Assume-se inicialmente a existência de n clientes (requisições de transporte) a serem atendidos por m veículos. Cada requisição de transporte especifica um local de embarque i e um de desembarque $n+i$. Para representar o problema então, são definidos os seguintes conjuntos:

- K : conjunto dos veículos disponíveis ($|K| = m$).
- G^- : conjunto de garagens de origem.
- G^+ : conjunto de garagens de destino.
- P : conjunto dos locais de embarque.
- U : conjunto dos locais de desembarque.
- $N = G^- \cup P \cup U \cup G^+$: conjunto com todos os locais (pontos).

Cada cliente i ($\forall i \in P$) especifica a “carga” q_i necessária para seu transporte, ou seja, o número de assentos no veículo que ele irá ocupar, e também dois intervalos de horários, um em que ele gostaria de embarcar na sua origem $[e_i, l_i]$, e outro que ele gostaria de desembarcar no seu destino $[e_{n+i}, l_{n+i}]$ (janelas de tempo). A carga q_i deverá ser um valor

positivo nos locais de embarque e o mesmo valor, porém negativo, nos respectivos locais de desembarque.

Cada veículo k ($\forall k \in K$) tem uma capacidade conhecida Q''_k (quantidade de assentos disponíveis), um tempo máximo de duração associado a sua rota T''_k , e começa sua rota em uma garagem específica g^-_k e termina em uma outra g^+_k , podendo ser a mesma ou não. Todas as garagens ainda têm suas próprias janelas de tempo para partida (garagem de origem) e chegada (garagem de destino).

A cada cliente i ($\forall i \in P$) é associado um tempo máximo de viagem R''_i , ou seja, o tempo máximo em que o cliente poderá ficar dentro do veículo. A cada local i ($\forall i \in \{P \cup U\}$) é associado um tempo máximo de espera W''_i , ou seja, o tempo máximo em que os veículos poderão ficar esperando até iniciar o “serviço”. Considera-se como serviço o embarque ou desembarque de um cliente em um determinado local, sendo atribuído um tempo s_i necessário para sua completude. Por fim, dados os locais referentes às garagens e aos pontos pertencentes às requisições, têm-se as distâncias d_{ij} e as durações das viagens t_{ij} entre os pontos i e j , $\forall i, j \in N$ e $i \neq j$.

O objetivo então é minimizar os custos operacionais e a “insatisfação” dos clientes, ou seja, minimizar os requisitos “não-essenciais” do problema. Estes requisitos estão relacionados à distância total percorrida pelos veículos, ao número de veículos utilizados na solução do problema, à duração das rotas, ao tempo de viagem dos clientes e ao tempo de espera nos locais de embarque e desembarque.

Existe ainda um conjunto de requisitos essenciais que devem ser obedecidos, ou seja, requisitos que obrigatoriamente devem ser atendidos para que se obtenha uma solução válida (viável em termos práticos) para o problema. Dentre esses requisitos estão:

- A duração da rota executada pelo veículo k ($\forall k \in K$) não deve exceder o tempo máximo permitido T''_k .
- O tempo de viagem do cliente i ($\forall i \in P$) não deve exceder o tempo máximo de viagem permitido R''_i ;

- O tempo de espera no local i ($\forall i \in \{P \cup U\}$) não deve exceder o tempo máximo de espera permitido W''_i ;
- A capacidade Q''_k ($\forall k \in K$) dos veículos não pode ser excedida em nenhum local;
- O início do serviço em todos os locais i ($\forall i \in N$) deve estar dentro dos intervalos pré-estabelecidos $[e_i, l_i]$.

4.2 Formulação matemática

De posse das informações descritas na seção anterior, pode-se definir as variáveis de decisão que irão fornecer toda a programação dos veículos. Considera-se A_i como sendo o horário de chegada no local i ($\forall i \in N$) pelo veículo que irá atender tal local, sendo $A_i = 0$ se $i \in G^-$ e $A_i = D_{i-1} + t_{i-1,i}$ se $i \in \{P \cup U \cup G^+\}$; D_i indica o horário de partida do local i ($\forall i \in N$), novamente pelo veículo que irá atendê-lo, sendo $D_i = 0$ se $i \in G^+$ e $D_i = B_i + s_i$ se $i \in \{P \cup U\}$ e $D_i = B_i$ se $i \in \{G^-\}$; B_i representa o horário de início do serviço no local i ($\forall i \in N$), sendo $B_i = D_i$ se $i \in G^-$ e $B_i = \max\{e_i, A_i\}$ se $i \in \{P \cup U \cup G^+\}$; o tempo de espera antes do início do serviço no local i ($\forall i \in N$) é dado por W_i , sendo $W_i = 0$ se $i \in G^-$ e $W_i = B_i - A_i$ se $i \in \{P \cup U \cup G^+\}$; Q_i indica a carga (número de assentos ocupados) do veículo que atende o local i ($\forall i \in N$), após o término do serviço, sendo $Q_i = 0$ se $i \in \{G^- \cup G^+\}$ e $Q_i = Q_{i-1} + q_i$ se $i \in \{P \cup U\}$; e por fim, considera-se R_i como sendo o tempo de viagem do cliente i ($\forall i \in P$), sendo $R_i = B_{n+i} - D_i$.

A partir de então, considerando $x^k_{ij} = 1$ caso o veículo k se desloque do local i para o local j e $x^k_{ij} = 0$ caso contrário, tem-se o seguinte modelo matemático:

Minimizar:

$$\omega_0 \sum_{k \in K} \sum_{i \in N} \sum_{j \in N; j \neq i} (d_{i,j} x^k_{i,j}) + \omega_1 \sum_{k \in K} \sum_{j \in P} x^k_{g_k^-, j} + \omega_2 \sum_{k \in K} (B_{g_k^+} - D_{g_k^-}) + \omega_3 \sum_{i \in P} R_i + \omega_4 \sum_{i \in \{P \cup U\}} W_i + \quad (1)$$

$$\beta_0 \sum_{k \in K} \max\{0, (B_{g_k^+} - D_{g_k^-}) - T''_k\} + \beta_1 \sum_{i \in P} \max\{0, R_i - R''_i\} + \beta_2 \sum_{i \in \{P \cup U\}} \max\{0, W_i - W''_i\} + \quad (2)$$

$$\beta_3 \sum_{k \in K} \max\{0, \left(Q_i \sum_{i \in \{P \cup U\}} \sum_{j \in \{P \cup U\}; j \neq i; j \neq n-i} x_{i,j}^k \right) - \tilde{Q}_k\} + \beta_4 \sum_{i \in N} (\max\{0, e_i - B_i\} + \max\{0, B_i - l_i\}) \quad (3)$$

