

PROGRAMAÇÃO DE TRIPULAÇÕES DE ÔNIBUS URBANO: UMA ABORDAGEM HEURÍSTICA

Marcone Jamilson Freitas Souza

Departamento de Computação
Universidade Federal de Ouro Preto
35.400-000 Ouro Preto, MG
marcone@iceb.ufop.br

Leonardo Xavier Teixeira Cardoso

Departamento de Computação
Universidade Federal de Ouro Preto
35.400-000 Ouro Preto, MG
leoxt@iceb.ufop.br

Gustavo Peixoto Silva

Departamento de Engenharia de Produção
Universidade Federal de Ouro Preto
35.400-000 Ouro Preto, MG
gpsilva@em.ufop.br

Resumo

Este trabalho aborda o Problema de Programação de Tripulações (PPT) no Sistema de Transporte Público. Tal problema consiste em atribuir diferentes tarefas aos tripulantes de forma que todas as viagens das linhas de uma empresa sejam executadas com o menor custo possível. A solução do PPT é um conjunto de jornadas diárias de trabalho de tripulantes, denominada escala. Neste trabalho, o PPT foi abordado utilizando as metaheurísticas *Simulated Annealing*, Busca Tabu e uma hibridização dessas duas técnicas. Cada escala gerada pelos métodos é avaliada por uma função baseada em penalidades que visa atender a legislação trabalhista, as regras operacionais da empresa, assim como melhorar o aproveitamento da mão-de-obra operacional. Os algoritmos foram testados com dados reais de uma empresa que opera na cidade de Belo Horizonte.

Palavras-chave: Programação de tripulações de ônibus, Metaheurísticas, Otimização Combinatória.

Abstract

This work deals with the Bus Crew Scheduling Problem (BCSP) related to a company operating in a public mass transit. Such problem consists in assigning the set of all vehicle trips of a given company to a set of drivers with minimal cost. The solution of BCSP is a set of driver duties named schedule. In this work the BCSP was solved through the Simulated Annealing and Tabu Search metaheuristics and a hybrid method of both. Each schedule is evaluated by a function based on penalties that has as goal to satisfy the labor agreement rules, the operational rules of the company, as well as to use adequately the crew work. The algorithms were tested with real data provided by a company operating in Belo Horizonte city.

Keywords: Bus crew scheduling, Metaheuristics, Combinatorial Optimization.

1. Introdução

O Problema de Programação de Tripulações de uma empresa do Sistema de Transporte Público (PPT) consiste em gerar uma escala de trabalho, isto é, um conjunto de jornadas de trabalho para as tripulações que conduzirão a frota em operação. Tais jornadas devem contemplar todas as viagens sob responsabilidade da empresa, satisfazer a um conjunto de leis trabalhistas e regras operacionais da empresa, com o menor custo possível. Este problema tem como dados de entrada a programação dos veículos. Portanto, considera-se que já estejam definidos os blocos de viagens de cada veículo, isto é, a sequência das viagens de cada ônibus.

A solução deste problema é uma tarefa complexa por envolver um grande número de possibilidades de solução. As escalas geradas manualmente têm como principal objetivo satisfazer as leis e regras operacionais, sem, no entanto, procurar reduzir os custos envolvidos. Esta é a situação em que se encontra a maioria das empresas nacionais, o que ressalta a importância de desenvolver técnicas de otimização voltadas para a resolução deste problema.

Na literatura, o PPT é classificado como um problema NP-difícil, o que torna inviável a solução de problemas de grande porte por métodos exatos. Dessa forma, o PPT é normalmente abordado através de técnicas heurísticas. Dentre essas técnicas, destacam-se as metaheurísticas, as quais, ao contrário das heurísticas convencionais, são providas de mecanismos para escapar de ótimos locais.

Este trabalho aborda o PPT através de modelos heurísticos baseados nas técnicas *Simulated Annealing* (SA) e Busca Tabu (BT) em suas versões puras, bem como em uma versão híbrida destas duas técnicas.

