

SIMULATED ANNEALING APLICADO A UM PROBLEMA DE ROTEIRIZAÇÃO E PROGRAMAÇÃO DE VEÍCULOS

Geraldo Regis Mauri

Laboratório Associado de Computação e Matemática Aplicada
Instituto Nacional de Pesquisas Espaciais
12.227-010, São José dos Campos – SP
mauri@lac.inpe.br

Luiz Antonio Nogueira Lorena

Laboratório Associado de Computação e Matemática Aplicada
Instituto Nacional de Pesquisas Espaciais
12.227-010, São José dos Campos – SP
lorena@lac.inpe.br

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 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.

PALAVRAS CHAVE. Roteirização e Programação de Veículos. *Simulated Annealing*. Modelo multi-objetivo. LGT - Logística e Transportes

ABSTRACT

This paper presents a general multi-objective mathematical model for dial-a-ride problem and an application of Simulated Annealing to solve it. The model deals with a static mode of problem and comprehend several distinct cases of the regular problem such as heterogeneous or homogeneous fleet of vehicles, multi or single depot and a multi-objective minimizing function that treats transportation costs and customer inconveniences over weighting against each other. The Simulated Annealing application is simple, but for the new neighbors' generation, three types of moves are randomly used through a uniform distribution, and the routes are clustered and scheduled in a separate way for other heuristic methods. Computational results are performed over publicly available data sets and the results are compared against current state-of-the-art methods.

KEYWORDS. Dial-a-ride problem. Simulated Annealing. Multi-objective model. LGT - Logistics and Transportation

1. Introdução

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, desde que é tido a ver com o 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).

Esse problema é conhecido por ser *NP-difícil*, 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”.

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.

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, e a metaheurística *Simulated Annealing* é utilizada juntamente com outras heurísticas para tratá-lo, 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 artigo está organizado como segue. A Seção 2 apresenta uma breve descrição do problema. O modelo é detalhado na Seção 3, enquanto a Seção 4 descreve os métodos utilizados para resolvê-lo. Os resultados computacionais obtidos são apresentados na Seção 5, e as conclusões são resumidas na Seção 6.

2. Descrição do Problema

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).

O Problema de Roteirização e Programação de Veículos ou *Dial-a-Ride Problem* – *DARP* 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.

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.

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.

3. Modelagem proposta

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 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 et al. (2004), Cordeau e Laporte (2003a), Cordeau e Laporte (2003b), etc...), e é bem próxima às situações reais do problema.

Nesse modelo, 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. E por fim, dados os locais referentes às garagens e aos pontos pertencentes às requisições, têm-se as distâncias $d_{i,j}$ e as durações das viagens $t_{i,j}$ 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 $[e_i, l_i]$.

De posse dessas informações, 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 $x^k_{i,j} = 1$ caso o veículo k vá do local i para o local j e $x^k_{i,j} = 0$ caso contrário, tem-se o seguinte modelo matemático:

Minimizar:

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

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

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

Sujeito à:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 saíra 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 são binárias, enquanto as restrições (16) e (17) garantem que as demais variáveis sejam irrestritas.

4. Simulated Annealing aplicado ao DARP

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. 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.

4.1. 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 sequência de atendimento destes. Essa heurística é apresentada na Figura 1.

1. CRIAR (m rotas vazias, e atribuí-las aos m veículos);
2. CRIAR (uma lista L com todas as requisições de transporte dos clientes);
3. PARA (cada rota k, k = 1,2,...,m) FAÇA
4. SELECIONAR ($\lfloor n/m \rfloor$ requisições de L);
5. PARA (cada requisição selecionada) FAÇA
6. Pos1 \leftarrow posição qualquer da rota k;
7. Pos2 \leftarrow posição qualquer da rota k, porém posterior a Pos1;
8. INSERIR (o ponto de embarque em Pos1);
9. INSERIR (o ponto de desembarque em Pos2);
10. FIM-PARA;
11. INSERIR (a garagem de origem do veículo k no início da rota k);
12. INSERIR (a garagem de destino do veículo k no fim da rota k);
13. REMOVER (as requisições selecionadas de L);
14. FIM-PARA;

Figura 1: Heurística de distribuição.

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.