Sujeito à:

$$\sum_{j \in \{P \cup \{g_k^+\}\}} x_{g_k^+, j}^k = 1 \quad \forall k \in K \quad (4)$$

$$\sum_{i \in \{U \cup \{g_k^+\}\}} x_{i, g_k^+}^k = 1 \quad \forall k \in K \quad (5)$$

$$\sum_{k \in K} \sum_{j \in \{P \cup U\}; j \neq i} x_{i,j}^k = 1 \quad \forall i \in P \quad (6)$$

$$\sum_{j \in \{P \cup U\}; j \neq i} x_{i,j}^k - \sum_{j \in \{P \cup U \cup \{g_k^+\}\}; j \neq i; j \neq n+i} x_{n+i,j}^k = 0 \quad \forall k \in K; i \in P \quad (7)$$

$$\sum_{j \in \{P \cup U \cup \{g_k^+\}\}; j \neq i; j \neq n+i} x_{j,i}^k - \sum_{j \in \{P \cup U\}; j \neq i} x_{i,j}^k = 0 \quad \forall k \in K; i \in P \quad (8)$$

$$\sum_{j \in \{P \cup U\}; j \neq i} x_{j,i}^k - \sum_{j \in \{P \cup U \cup \{g_k^+\}\}; j \neq i; j \neq n-i} x_{i,j}^k = 0 \quad \forall k \in K; i \in U \quad (9)$$

$$B_j = (B_i + s_i + t_{i,j} + W_j) \sum_{k \in K} x_{i,j}^k \quad \forall i, j \in N; i \neq j \quad (10)$$

$$Q_j = (Q_i + q_j) \sum_{k \in K} x_{i,j}^k \quad \forall i, j \in N; i \neq j \quad (11)$$

$$A_i = B_i - W_i \quad \forall i \in \{P \cup U \cup G^+\} \quad (12)$$

$$D_i = B_i + s_i \quad \forall i \in \{P \cup U \cup G^-\} \quad (13)$$

$$R_i = B_{n+i} - D_i \quad \forall i \in P \quad (14)$$

$$A_{g_k^-} = D_{g_k^+} = Q_{g_k^-} = Q_{g_k^+} = W_{g_k^-} = 0 \quad \forall k \in K \quad (15)$$

$$A_i, W_i, B_i, D_i, Q_i \text{ irrisritas} \quad \forall i \in N \quad (16)$$

$$R_i \text{ irrisrita} \quad \forall i \in P \quad (17)$$

$$x_{i,j}^k \in \{0,1\} \quad \forall k \in K; \forall i, j \in N; i \neq j \quad (18)$$

A função objetivo é dividida em duas partes (equações (1) e equações (2) e (3)). A primeira (1) visa minimizar os requisitos não-essenciais do problema, enquanto a segunda (2) e (3) visa à minimização das violações nos requisitos essenciais. Além disso, os requisitos não-essenciais são “penalizados” através de um vetor de números

inteiros positivos (pesos) $\omega = [\omega_0, \omega_1, \omega_2, \omega_3, \omega_4]$, e os essenciais com um vetor semelhante $\beta = [\beta_0, \beta_1, \beta_2, \beta_3, \beta_4]$.

As restrições (4) e (5) garantem, respectivamente, que cada veículo partirá de sua garagem de origem e chegará à sua garagem de destino uma única vez, ou seja, cada rota será iniciada em sua garagem de origem e terminará em sua garagem de destino. A restrição (6) garante que cada cliente será atendido uma única vez, ou seja, por apenas um veículo. A restrição (7) garante que um local de embarque estará sempre na mesma rota que seu respectivo local de desembarque, enquanto que a contenção do fluxo, ou seja, tudo que entra é igual a tudo que sai, é garantida pelas restrições (8) e (9).

A restrição (10) determina o horário de início do serviço e o tempo de espera em cada local e também o veículo que o atenderá. Como mostrado em Cordeau (2004), essas restrições, em conjunto, garantem a eliminação de sub-rotas (*sub-tours*) no caso do DARP. A restrição (11) determina a carga dos veículos em cada local, enquanto as restrições (12), (13) e (14) garantem, respectivamente, um cálculo correto dos horários de chegada e partida nos locais e dos tempos de viagem dos clientes. A restrição (15) “inicializa” algumas variáveis referentes às garagens. Por fim, a restrição (18) apenas garante que as variáveis de decisão x_{ij}^k sejam binárias, enquanto as restrições (16) e (17) garantem que as demais variáveis sejam irrestritas.

4.3 Método de solução

O *Simulated Annealing* - SA é um método de busca local que aceita movimentos de piora para escapar de ótimos locais. Ele foi proposto originalmente por Kirkpatrick et al. (1983), e se fundamenta em uma analogia com a termodinâmica ao simular o resfriamento de um conjunto de átomos aquecidos. Esta técnica começa sua busca a partir de uma solução inicial qualquer.

O procedimento principal consiste em um laço que gera aleatoriamente, em cada iteração, um único vizinho S' da solução corrente S . A cada geração de um vizinho S' de S , é testada a variação do valor da função objetivo (custo), isto é, $\Delta = f(S') - f(S)$.

Para um problema de minimização, se $\Delta < 0$, o método aceita a solução e S' passa a ser a nova solução corrente. Caso $\Delta \geq 0$ a solução vizinha candidata também poderá ser aceita, mas neste caso, com uma probabilidade $e^{-\Delta/T}$, onde T é um parâmetro do método, chamado de temperatura, que regula a probabilidade de aceitação de soluções de pior custo.

A temperatura T assume, inicialmente, um valor elevado T_0 . Após um número fixo de iterações: S_{Amax} (o qual representa o número de iterações necessárias para o sistema atingir o equilíbrio térmico em uma dada temperatura), a temperatura é gradativamente diminuída por uma razão de resfriamento α ($0 < \alpha < 1$). Com esse procedimento, no início, dá-se uma chance maior para escapar de mínimos locais e, à medida que T se aproxima de zero, o algoritmo comporta-se como o método de descida, uma vez que diminui a probabilidade de se aceitar movimentos de piora ($T \rightarrow 0 \Rightarrow e^{-\Delta/T} \rightarrow 0$).

O procedimento pára quando a temperatura chega a um valor próximo de zero (temperatura de congelamento: T_c) e nenhuma solução que piore o valor da melhor solução é mais aceita, isto é, quando o sistema está estável. A solução obtida quando o sistema encontra-se nesta situação evidencia o encontro de um mínimo local, que em alguns casos também pode representar um mínimo global. Os parâmetros de controle do procedimento são a razão de resfriamento α , o número de iterações para cada temperatura S_{Amax} , a temperatura inicial T_0 e a temperatura de congelamento T_c .

Para a utilização do SA, deve-se definir a priori, um método para geração de uma solução inicial S , um método para geração das soluções vizinhas S' (estrutura de vizinhança), e uma função objetivo $f(S)$ a ser otimizada.

4.3.1 Geração da solução inicial

Neste trabalho, a solução inicial é gerada através de uma *heurística de distribuição*, que é responsável pela roteirização dos veículos, ou seja, pela formação dos agrupamentos dos locais nas rotas e da seqüência de atendimento destes. Essa heurística é apresentada na Figura 4.1.