2. Revisão de Literatura

O Problema de Programação da Tripulação tem sido largamente estudado e aplicado nos países mais desenvolvidos. Uma abordagem de programação matemática clássica para o PPT é aquela que o formula como um problema de recobrimento ou de particionamento e utiliza a técnica de geração de colunas para resolvê-lo (SMITH & WREN 1988, DESROCHERS & SOUMIS 1989). A variedade de trabalhos que utiliza tais abordagens deriva das diferentes possibilidades de encontrar uma solução inteira a partir da solução do problema linear. As técnicas mais exploradas são as de *branch-and-bound*, *branch-and-price* e a relaxação lagrangeana, que permitem formas alternativas de implementação, tendo em vista as características do problema (WREN & ROUSSEAU 1995).

Os modelos de fluxo em rede, embora menos eficazes na representação do PPT, também têm sido explorados para resolvê-lo. Carraresi & Gallo (1984) descrevem formulações para o problema que se baseiam no modelo de emparelhamento. Este modelo pode ser usado iterativamente, resolvendo-se o problema diário, semanal ou mensal (SIQUEIRA 1999). Ball & Benoit-Thompson (1988) abordam o PPT encontrando um conjunto de caminhos que representam as jornadas de trabalho em um grafo. Neste grafo, os nós são as oportunidades de troca e os arcos são de dois tipos: pedaços de jornadas e conexões factíveis entre tarefas. Esta abordagem adota a relaxação lagrangeana como forma de incluir restrições adicionais ao modelo de fluxo em rede.

Os sistemas heurísticos foram os primeiros a serem utilizados na resolução do PPT (ELIAS 1964). Os primeiros sistemas desenvolvidos consistiam, no entanto, apenas na automação do trabalho antes realizado manualmente. A principal desvantagem de tais sistemas é a sua incapacidade de detectar possibilidades de otimização. Manington & Wren (1975) iniciaram a inclusão de procedimentos de otimização neste tipo de sistema, desenvolvendo heurísticas de otimização. Com o surgimento das metaheurísticas, tais como Algoritmos Genéticos (GOLDBERG, 1989), Busca Tabu (GLOVER 1989) e *Simulated Annealing* (KIRKPATRICK ET AL, 1983), abriu-se um novo horizonte na resolução do PPT. Embora tais métodos não garantam a obtenção do ótimo global, eles permitem incluir com facilidade qualquer tipo de restrição.

Várias experiências com aplicação de metaheurísticas ao PPT têm sido relatadas na literatura. Dentre esses trabalhos citamos: Wren & Wren (1995), Clement & Wren (1995), Kwan et al. (1999), Shen & Kwan (2000, 2001).

Embora o PPT tenha sido largamente estudado e aplicado nos países mais desenvolvidos, suas técnicas de resolução são ainda pouco difundidas e raramente aplicadas à nossa realidade. Isso se deve, em parte, pelo estágio primário em que se encontram as empresas do setor de transporte público no Brasil. Por outro lado, as principais restrições presentes neste problema estão relacionadas ao cumprimento das legislações trabalhistas e às normas operacionais vigentes nas empresas que atuam no sistema, o que impede que um modelo desenvolvido em um outro país seja aplicado no Brasil.

Em Silva et al. (2002a, 2002b) aplica-se *Simulated Annealing* para resolver o PPT considerando as características da realidade brasileira. Os resultados alcançados nesse trabalho demonstraram a potencialidade dessa metaheurística na resolução de problemas dessa natureza. Dando continuidade a essa iniciativa, o presente trabalho relata uma experiência com a utilização do método Busca Tabu e de uma hibridização *Simulated Annealing* – Busca Tabu.

3. Descrição do problema de programação abordado

O problema considerado para análise é o de uma empresa de transporte público, situada em Belo Horizonte, responsável por 11 linhas de ônibus, com frota empenhada de 111 veículos. Para operar esta frota a empresa conta com 219 tripulações.

No transporte público, usualmente a programação da tripulação é feita após a programação dos veículos. Nesta, as viagens são reunidas em *blocos*. Um *bloco* apresenta a seqüência de viagens que um determinado veículo tem que realizar em um dia, começando e terminando na garagem. Cada bloco mostra também as Oportunidades de Troca (OT). Uma OT é um intervalo de tempo suficiente, em um ponto apropriado, para haver a troca das tripulações. A partir do bloco de um veículo são criadas as tarefas. Cada tarefa é um conjunto de viagens compreendidas entre duas OT's. Assim, durante sua realização não é possível que haja troca de tripulação.