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, o local de embarque sempre deverá ser anterior ao 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 veículos, 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 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. $B_0 \leftarrow e_0 + \min \left\{ F_0, \sum_{0 < p \leq z} W_p \right\}; D_0 \leftarrow B_0;$
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 $v_i \in P$) FACA
8. CALCULAR (F_i);
9. $B_i \leftarrow B_i + \min \left\{ F_i, \sum_{i < p \leq z} W_p \right\};$
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 2: Heurística de programação.

Considerando então uma rota $V_k = \{v_0, v_b, v_j, v_{n+j}, v_{n+i}, \dots, v_z\}$, 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, sempre lembrando 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 2.

$$F_i = \min_{i \leq j \leq z} \left\{ \sum_{i < p \leq j} W_p + \left(\min \{ 1_j - B_j, \tilde{R}_i - R_j \} \right)^+ \right\} \quad \text{se } i \in P \quad (19)$$

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

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

4.2. 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 frequentemente nos trabalhos referentes ao DARP (ver Bergvinsdottir et al. (2004), Cordeau e Laporte (2003b), Savelsbergh (1992) e Hart (1996)).

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.

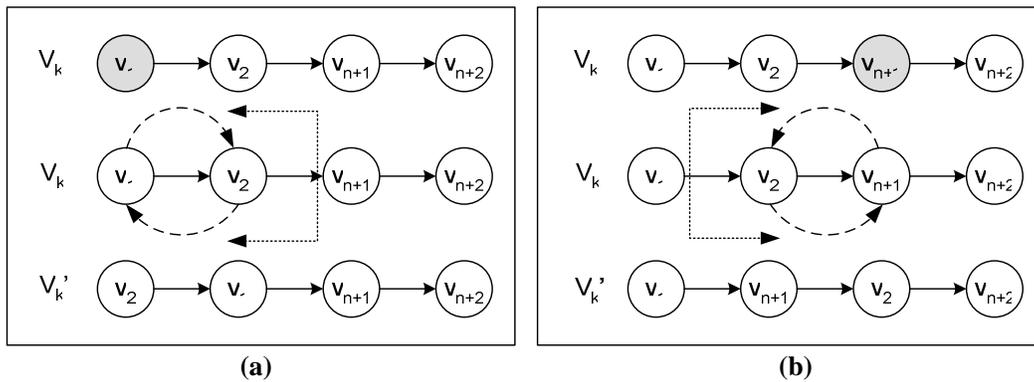


Figura 3: Movimento *Re-ordenar rota*.

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 3.

Como é selecionado apenas um ponto na rota, este pode ser de embarque ou desembarque. No primeiro caso (Figura 3a), a nova posição deverá, obrigatoriamente, ser **anterior** ao seu respectivo ponto de desembarque. Já no segundo caso (Figura 3b), a nova posição deverá ser **posterior** ao respectivo ponto de embarque. Esses “limites” são apresentados através das linhas pontilhadas na Figura 3.

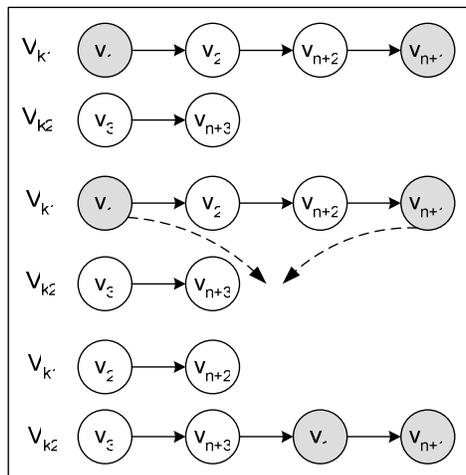


Figura 4: Movimento *Re-alocar pontos*.

Já 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. Esse movimento é ilustrado na Figura 4.

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).