Nessa heurística, são criadas inicialmente m rotas vazias, sendo atribuído, a cada uma delas, um veículo específico. Posteriormente, todas as requisições de transporte dos clientes (pontos de embarque e seus respectivos pontos de desembarque) são distribuídas aleatoriamente, porém de forma uniforme, a essas rotas, ou seja, as n requisições são divididas igualmente entre os m veículos. Obviamente em alguns casos a divisão de n por m não será inteira, e nesses casos a última rota deverá atender a um número superior de requisições. A seleção das requisições e das rotas que as atenderão também é aleatória.

- | |
|--|
| <ol style="list-style-type: none"> 1. <u>CRIAR</u> (m rotas vazias, e atribuí-las aos m veículos); 2. <u>CRIAR</u> (uma lista L com todas as requisições de transporte dos clientes); 3. <u>PARA</u> (cada rota k, $k = 1, 2, \dots, m$) <u>FACA</u> 4. <u>SELECIONAR</u> ($\lfloor n/m \rfloor$ requisições de L); 5. <u>PARA</u> (cada requisição selecionada) <u>FACA</u> 6. Pos1 \leftarrow posição qualquer da rota k; 7. Pos2 \leftarrow posição qualquer da rota k, porém posterior a Pos1; 8. <u>INSERIR</u> (o ponto de embarque em Pos1); 9. <u>INSERIR</u> (o ponto de desembarque em Pos2); 10. <u>FIM-PARA</u>; 11. <u>INSERIR</u> (a garagem de origem do veículo k no início da rota k); 12. <u>INSERIR</u> (a garagem de destino do veículo k no fim da rota k); 13. <u>REMOVER</u> (as requisições selecionadas de L); 14. <u>FIM-PARA</u>; |
|--|

Figura 4.1: Heurística de distribuição.

As posições de inserção dos pontos (locais) na rota também são selecionadas de forma aleatória, mas sempre atendendo a restrição de precedência oriunda da requisição de transporte do cliente em questão, ou seja, um local de embarque sempre deverá ser anterior ao seu respectivo local de desembarque. Por fim, devido ao fato de toda rota começar e terminar em garagens específicas (referentes ao veículo que irá executá-la), pode-se considerar que os locais referentes a estas garagens sempre serão alocados como primeiro e último pontos em cada uma das rotas criadas.

Como descrito anteriormente, essa heurística trata apenas da roteirização dos veículos, entretanto ainda deve-se fazer a programação destes, ou seja, determinar os horários de chegada nos locais, os horários de partida, etc. Para isso, é utilizada uma outra heurística, denominada *heurística de programação* (Figura 4.2), que é adaptada da apresentada em Cordeau e Laporte (2003b), e realiza a programação de uma forma a

tentar reduzir as violações nas janelas de tempo, a duração das rotas e os tempos de viagem dos clientes.

Nessa heurística, é utilizado um conceito de “atraso”, proposto inicialmente por Savelsbergh (1992), que consiste basicamente em atrasar, na medida do possível, o horário de partida da garagem de origem e o início do serviço nos locais de embarque.

1. $B_0 \leftarrow e_0; D_0 \leftarrow B_0;$
2. CALCULAR (A_i, B_i, W_i, D_i, Q_i para cada ponto $v_i \in V_k$ e $v_i \neq v_0$);
3. CALCULAR (F_0);
4. $B_0 \leftarrow e_0 + \min \left\{ F_0, \sum_{0 < p \leq z} \bar{W}_p \right\}; D_0 \leftarrow B_0;$
5. ATUALIZAR (A_i, B_i, W_i, D_i para cada ponto $v_i \in V_k$ e $v_i \neq v_0$);
6. CALCULAR (R_i para cada ponto $v_i \in V_k$ e $v_i \in P$);
7. PARA (cada ponto $v_i \in V_k$ e $v_i \in P$) FAÇA
8. CALCULAR (F_i);
9. $B_i \leftarrow B_i + \min \left\{ F_i, \sum_{i < p \leq z} \bar{W}_p \right\};$
10. $D_i \leftarrow B_i + s_i; W_i \leftarrow B_i - A_i;$
11. ATUALIZAR (A_j, B_j, W_j, D_j para cada ponto $v_j \in V_k$ e v_j posterior a v_i);
12. ATUALIZAR (R_j para cada ponto $v_j \in V_k, v_j \in P$ e v_{n+j} posterior a v_i);
13. FIM-PARA;

Figura 4.2: Heurística de programação.

Inicialmente, o horário de partida da garagem de origem é marcado para o horário de início da respectiva janela de tempo. A partir de então, os demais cálculos (horários de chegada, início do serviço, tempo de espera, horário de partida e carga do veículo) são realizados para todos os pontos seguintes na rota. Posteriormente é calculado o atraso para a partida da garagem de origem, e então o horário de partida é ajustado de forma a não aumentar as violações nas janelas de tempo. É feita então uma atualização nos horários para todos os pontos posteriores a garagem de origem, e também é calculado o tempo de viagem dos clientes. Por fim, para cada ponto de embarque pertencente à rota em questão, é calculado o seu respectivo atraso (sendo o horário de início do serviço ajustado de forma a reduzir a duração da rota e o tempo de viagem dos clientes e não aumentar as violações nas janelas de tempo), e os horários de todos os pontos posteriores são atualizados, assim como o tempo de viagem dos clientes cujo desembarque é posterior ao ponto de embarque em questão.

Considerando então uma rota $V_k = \{v_0, v_b, v_j, v_{n+b}, v_{n+b}, \dots, v_z\}$, executada pelo veículo k ($\forall k \in K$), tem-se que v_0 e v_z representam, respectivamente, as garagens de origem e de destino de k ($v_0 \in G^-$; $v_z \in G^+$), e os demais pontos representam os pontos de embarque e desembarque, sendo que o ponto v_{n+i} representa o ponto de desembarque referente ao ponto de embarque v_i , ou seja $v_i \in P$ e $v_{n+i} \in U$. Para uma rota V_k qualquer, o atraso é calculado da maneira apresentada nas equações (19), (20) e (21), e sua programação é ilustrada na Figura 4.2.

$$F_i = \min_{i \leq j \leq z} \left\{ \sum_{i < p \leq j} W_p + \left(\min\{l_j - B_j, \tilde{R}_i - R_j\} \right)^+ \right\} \quad \text{se } i \in P \quad (19)$$

$$F_i = \min_{i \leq j \leq z} \left\{ \sum_{i < p \leq j} W_p + (l_j - B_j) \right\} \quad \text{se } i \in G^- \quad (20)$$

$$F_i = 0 \quad \text{se } i \in \{U \cup G^+\} \quad (21)$$

4.3.2 Estrutura de vizinhança

Como estrutura de vizinhança, foram utilizados três diferentes movimentos de troca: *Re-ordenar rota*, *Re-alocar ponto* e *Trocar pontos*. Esses movimentos são baseados em outros encontrados freqüentemente nos trabalhos referentes ao DARP (ver Bergvinsdottir (2004), Bergvinsdottir et al. (2004), Cordeau e Laporte (2003b), Hart (1996) e Savelsbergh (1992)).

É interessante destacar que nesses movimentos não são consideradas as garagens, pois essas são “fixas” em todas as rotas, e conseqüentemente suas posições não poderão ser alteradas.