A programação de uma tripulação é formada por um conjunto de tarefas, chamado de *jornada*. As jornadas são divididas em dois tipos: Pegada Simples ou Dupla Pegada. No primeiro tipo as tarefas são realizadas de uma única vez e os intervalos de tempo entre as tarefas são inferiores a 2 horas. Caso ocorra um intervalo maior que duas horas, a jornada é classificada como de dupla pegada. Este intervalo não é contabilizado na remuneração da tripulação.

Ao se reunir as tarefas formando as jornadas, deve-se levar em conta inúmeras restrições operacionais e trabalhistas. Essas restrições podem ser classificadas em dois tipos: restrições essenciais (são aquelas que devem ser satisfeitas para gerar uma escala viável) e restrições não essenciais (são aquelas cujo atendimento melhoram a qualidade da escala mas que, se não satisfeitas, não geram escalas inviáveis).

Foram consideradas as seguintes restrições essenciais:

- (a) Uma tripulação não pode realizar mais de uma tarefa ao mesmo tempo;
- (b) As trocas das tripulações só podem ocorrer nas OT's;
- (c) As trocas das tripulações só podem ocorrer entre grupos de linhas predeterminadas, ou seja, grupos de linhas com as mesmas características;
- (d) Uma tripulação tem direito a 30 minutos de descanso/alimentação durante sua jornada diária de trabalho, podendo este período ser fracionado em intervalos menores, desde que um deles seja maior ou igual a 15 minutos. Numa jornada com dupla pegada, esses 30 minutos são desconsiderados;
- (e) A jornada máxima de trabalho diário é de 7:10 horas para as tripulações com pegada simples e 6:40 horas para aquelas com dupla pegada, acrescidas de até duas horas extras;
- (f) O tempo entre o final de uma jornada diária de trabalho e o seu início no dia seguinte deve ser de, no mínimo, 11 horas.

As restrições não essenciais consideradas foram as seguintes:

- (g) O tempo ocioso de uma tripulação deve ser o menor possível;
- (h) O número de tripulações com dupla pegada deve estar limitado a um certo valor;
- (i) O número de horas extras deve ser minimizado;

- (j) O número de tripulações deve ser mínimo;
- (k) O número de vezes que uma tripulação troca de veículo deve ser reduzido;
- (l) O número de vezes que uma tripulação com dupla pegada finaliza a primeira parte da jornada em um ponto e inicia a segunda parte em um outro ponto deve ser reduzido;

Essas restrições constam na Convenção Coletiva de Trabalho 2002/2003 celebrada entre o Sindicato das Empresas de Transporte de Passageiros de Belo Horizonte (SETRABH) e o Sindicato dos Trabalhadores em Transporte Rodoviário de Belo Horizonte (STTRBH).

4. Modelagem do problema

4.1 Representação

Uma solução do problema, isto é, uma escala, é representada por um vetor de tripulantes onde cada tripulante deste vetor possui uma lista de tarefas representando sua jornada de trabalho diária. Um exemplo de representação é dado pela Figura 1. Essa figura mostra, por exemplo, que a jornada de trabalho diária do tripulante 1 é constituída pelas tarefas 1, 3 e 7 respectivamente.