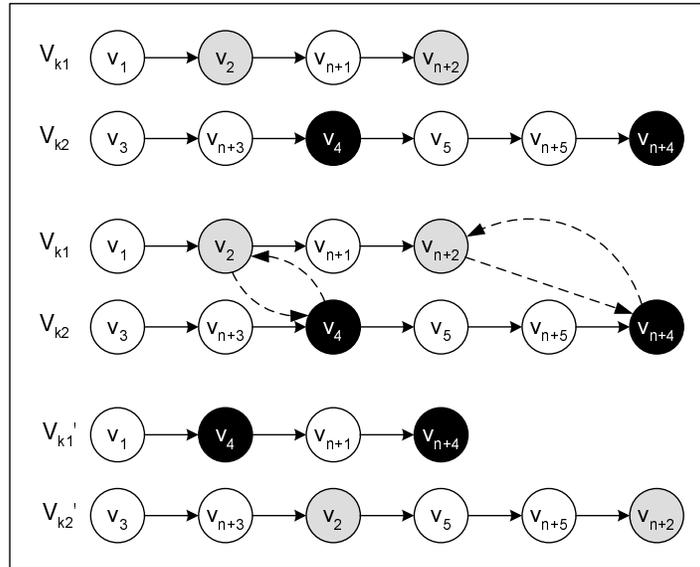


Figura 5: Movimento *Trocar pontos*.

1. DADO (α , S_{\max} , T_0 e T_c) FAÇA
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 ($T > T_c$) FAÇA
8. ENQUANTO ($IterT < S_{\max}$) FAÇA
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 6: Algoritmo *Simulated Annealing* implementado.

O movimento *Trocar pontos* consiste basicamente em selecionar duas rotas quaisquer pertencentes à solução, selecionar uma “requisição” (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 5. Nesse caso, como são trocadas requisições, 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.

A partir dessa estrutura de vizinhança, 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.

A função objetivo $f(S)$ utilizada para avaliar as soluções é aquela descrita pelas equações (1), (2) e (3) (ver Seção 3), 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 6.

5. Resultados Computacionais

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/>>), que são referências em inúmeros trabalhos de considerável importância para resolução do DARP (ver 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 3) 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).

Tabela 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 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 $\beta = [1500,1500,1500,1500,1500]$.

Para validação do modelo proposto, e da aplicação do SA ao DARP, foram realizados 5 testes para cada instância. A Tabela 1 apresenta um resumo dos resultados obtidos nesses testes ($f(S)$ 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 esses testes, 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.

Os melhores resultados obtidos (Tabela 2) ainda são comparados aos obtidos por Cordeau e Laporte (2003b), através da metaheurística Busca Tabu (Tabela 3), e por Bergvinsdottir et al. (2004), através de um Algoritmo Genético (Tabela 4). Essas referências foram escolhidas devido ao fato de representarem o atual estado-da-arte em que o problema se encontra. 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 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.

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.

Tabela 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 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

6. Conclusões

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.

Os resultados obtidos (ver Tabelas 1 e 2) mostram que o *Simulated Annealing*, juntamente com o modelo proposto (Seção 3) e as demais heurísticas descritas na Seção 4, 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.

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.

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.

Agradecimentos

Os autores agradecem à Fundação de Amparo à Pesquisa do Estado de São Paulo – FAPESP (processo 04/11053-9) e ao Conselho Nacional de Pesquisas – CNPq (processo 304598/2003-8) pelo apoio financeiro dado ao desenvolvimento deste trabalho.

Referências

- Bergvinsdottir, K. B., Larsen, J. e Jorgensen, R. M.** (2004), Solving the dial-a-ride problem using genetic algorithms. IMM-Technical report-2004-202004. Technical University of Denmark.
- Cordeau, J. F.** (2004), A branch-and-cut algorithm for the dial-a-ride problem. Technical report 2004-23. Center for Research on Transportation (CRT), Montreal.
- Cordeau, J. F. e Laporte, G.** (2003a), 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.
- Cordeau, J. F. e Laporte, G.** (2003b), A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research B, 37: 579-594.
- Hart, S. M.** (1996), 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.
- Kirkpatrick, S., Gellat, D. C. e Vecchi, M. P.** (1983), Optimization by simulated annealing. Science, 220: 671-680.
- Savelsbergh, M. W. P.** (1992), The vehicle routing problem with time windows: Minimizing route duration. ORSA Journal on Computing, 4:146-154.
- Xu, H., Chen, Z. L., Rajagopal, S. e Arunapuram, S.** (2003), Solving a practical pickup and delivery problem. Transportation Science, 37(3): 347-364.
- Znamensky, A. e Cunha, C. B.** (1998), Um modelo para o problema de roteirização e programação do transporte de deficientes. XIII Congresso de Pesquisa e Ensino em Transportes.