O movimento *Re-ordenar rota* consiste basicamente em selecionar uma rota qualquer pertencente à solução, selecionar um ponto qualquer nessa rota, selecionar uma nova posição para esse ponto e trocar sua posição para a nova. Esse movimento é ilustrado na Figura 4.3.

Como é selecionado apenas um ponto na rota, este pode ser de embarque ou desembarque. No primeiro caso (Figura 4.3a), a nova posição deverá, obrigatoriamente,

ser **anterior** ao seu respectivo ponto de desembarque. Já no segundo caso (Figura 4.3b), a nova posição deverá ser **posterior** ao respectivo ponto de embarque. Esses “limites” são apresentados através das linhas pontilhadas na Figura 4.3.

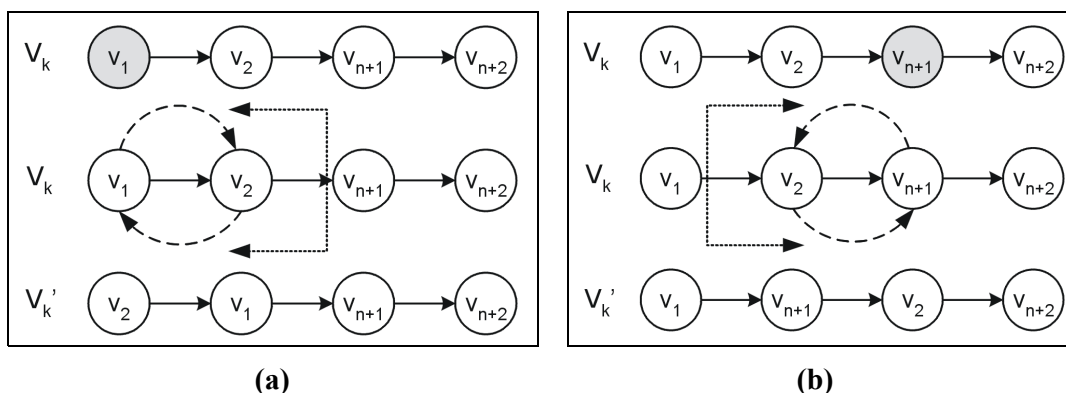


Figura 4.3: Movimento *Re-ordenar rota*.

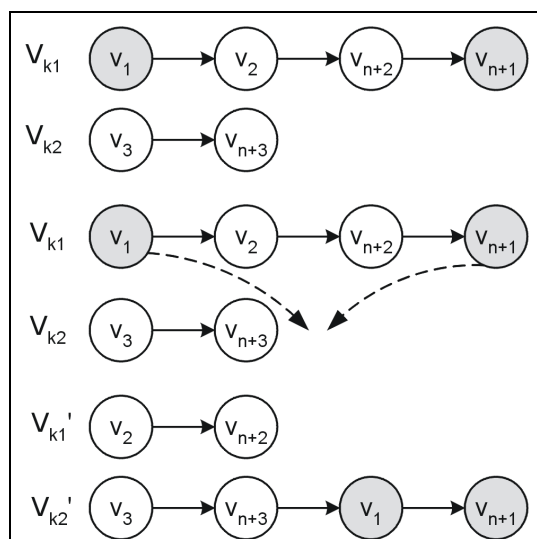


Figura 4.4: Movimento *Re-alocar pontos*.

O movimento *Re-alocar pontos* consiste basicamente em também selecionar duas rotas quaisquer pertencentes à solução, selecionar uma requisição qualquer em apenas uma das duas rotas, extraí-la (seus pontos de embarque e desembarque) de sua rota de origem e adicioná-la na outra rota, em posições quaisquer. O ponto de embarque e seu respectivo ponto de desembarque são extraídos simultaneamente, porém sua inserção na “outra” rota pode ser feita de forma separada, ou seja, esses pontos são alocados

individualmente em qualquer posição da rota, porém sempre mantendo a condição de que o ponto de embarque fique antes do ponto de desembarque (restrição de precedência). Esse movimento é ilustrado na Figura 4.4.

Já o movimento *Trocar pontos* consiste basicamente em selecionar duas rotas quaisquer pertencentes à solução, selecionar uma requisição de transporte (ponto de embarque e seu respectivo ponto de desembarque) qualquer em cada uma das duas rotas, e trocá-las. Esse movimento é ilustrado na Figura 4.5.

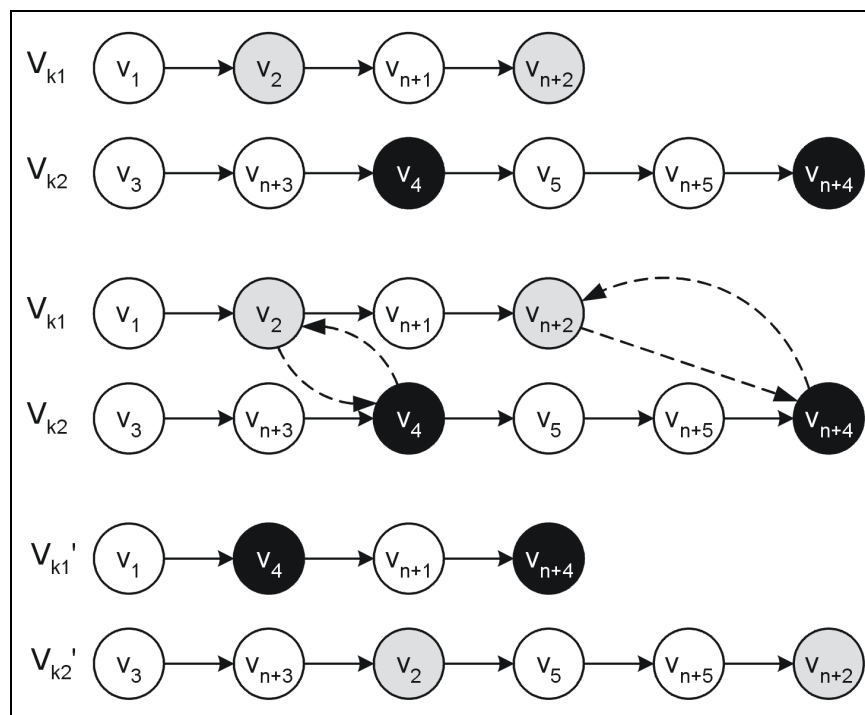


Figura 4.5: Movimento *Trocar pontos*.

Nesse caso, como são trocadas requisições de transporte, ou seja, seus pontos de embarque e desembarque são trocados simultaneamente (trocados em pares), pode-se garantir que o ponto de embarque será sempre anterior ao seu respectivo ponto de desembarque.