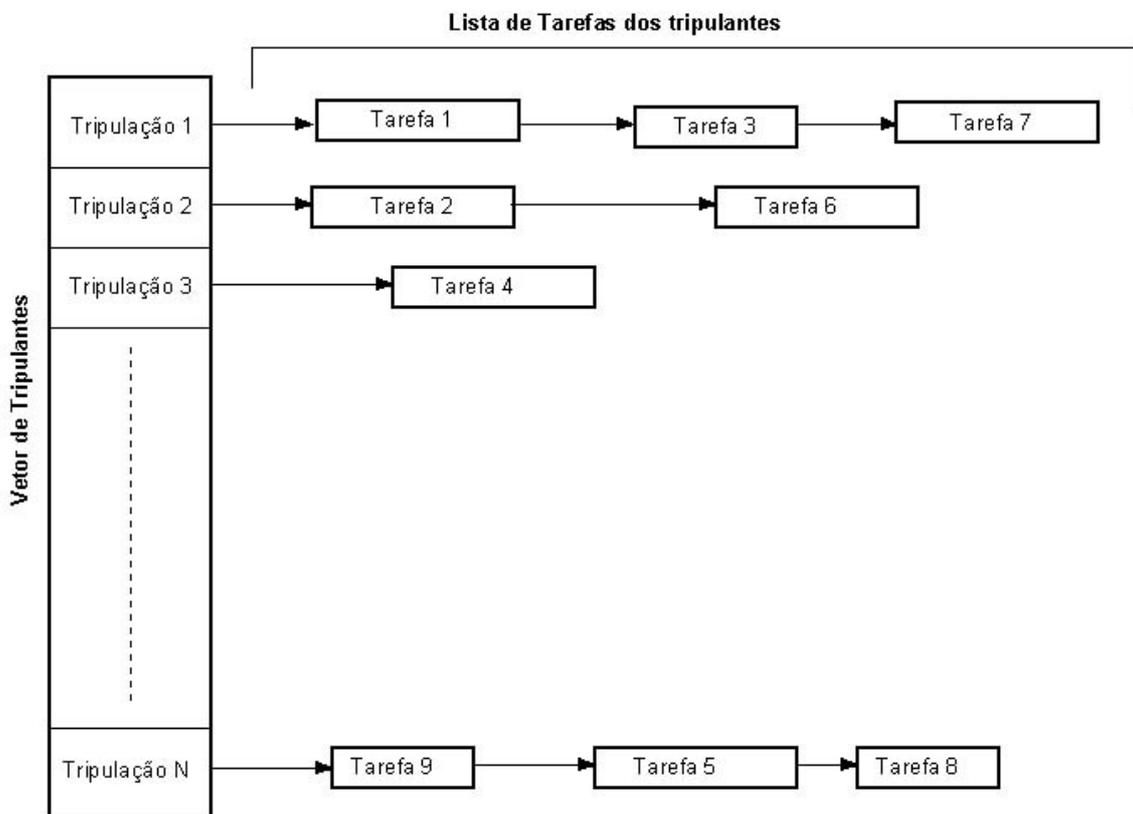


Figura 1: Representação de uma escala

4.2 Estrutura de vizinhança

Dada uma solução s , para atingir uma solução s' , onde s' é dito vizinho de s , é usado somente um tipo de movimento, a saber, realocação, para definir a estrutura de vizinhança $N(s)$.

Este movimento consiste em realocar uma tarefa de um determinado tripulante i a um outro tripulante j . Para a realização deste movimento é necessário que o tripulante i tenha pelo menos uma tarefa.

A Figura 2 ilustra o movimento de realocação. Nesta figura temos em (a) duas tripulações antes do movimento, formando uma solução s . Em (b) ocorre o movimento, onde se mostra que a tarefa escolhida passa da tripulação i para a j e que, portanto, as ligações anteriores entre as tarefas deixam de existir (linhas tracejadas da figura). Em (c) mostra-se a solução s'

vizinha de s .