O SA foi implementado de uma forma em que cada solução vizinha é gerada por apenas um desses movimentos, sendo a sua escolha feita de forma aleatória, porém

uniformemente distribuída, possibilitando assim uma boa diversidade entre as soluções intermediárias geradas, e conseqüentemente uma boa exploração do espaço de soluções.

```

1. DADO ( $\alpha$ ,  $S_{Amax}$ ,  $T_0$  e  $T_C$ ) FACA
2. GERAR (uma solução  $S$  através da heurística de distribuição);
3. APLICAR (a heurística de programação em todas as rotas de  $S$ );
4.  $S^* \leftarrow S$ ; {Melhor solução obtida até então}
5.  $IterT \leftarrow 0$ ; {Número de iterações na temperatura  $T$ }
6.  $T \leftarrow T_0$ ; {Temperatura corrente}
7. ENQUANTO ( $T > T_C$ ) FACA
8. ENQUANTO ( $IterT < S_{Amax}$ ) FACA
9.  $IterT \leftarrow IterT + 1$ ;
10. GERAR (um vizinho qualquer  $S'$  através de um dos mov. de troca);
11. APLICAR (a heurística de programação em todas as rotas de  $S'$ );
12.  $\Delta \leftarrow f(S') - f(S)$ ;
13. SE ( $\Delta < 0$ )  $S \leftarrow S'$ ;
14. SE ( $f(S') < f(S^*)$ )  $S^* \leftarrow S'$ ; FIM-SE
15. SENÃO
16. TOMAR ( $x \in [0,1]$ );
17. SE ( $x < e^{-\Delta/T}$ )  $S \leftarrow S'$ ; FIM-SE
18. FIM-SE
19. FIM-ENQUANTO
20.  $T \leftarrow \alpha * T$ ;  $IterT \leftarrow 0$ ;
21. FIM-ENQUANTO
22.  $S \leftarrow S^*$ ;
23. RETORNAR ( $S$ );

```

Figura 4.6: Algoritmo *Simulated Annealing* implementado.

A função objetivo $f(S)$ utilizada para avaliar as soluções é aquela descrita pelas equações (1), (2) e (3) (ver Seção 4.2), e as restrições apresentadas no modelo proposto na Seção 3 ((4) a (18)) são atendidas implicitamente nas heurísticas de distribuição e programação e nos movimentos de troca aqui descritos. Um pseudo-código do *SA* implementado é apresentado na Figura 4.6.

4.4 Considerações finais

A formulação matemática apresentada neste capítulo pode ser facilmente utilizada para solucionar diversos outros modelos já conhecidos para o DARP, como por exemplo, modelos com garagem única, com apenas um veículo, com frota homogênea, e o mais interessante, com vários objetivos distintos. Essa flexibilidade na escolha do objetivo final é dada pela utilização de uma função multi-objetivo, que utiliza pesos para determinar a “importância” de cada requisito na solução do problema.

É interessante destacar também que essa formulação é baseada em várias outras encontradas na literatura (ver Bergvinsdottir (2004), Bergvinsdottir et al. (2004), Cordeau (2004) e Cordeau e Laporte (2003b)), e como em algumas dessas, ela representa o problema de forma “relaxada”, tratando os requisitos essenciais como parte da função a ser minimizada, e não como restrições do modelo. Essa é uma abordagem interessante que torna a resolução do problema menos árdua, porém permite que soluções inválidas sejam obtidas. Entretanto, soluções inválidas podem ser interessantes em alguns casos, pois como é sabido (Cordeau, 2004), os modelos exatos são incapazes de resolver de forma satisfatória um problema de tamanho “real”, e em alguns casos, uma solução com violações mínimas nos requisitos essenciais pode ser aceitável.

Como sugerido em Cordeau (2004), nesse modelo é feita uma redução no número de arcos (caminhos) existentes entre os locais (como pode ser visto na formulação matemática). Tal fato se deve às características do DARP, que por exemplo, não permite um arco ligando um local a si próprio, não permite que um desembarque seja realizado antes de seu respectivo embarque, etc. Além disso, ao contrário da formulação proposta por Bergvinsdottir et al. (2004), que utiliza as variáveis referentes aos locais de forma exclusiva para cada veículo $(A^k_i, B^k_i, D^k_i, W^k_i, Q^k_i, R^k_i)$, nessa formulação são utilizadas variáveis únicas $(A_i, B_i, D_i, W_i, Q_i, R_i)$, o que reduz significativamente o número de variáveis de decisão do modelo.

A escolha do *Simulated Annealing* se deve aos excelentes resultados obtidos para diversos problemas de otimização (ver Baugh jr. et al. (1998), Biajoli et al. (2003), Hart (1996) e Mauri (2003)), e também à sua simplicidade. Já a escolha das demais heurísticas e da estrutura de vizinhança, foi baseada em outras aplicações das mesmas ao DARP.

Por fim, deve-se ressaltar que o software CPLEX 9.1 (ver ILOG (2005)), que é uma referência para resolver problemas de otimização, não é capaz de resolver esse modelo na sua forma exata (tratando os requisitos essenciais como restrições do modelo) para as instâncias utilizadas neste trabalho.

CAPÍTULO 5

EXPERIÊNCIA COMPUTACIONAL

5.1 Considerações iniciais

Visando avaliar o potencial da abordagem apresentada neste trabalho, foram realizados vários experimentos nos quais foram utilizadas as instâncias apresentadas por Cordeau e Laporte (2003b) (disponíveis em: <http://www.hec.ca/chairedistributique/data/darp/>). Essas instâncias são referências nos trabalhos mais atuais e que têm considerável importância para resolução do DARP (ver Bergvinsdottir (2004), Bergvinsdottir et al. (2004), Cordeau e Laporte (2003b), etc.). Estão disponíveis 20 instâncias, combinadas entre 24 a 144 requisições de transporte (48 a 288 pontos) e 3 a 13 veículos, sendo as primeiras instâncias (R1a – R10a) formadas por janelas de tempo “largas” e o restante (R1b – R10b) formado por janelas de tempo mais “estreitas”.

Essas instâncias representam problemas com garagem única e frota homogênea, e não adotam o conceito de tempo máximo de espera. Entretanto, o modelo aqui proposto (Seção 4.2) se adapta facilmente a elas. Para isso, basta tratar essa única garagem como garagem de origem e destino de todos os veículos, e um valor muito grande como tempo máximo de espera (que irá eliminar a possibilidade de ocorrer qualquer violação).

Os parâmetros utilizados pelo SA, em todos os experimentos foram $T_0 = 20000$, $\alpha = 0.975$, $T_c = 0.01$ e $S_{Amax} = 1000$. Já a escolha dos valores das penalizações (vetores ω e β) aplicadas na função objetivo ((1), (2) e (3)) foi baseada em uma análise apresentada por Bergvinsdottir et al. (2004). Porém, neste trabalho, a penalização para as violações nos requisitos essenciais foi mais “pesada”, visando evitar a obtenção de soluções inválidas para o problema. Os pesos utilizados foram: $\omega = [8,0,1,3,1]$ e $\beta = [1500,1500,1500,1500,1500]$.