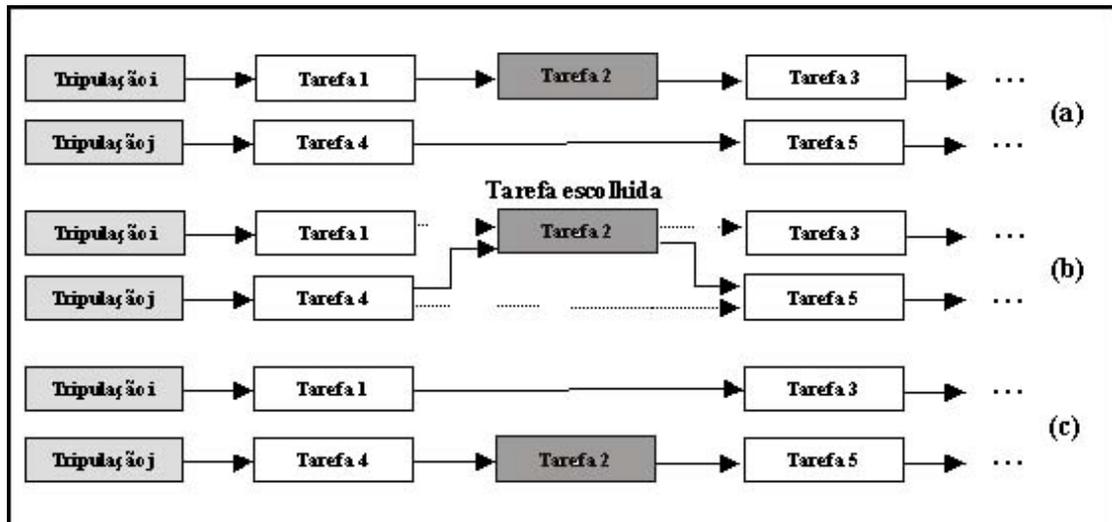


Figura 2: Ilustração do movimento de realocação

4.3 Função de avaliação

Uma solução (ou escala) s é avaliada por uma função $f : s \in S \rightarrow \mathcal{R}$, onde S é o conjunto de todas as possíveis soluções. A função de avaliação adotada baseia-se na penalização de cada um dos requisitos essenciais e não essenciais (vide seção 3) não atendidos.

Desse modo, uma escala s é avaliada com base em duas componentes, uma de inviabilidade ($g(s)$), a qual mede o não atendimento aos requisitos essenciais, e outra de qualidade ($h(s)$), a qual mede o não atendimento aos requisitos considerados não essenciais. A soma dessas duas componentes constitui a função de avaliação de uma solução s , que deve ser minimizada, isto é:

$$f(s) = g(s) + h(s)$$

Deve ser observado que uma solução s é viável se, e somente se, $g(s) = 0$. Nas componentes da função f , os pesos atribuídos às diversas medidas refletem a importância relativa de cada uma delas e, sendo assim, deve-se tomar um valor bem mais elevado para os requisitos essenciais, de forma a privilegiar a eliminação das soluções inviáveis.

4.4 Geração de uma solução inicial

A solução inicial pode ser gerada por várias metodologias. Desenvolvemos duas: a primeira, seguindo a filosofia da empresa; a segunda, utilizando a fase de construção do método GRASP.

4.4.1 Solução inicial segundo a filosofia da empresa

Esta metodologia baseia-se na repartição dos blocos dos veículos. Cada bloco de um veículo contém um conjunto de tarefas a serem executadas pelo veículo no dia. Este bloco é dividido seguindo dois critérios: sua duração e o fato de o veículo retornar ou não à garagem durante sua jornada diária. Caso o veículo não retorne à garagem, a divisão do bloco é feita de acordo com a sua duração. O bloco é dividido ao meio caso cada metade não ultrapasse 9:10 horas de trabalho, que é o tempo máximo de trabalho permitido por lei, incluindo as 2:00 horas extras. Caso contrário, o bloco é dividido em três partes, sendo as duas primeiras de 7:10 horas e a última com as tarefas remanescentes. Cada parte de um bloco é alocada a um tripulante diferente, que encontra-se inicialmente desprovido de tarefas.

Caso o veículo retorne à garagem durante sua jornada, a repartição do bloco também se dá conforme a duração da jornada, porém a divisão é feita no intervalo de tempo em que a diferença entre o final da última tarefa e o início da primeira tarefa, excluindo o tempo em que o veículo ficou na garagem, for inferior a 8:40 horas. O tempo em que o veículo permanece na

garagem é excluído porque o tripulante que receber essa jornada fará dupla pegada e, portanto, não necessitará dos 30 minutos de intervalo para repouso e/ou alimentação.

Uma das vantagens de gerar a solução inicial seguindo a filosofia da empresa é que esta não contém sobreposições de tarefas nem trocas de pontos ou linhas proibidas. Entretanto, pode ocorrer excesso nas jornadas diárias de trabalho.

4.4.2 Solução inicial segundo a fase de construção GRASP

Nesta metodologia, uma solução inicial para o PPT é construída através do procedimento da fase de construção do método GRASP (Feo & Resende 1995). A fase de construção deste método é usada para se construir iterativamente, elemento por elemento, uma boa solução inicial. Em cada iteração da fase de construção, o próximo elemento candidato a ser adicionado à solução é determinado ordenando-se todos os elementos em uma lista restrita de candidatos (LRC) de acordo com os valores de uma função gulosa de avaliação, que estima o benefício de se selecionar cada elemento. A componente probabilística do método é caracterizada por randomicamente escolher um dos melhores elementos da lista, mas não necessariamente o melhor.