5.2 Resultados

Para validação do modelo proposto, e da aplicação do SA ao DARP, foram realizados 5 testes para cada instância. A Tabela 5.1 apresenta um resumo dos resultados obtidos ($f(S)$ é o valor da função objetivo obtida). Nessa tabela, a coluna *Melhor $f(S)$* indica o valor da função objetivo da melhor solução encontrada nos 5 testes (para cada instância). A coluna *$f(S)$ média* apresenta a média aritmética das 5 funções objetivo encontradas, e a coluna *Desvio* apresenta o “desvio” obtido entre esses testes (ver equação (22)).

$$\text{Desvio} = \left(\frac{f(S) \text{ média} - \text{Melhor } f(S)}{\text{Melhor } f(S)} \right) * 100 \quad (22)$$

Em todos os experimentos realizados, o número de veículos utilizados foi igual ao disponível, todas as requisições de transportes foram atendidas, e o mais importante, todas as soluções obtidas foram válidas, ou seja, todas as restrições essenciais foram satisfeitas.

Tabela 5.1: Resumo dos experimentos realizados.

Instância	Nº de veículos	Nº de requisições	$f(S)$ média	Melhor $f(S)$	Desvio (%)
R1a	3	24	3721,57	3677,91	1,19
R2a	5	48	7101,87	7017,34	1,20
R3a	7	72	11982,18	11873,76	0,91
R4a	9	96	13982,52	13725,92	1,87
R5a	11	120	16006,82	15736,66	1,72
R6a	13	144	20607,77	20465,39	0,70
R7a	4	36	5718,46	5610,05	1,93
R8a	6	72	11554,71	11343,19	1,86
R9a	8	108	17345,60	15632,09	10,96
R10a	10	144	23104,61	22430,00	3,01
R1b	3	24	3407,87	3379,74	0,83
R2b	5	48	5925,35	5889,56	0,61
R3b	7	72	11045,26	11006,12	0,36
R4b	9	96	12856,75	12807,87	0,38
R5b	11	120	14874,48	14544,13	2,27
R6b	13	144	18795,79	18518,82	1,50
R7b	4	36	5202,93	5136,37	1,30
R8b	6	72	10791,67	10703,17	0,83
R9b	8	108	15180,91	15013,71	1,11
R10b	10	144	20492,56	19969,15	2,62

Os melhores resultados obtidos (Tabela 5.2) ainda são comparados aos obtidos por Cordeau e Laporte (2003b), através da metaheurística Busca Tabu (Tabela 5.3), e por Bergvinsdottir et al. (2004), através de um Algoritmo Genético (Tabela 5.4). Apenas os resultados obtidos para algumas instâncias são comparados a essas referências, pois Bergvinsdottir et al. (2004) não apresenta os resultados para as demais.

Tabela 5.2: Melhores soluções obtidas pelo *Simulated Annealing*.

Instância	Distância percorrida	Duração das rotas (min)	Tempo de espera (min)		Tempo de viagem (min)		CPU (min)
			total	médio	total	médio	
R1a	252,79	831,3	98,51	2,05	241,93	10,08	1,00
R2a	437,45	1992,34	594,9	6,2	310,17	6,46	1,20
R3a	831,74	2404,67	132,93	0,92	894,08	12,42	1,46
R5a	1085,45	3920,25	434,81	1,81	899,35	7,49	1,79
R9a	1064,23	3258,66	34,42	0,16	1275,06	11,81	2,28
R10a	1392,09	4475,42	203,33	0,71	2204,85	15,31	2,72
R1b	251,85	738,42	6,57	0,14	206,66	8,61	0,92
R2b	436,69	1428,44	31,75	0,33	311,95	6,5	1,30
R5b	1010,09	3654,02	243,94	1,02	855,16	7,13	1,95
R6b	1289,31	4318,33	149,02	0,52	1245,66	8,65	1,94
R7b	375,67	1095,67	0	0	345,1	9,59	1,05
R9b	1041,09	3315,28	114,19	0,53	1085,18	10,05	2,26
R10b	1414,65	4332,69	38,04	0,13	1427,08	9,91	2,77
TOTAL	10883,10	35765,49	2082,41	14,52	11302,23	124,01	22,64

Comparando os resultados obtidos com os apresentados por Cordeau e Laporte (2003b), pode-se notar que a abordagem proposta neste trabalho apresentou um aumento de 63,53% na distância total percorrida pelos veículos, porém no tempo de duração das rotas, houve um aumento de apenas 0,64%. Já no tempo médio de espera dos veículos, esta abordagem foi capaz de obter uma redução de 66,17%, no tempo médio de viagem dos clientes 80,46%, e o mais interessante, 95,45% no tempo de processamento necessário para obter as soluções.

Em relação aos resultados obtidos por Bergvinsdottir et al. (2004), a abordagem proposta neste trabalho foi capaz de reduzir 11,71% no tempo de duração das rotas, 48,53% no tempo médio de espera dos veículos, 75,33% no tempo médio de viagem dos clientes, e 95,37% no tempo de processamento. Nesse caso, como as distâncias percorridas pelos veículos não são apresentadas em Bergvinsdottir et al. (2004), não é feita nenhuma comparação. Já as distâncias apresentadas em Cordeau e Laporte

(2003b), e neste trabalho, são distâncias Euclidianas entre os pontos, e conseqüentemente, não apresentam unidade de medida.

Tabela 5.3: Melhores soluções obtidas por Cordeau e Laporte (2003b).

Instância	Distância percorrida	Duração das rotas (min)	Tempo de espera (min)		Tempo de viagem (min)		CPU (min)
			total	médio	total	médio	
R1a	190,02	881,16	211,15	4,40	1094,99	45,62	1,90
R2a	302,08	1985,94	723,87	7,54	1976,73	41,18	8,06
R3a	532,08	2579,35	607,27	4,22	3586,68	49,82	17,18
R5a	636,97	3869,95	832,98	3,47	6156,48	51,30	46,24
R9a	672,44	3155,49	323,05	1,50	5621,77	52,05	50,51
R10a	878,76	4480,10	721,33	2,50	7163,71	49,75	87,53
R1b	164,46	965,06	320,60	6,68	1041,50	43,40	1,93
R2b	296,06	1564,74	308,68	3,22	2393,18	49,86	8,29
R5b	589,74	3595,63	605,89	2,52	6104,72	50,87	54,33
R6b	743,60	4072,47	448,88	1,56	7347,39	51,02	73,70
R7b	248,21	1097,25	129,03	1,79	1761,99	48,94	4,23
R9b	601,96	3249,29	487,33	2,26	5581,02	51,68	51,28
R10b	798,63	4040,99	362,37	1,26	7072,29	49,11	92,41
TOTAL	6655,01	35537,42	6082,43	42,92	56902,45	634,60	497,59

Tabela 5.4: Melhores soluções obtidas por Bergvinsdottir et al. (2004).