No PPT tem-se uma lista de tripulantes e uma lista de tarefas que devem ser alocadas a esses tripulantes. Destas duas listas, extraímos duas listas restritas de candidatos (LRC) contendo os melhores candidatos, a saber, uma LRC das tarefas (LRCTar) e uma dos tripulantes (LRCTrip). A LRC das tarefas é ordenada em ordem decrescente de acordo com a duração da tarefa (Horário de término – Horário de início), ou seja, as tarefas com maior duração estarão no início da lista. A justificativa para isto é que as tarefas com maior duração são as mais “difíceis” de serem alocadas aos tripulantes, principalmente no final do processo de construção. A LRC dos tripulantes é ordenada em ordem crescente segundo sua respectiva função objetivo. Nesta segunda lista considera-se que todos os tripulantes receberão, para fins de avaliação, a tarefa sorteada da LRCTar, isto é, a tarefa sorteada é alocada a cada um dos tripulantes e avaliada pela função objetivo do tripulante. Somente após essa avaliação é construída a LRC dos melhores tripulantes.

Neste método tem-se dois parâmetros, γ e β , os quais definem o nível de aleatoriedade e gulosidade da construção. O parâmetro γ , onde $0 \leq \gamma \leq 1$, define o tamanho da lista restrita de candidatos das tarefas, isto é: $|LRCTar| = \gamma \times |LCTar|$, sendo LCTar o conjunto de todas as tarefas ainda não alocadas. Já o parâmetro β , onde $0 \leq \beta \leq 1$, define o tamanho da lista restrita de candidatos dos tripulantes, isto é: $|LRCTrip| = \beta \times |LCTrip|$, sendo LCTrip o conjunto de todos os tripulantes. Valores de α e β próximos a zero fazem gerar soluções praticamente gulosas; enquanto que α e β próximos a 1 fazem produzir soluções praticamente aleatórias. Esses são os únicos parâmetros que devem ser ajustados na construção de uma solução inicial para o problema.

A solução inicial para o PPT é obtida usando o seguinte procedimento: Primeiramente escolhemos aleatoriamente uma tarefa a ser alocada a partir da LRCTar. A seguir, alocamos essa tarefa a um tripulante qualquer escolhido da LRCTrip. Esse procedimento é repetido até que não haja mais tarefas a serem alocadas, isto é, $LRCTar = \emptyset$.

A heurística é adaptativa porque os benefícios associados com a escolha de cada tarefa e tripulante são atualizados em cada iteração da fase de construção para refletir as mudanças oriundas da seleção da tarefa e do tripulante anterior.

4.5 Simulated Annealing aplicado à programação de tripulações

Simulated Annealing (SA) é uma técnica de busca local probabilística, proposta originalmente por Kirkpatrick et al. (1983), que admite soluções de piora para escapar de ótimos locais.

O algoritmo parte de uma solução inicial s gerada conforme seção 4.4. Na sequência, dois tripulantes são sorteados e é gerado um vizinho qualquer s' de acordo com a estrutura de vizinhança definida na seção 4.2. Se o valor da função de avaliação $f(s')$ do vizinho for menor que o da solução corrente $f(s)$ e gerar uma variação de energia $\Delta = f(s') - f(s) < 0$, este vizinho é

aceito e passa a ser a nova solução corrente. Caso contrário, ou seja, se ocorrer um movimento de piora, a solução é aceita com uma probabilidade definida por $e^{-\Delta T}$, sendo T um parâmetro do método. Este parâmetro, conhecido como temperatura, regula a probabilidade de se aceitar soluções de piora. A temperatura T é inicializada com um valor T_0 elevado. Após um número fixo de iterações SA_{max} (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 , tal que $T_n \leftarrow \alpha \times T_{n-1}$, sendo $0 < \alpha < 1$. Com esse procedimento, existe, no início uma chance maior para escapar de mínimos locais e, à medida que T aproxima-se 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 método *SA* é interrompido quando a temperatura T for menor do que um dado valor positivo T_{final} .