Instância	Distância percorrida	Duração das rotas (min)	Tempo de espera (min)		Tempo de viagem (min)		CPU (min)
			total	médio	total	médio	
R1a	-	1039	260	5,42	310	12,90	5,57
R2a	-	1994	514	5,36	1330	27,72	11,43
R3a	-	2781	301	2,09	2894	40,20	21,58
R5a	-	4274	527	2,20	4837	40,30	58,23
R9a	-	3526	32	0,15	6719	62,21	40,78
R10a	-	5025	246	0,86	8341	57,92	65,98
R1b	-	928	164	3,42	549	22,89	5,46
R2b	-	1710	162	1,69	1300	27,07	11,72
R5b	-	4336	568	2,37	4720	39,33	58,93
R6b	-	5227	513	1,78	6397	44,42	81,23
R7b	-	1316	128	1,78	784	21,76	8,29
R9b	-	3676	177	0,82	5358	49,61	44,66
R10b	-	4678	85	0,29	8119	56,38	66,41
TOTAL	-	40508	3678	28,21	51657	502,72	488,61

5.3 Comportamento do *Simulated Annealing*

O comportamento do *Simulated Annealing* foi analisado através de uma execução com a instância R10b (instância mais “difícil” de ser resolvida). Para isso, foram analisados os valores da função objetivo $f(S)$ obtidos toda vez em que uma solução melhor foi encontrada, ou seja, a cada melhora na solução.

Os gráficos apresentados nas Figuras 5.1 e 5.2 mostram o início e o final da execução do SA, respectivamente. Essa divisão em dois gráficos se deve a grande variação na escala dos valores obtidos para $f(S)$ (eixo y), o que dificultaria a visualização em apenas um gráfico.



Figura 5.1: Início da execução do SA para a instância R10b.

Nota-se claramente na Figura 5.1 que o valor da função objetivo, da solução inicial, é muito alto ($\approx 80 * 10^6$). Além disso, nota-se também que o SA utiliza boa parte de sua execução até obter uma solução válida para o problema ($\approx 34 * 10^3$).

A partir da obtenção de uma solução válida (ver Figura 5.2), o SA ainda consegue melhorar significativamente o valor de $f(S)$. Na Figura 5.2 pode-se notar que após a obtenção da primeira solução válida, algumas soluções inválidas ainda são encontradas. Esse fato é justificado através da escolha das penalizações para a função objetivo (ω e β), pois caso sejam utilizados valores muito altos para β (em relação à ω), a tendência é

que essas soluções inválidas não apareçam após a obtenção da primeira válida, pois o custo das inviabilidades resultará em soluções com valores de função objetivo também muito altos. Deve-se evitar esse tipo de comportamento, pois nesse caso, a probabilidade de aceitação de soluções piores (vizinhos piores – ver Seção 4.3) seria muito baixa, e conseqüentemente a exploração do espaço de soluções não seria suficiente para gerar soluções finais satisfatórias.

Já no caso da utilização de valores muito baixos para β , a ocorrência de soluções inválidas após a obtenção da primeira válida seria mais intensa, o que possivelmente resultaria em uma solução final inválida.

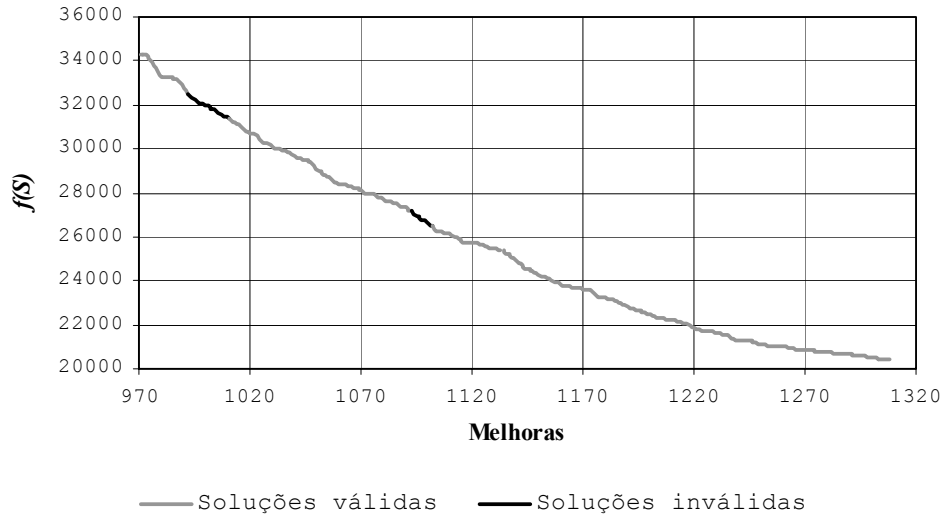


Figura 5.2: Final da execução do SA para a instância R10b.

Enfim, da forma “equilibrada” como os pesos ω e β foram utilizados, permitiu-se ao SA uma melhor exploração do espaço de soluções, e conseqüentemente a obtenção soluções válidas e de baixo custo para o problema.

5.4 Considerações finais

As referências utilizadas como comparação com a metodologia proposta foram escolhidas devido ao fato de representarem o atual estado-da-arte em que o problema se encontra.

Em relação ao trabalho de Cordeau e Laporte (2003b), a metodologia proposta apresentou um acréscimo na distância percorrida pelos veículos, apesar da duração das rotas terem sido praticamente iguais. Isso é justificado devido ao fato de que nos resultados apresentados por Cordeau e Laporte (2003b) os veículos, apesar de “andarem” menos, gastam boa parte de seu tempo “esperando”, e conseqüentemente os clientes também. Já nos resultados obtidos pelo SA, essa espera é menor, pois os veículos priorizam (através das penalizações) o atendimento aos clientes, ou seja, a qualidade do serviço prestado.

Já em relação ao trabalho de Bergvinsdottir et al. (2004), a metodologia proposta apresentou reduções expressivas em todos o quesitos analisados.

Todos os testes foram realizados em um *laptop Toshiba A10 S127* com processador *Intel Celeron®* de 2.0 GHz e 256Mb de memória RAM. Toda a implementação foi desenvolvida na linguagem C++. As soluções obtidas por Cordeau e Laporte (2003b) foram executadas em um *PC Intel Pentium 4 ®* de 2.0 GHz, e as obtidas por Bergvinsdottir et al. (2004) em um *PC Celeron®* de 2.0 GHz.

CAPÍTULO 6

CONCLUSÃO

Este trabalho apresentou uma nova abordagem para resolver o problema de roteirização e programação de veículos. O modelo proposto foi capaz de representar o problema de uma forma generalizada, sendo facilmente adaptada a outros modelos já conhecidos.

O agrupamento e o seqüenciamento das rotas foram realizados separadamente da programação em si, porém ambos através da estrutura de vizinhança e das heurísticas propostas.

O *Simulated Annealing*, integrado com as demais heurísticas apresentadas na Seção 4.3, foi capaz de obter, em todos os casos, e com pouco tempo de processamento, soluções válidas para o problema. Além disso, se mostrou robusto, pois como pode ser observado na Tabela 5.1, os desvios apresentados foram satisfatoriamente baixos. Além disso, a estrutura de vizinhança, através dos movimentos de troca, mostrou ser adequada e eficiente para exploração do espaço de soluções.