4.6 Busca Tabu aplicado à programação de tripulações

De forma semelhante ao método *Simulated Annealing*, Busca Tabu é um procedimento de otimização local que admite soluções de piora. Em sua forma clássica, a cada iteração procura-se um ótimo local selecionando-se o melhor vizinho s' de um subconjunto V da vizinhança $N(s)$ da solução corrente s . Em nosso estudo, visto que a análise completa da vizinhança é uma operação custosa computacionalmente, a Busca Tabu foi implementada analisando a cada iteração somente um percentual da vizinhança. A vizinhança analisada, ou seja, os tripulantes cujas jornadas são analisadas, são gerados de forma aleatória a cada iteração do método.

Nesse método, independentemente de $f(s')$ ser melhor ou pior que $f(s)$, s' será sempre a nova solução corrente. Entretanto, apenas esse mecanismo não é suficiente para escapar de ótimos locais, uma vez que pode haver retorno a uma solução previamente gerada. Para prevenir isso, o algoritmo usa o conceito de lista tabu. Esta lista é normalmente constituída por atributos dos movimentos realizados, atributos esses que sejam fáceis de detectar e que possam prevenir o retorno a soluções geradas recentemente. Movimentos que contenham esses atributos ficam proibidos de serem feitos por um certo número $|T|$ de iterações, conhecido como *tempo tabu*. Para o PPT, quando um movimento de realocação é feito, armazena-se na lista tabu a dupla $\langle TR_{fonte}, TA \rangle$, onde TR_{fonte} é a tripulação de onde foi retirada a tarefa e TA a tarefa realocada. Desta forma, por $|T|$ iterações não será permitido realocar a tarefa TA à tripulação TR_{fonte} . Como a lista tabu pode ser muito restritiva, é comum usar um critério de aspiração, isto é, um movimento mesmo que contenha um atributo tabu pode ser realizado se satisfizer a um certo critério. Para o problema abordado, consideramos o critério de aspiração por objetivo. Assim, um movimento ainda que tabu é realizado se produzir uma solução de valor menor do que o da melhor solução encontrada até então. Visto que mesmo utilizando uma lista tabu pode ocorrer ciclagem, utilizou-se uma lista tabu de tamanho dinâmico para minimizar essa possibilidade de retorno a soluções anteriormente geradas pelo método. Essa lista varia de tamanho entre $|T_{min}|$ e $|T_{max}|$ a cada $|T|$ iterações.

Detalhes adicionais desse método podem ser encontrados em Glover (1986, 1989).

4.7 Algoritmo híbrido SA+BT aplicado à programação de tripulações

O algoritmo híbrido considerado possui três fases. Na primeira, uma solução é construída seguindo a metodologia estabelecida na seção 4.4. Na fase seguinte essa solução é refinada pelo método *Simulated Annealing*. Na terceira e última fase, a solução resultante deste método é submetida a novo refinamento, desta vez pelo método de Busca Tabu.

A Figura 3 apresenta o pseudo-código de tal procedimento.

procedimento SA+BT 1. $s^0 \leftarrow \text{ConstruaSolucaoInicial}();$ 2. $s^1 \leftarrow \text{SA}(s^0);$ 3. $s^* \leftarrow \text{SA}(s^1);$ fim SA+BT;
--

Figura 3: Procedimento híbrido SA+BT

5. Resultados

Os algoritmos foram desenvolvidos na linguagem C++ usando o compilador Borland C++ Builder 6.0. Os testes foram realizados em um microcomputador com processador Pentium IV, 1.8 MHz, com 256 GB de memória, sob sistema operacional Windows XP.

Os pesos utilizados para penalizar os diversos requisitos da função de avaliação de uma solução foram os mesmos utilizados em Silva et al. (2002a). O critério de parada adotado nos testes foi o tempo total de execução, restrito a 60 minutos. No caso do algoritmo baseado em *Simulated Annealing* puro, sempre que o sistema congelava antes de se atingir 60 minutos de execução, este era submetido a um reaquecimento partindo-se da melhor solução gerada até então e de uma temperatura inicial de 50% da temperatura inicial anterior. Já o método híbrido partia da solução final do método *Simulated Annealing* puro. Dessa forma, os resultados desse método foram obtidos após 120 minutos de execução.