Os resultados obtidos (ver Tabelas 5.1 e 5.2) mostram que o *Simulated Annealing*, juntamente com o modelo proposto (Seção 4.2) e as demais heurísticas descritas na Seção 4.3, foram capazes de gerar soluções de boa qualidade para todas as instâncias em tempos computacionais expressivamente baixos. Esses resultados ainda foram comparados com duas outras abordagens recentes encontradas na literatura, e em todos os casos, a “qualidade do serviço” foi expressivamente maior, ou seja, a inconveniência dos clientes foi significativamente reduzida, o que na prática, em se tratando da perspectiva humana, reflete melhores soluções.

Enfim, os resultados mostram claramente o potencial da abordagem apresentada, onde soluções de alta qualidade são obtidas, para problemas relativamente grandes, em tempos de processamento expressivamente baixos.

6.1 Trabalhos futuros

A partir deste trabalho tem-se um grande campo de pesquisa a ser explorado. Uma possível extensão é a utilização da metodologia proposta neste trabalho, juntamente com Sistemas de Informações Geográficas – SIGs, para resolver problemas reais do município de São José dos Campos, por exemplo.

Outra possibilidade é a utilização de outros métodos como por exemplo *GRASP* (ver Feo e Resende (1995)) e *Algoritmo de Treinamento Populacional* (ver Oliveira (2002)) para resolver o DARP, possibilitando assim a realização de mais comparações com a metodologia proposta.

Por fim, a continuação deste trabalho será voltada para o Projeto de Pesquisa de Doutorado em Computação Aplicada proposto à FAPESP (processo 04/11053-9), cujo foco principal é a solução do DARP com a metodologia apresentada neste trabalho juntamente com outras técnicas baseadas na metodologia ATP/PL (ver Mauri (2005)), desenvolvida durante o Mestrado em Computação Aplicada (INPE).

REFERÊNCIAS BIBLIOGRÁFICAS

- Baugh jr., J. W., Kakivaya, D. K. R. e Stone, J. R. Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing. **Engineering Optimization**, 30(2): 91-124, 1998.
- Biajoli, F. L., Chaves, A. A., Mine, O. M. e Souza, M. J. F. Escala de jogos de torneios esportivos: uma abordagem via simulated annealing. **XXXV SBPO - Simpósio Brasileiro de Pesquisa Operacional** – Natal, p. 1295-1306, 2003.
- Bergvinsdottir, K. B. The genetic algorithm for solving the dial-a-ride problem. Master Thesis of Science in Engineering. Department of Informatics and Mathematical Modelling (IMM), **Technical University of Denmark (DTU)**, 2004.
- Bergvinsdottir, K. B., Larsen, J. e Jorgensen, R. M. Solving the dial-a-ride problem using genetic algorithms. IMM-Technical report-2004-202004. **Technical University of Denmark**, 2004.
- Calvo, R. W., Luigi, F., Haastrup, P. e Maniezzo, V. A distributed geographic information system for the daily car pooling problem. **Computers & Operations Research** 31: 2263-2278, 2004.
- Cordeau, J. F. A branch-and-cut algorithm for the dial-a-ride problem. Technical report 2004-23. **Center for Research on Transportation (CRT)**, Montreal, 2004.
- Cordeau, J. F. e Laporte, G. The dial-a-ride problem (DARP): Variants, modeling issues and algorithms. **4OR - Quarterly Journal of the Belgian, French and Italian Operations Research Societies**, 1: 89-101, 2003a.
- Cordeau, J. F. e Laporte, G. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. **Transportation Research B**, 37: 579-594, 2003b.
- Desrosiers, J., Dumas, Y. e Soumis, F. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. **American Journal of Mathematical and Management Sciences**, v.6, n.3&4, p.301-325, 1986.

Dumas, Y., Desrosiers, J. e Soumis, F. The pickup and delivery problem with time windows. **European Journal of Operational Research**, v.54, p.7-22, 1991.

Feo, T.A. e Resende, M. G. C. Greedy randomized adaptive search procedures, **Journal of Global Optimization**, 6:109-133, 1995.

Hart, S. M. The modeling and solution of a class of dial-a-ride problems using simulated annealing. **Control and Cybernetics Special Issue on Simulated Annealing Applied to Combinatorial Optimization**, 25(1): 131-157, Polish Academy of Sciences Systems Research Institute, 1996.

ILOG. **ILOG CPLEX 7.5: user's manual**. France, 2005. 426 p.

Ioachim, I., Desrosiers, J., Dumas, Y., Solomon, M. M. e Villeneuve, D. A request clustering algorithm for door-to-door handicapped transportation. **Transportation Science**, v.29, n.1, p.63-78, 1995.

Jaw, J., Odoni, A. R., Psaraftis, H. N. e Wilson, N. H. M. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. **Transportation Research B** 20: 243-257, 1986.

Jih, W. R., Kao, C. Y. e Hsu, F. Y. J. Using family competition genetic algorithm in pickup and delivery problem with time window constraints. **In Proceedings of the 2002 IEEE International Symposium on Intelligent Control**, Vancouver, Canada, 496 - 501, 2002.

Kirkpatrick, S., Gellat, D. C. e Vecchi, M. P. Optimization by simulated annealing. **Science**, 220: 671-680, 1983.

Madsen, O. B. G., Ravn, H. F. e Rygaard, J. M. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. **Annals of Operations Research**, 60: 193-208, 1995.

Mauri, G. R. Resolução do problema de programação de tripulações de um sistema de transporte público via simulated annealing. Relatório técnico 02/2003 - Departamento de Ciência da Computação - **Universidade Federal de Ouro Preto**, 2003.

Mauri, G. R. Novas heurísticas para o problema de escalonamento de tripulações. Dissertação de Mestrado do Curso de Pós-graduação em Computação Aplicada no **Instituto Nacional de Pesquisas Espaciais – INPE**, 2005.

Oliveira, A. C. M. Treinamento populacional em heurísticas: aplicações em otimização. Proposta de Tese (Doutorado em Computação Aplicada) – **Instituto Nacional de Pesquisas Espaciais (INPE)**, São José dos Campos. 2002.

Psaraftis, H. N. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. **Transportation Science**, v.14, n.2, p.130-154, 1980.

Psaraftis, H. N. An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. **Transportation Science**, v.17, n.3,p.351-357, 1983.

Savelsbergh, M. W. P. The vehicle routing problem with time windows: Minimizing route duration. **ORSA Journal on Computing**, 4:146-154, 1992.

Toth, P. e Vigo, D. Fast local search algorithms for the handicapped persons transportation problem. In: Osman, I. H. e Kelly, J. P. (eds). **Meta-heuristics: Theory and applications**. Kluwer, Boston, 677- 690, 1996.

Toth, P. e Vigo, D. Heuristic algorithms for the handicapped persons transportation problem. **Transportation Science**, 31: 60-71, 1997.

Xu, H., Chen, Z. L., Rajagopal, S. e Arunapuram, S. Solving a practical pickup and delivery problem. **Transportation Science**, 37(3): 347-364, 2003.

Znamensky, A. e Cunha, C. B. Um modelo para o problema de roteirização e programação do transporte de deficientes. **XIII Congresso de Pesquisa e Ensino em Transportes**, São Carlos, 1998.