Os parâmetros adotados nos algoritmos foram os seguintes: tamanho mínimo da lista tabu = $|T_{\min}| = 250$; tamanho máximo da lista tabu = $|T_{\max}| = 260$; número de iterações em que a lista tabu muda de tamanho = $\text{IterT} = 150$; percentual da vizinhança analisada a cada iteração do método BT = 50%; temperatura inicial = $T_0 = 1000000$; temperatura final = $T_{\text{final}} = 0.1$; número máximo de iterações a uma dada temperatura = $\text{SAm}_{\max} = 200000$; taxa de resfriamento = $\alpha = 0.975$, taxa de aleatoriedade para definir o tamanho da lista restrita de candidatos das tarefas = $\gamma = 0.01$; taxa de aleatoriedade para definir o tamanho da lista restrita de candidatos dos tripulantes = $\beta = 0.01$.

Cada algoritmo foi executado 10 vezes, cada qual partindo de uma semente diferente de números aleatórios. A Tabela 1 apresenta, para cada algoritmo testado, o melhor valor obtido, o valor médio, bem como o desvio em relação ao melhor valor encontrado, isto é:

$$\text{desvio} = ((\text{Valor Médio} - \text{Melhor Valor}) \times 100) / (\text{Melhor Valor})$$

Nas Tabelas 1 e 2, o prefixo FE associado ao método indica que a solução inicial foi construída seguindo a filosofia da empresa (vide seção 4.4.1) e o prefixo GRASP que a solução inicial foi gerada usando a fase de construção do algoritmo GRASP (vide seção 4.4.2).

	FE+SA	FE+BT	GRASP+BT	FE+SA+BT
Melhor valor	1642800	1270980	1410400	1279420
Valor médio	1812408	1304270	1491832	1341188
Desvio	42,6%	2,6%	17,4%	5,5%

Tabela 1: Desempenho dos algoritmos

A Tabela 2 apresenta algumas das características da solução em uso na empresa estudada, bem como as características da melhor solução produzida por cada um dos algoritmos testados.

	Empresa	FE+SA	FE+BT	GRASP+BT	FE+SA+BT
Nº tripulações	219	217	218	219	219
Excesso duplas pegadas	0	19	0	5	0
Hora extra (hh:mm)	116:00	83:25	84:11	82:45	79:59
Troca de veículos	0	23	9	15	13
Troca de linhas	0	7	8	2	4
Função de avaliação	---	1.642.800	1.270.980	1.410.400	1.279.420

Tabela 2: Características das escalas geradas pelos algoritmos

Pode-se observar que as soluções obtidas com os métodos heurísticos reduziram o número de tripulações necessárias para a operação da frota (exceto nos métodos GRASP+BT e SA+BT), assim como o total de horas extras diárias pagas pela empresa. Tais reduções foram possíveis devido à flexibilização da operação, com a realização de troca de veículos e de linhas. A Tabela 2 mostra ainda que a solução produzida pelo método FE+BT apresenta um valor da função de avaliação com menor custo do que aquele produzido pelos outros métodos. A superioridade do método de Busca Tabu com solução inicial seguindo a filosofia da empresa se deve também pela sua robustez, uma vez que, de acordo com a Tabela 1, o desvio em relação à melhor solução encontrada é menor do que os dos outros métodos. O método FE+SA+BT apresenta soluções finais também de boa qualidade, mas a um custo computacional mais elevado, uma vez que este método demandou 60 minutos a mais de processamento do que os demais.

6. Conclusões

Este trabalho contribui com a apresentação de modelos heurísticos baseados em *Simulated Annealing* e Busca Tabu para a resolução do problema de programação de tripulações de ônibus urbano considerando a realidade brasileira.

O método de Busca Tabu, partindo de uma solução inicial seguindo a filosofia da empresa, mostrou ser o mais eficiente na abordagem do PPT, pois foi o mais robusto dos métodos e o que produziu a solução final de melhor qualidade, com redução no número de tripulações e horas extras, dois dos principais itens na planilha de custos com a mão-de-obra operacional, além de produzir soluções com características mais próximas daquelas adotadas pela empresa.

Agradecimentos

Os autores agradecem aos programas PIBIC/CNPq/UFOP e PIP/UFOP pelo apoio dado ao desenvolvimento deste trabalho, bem como à Borland Latin America pela cessão de uma licença de uso do software C++ Builder 6.0.

Referências

- BALL, M. & BENOIT-THOMPSON, H. (1988) *A lagrangean relaxation based heuristic for the urban transit crew scheduling problem*. In: Daduna, J. R.; Wren, A. (eds), p. 54-67.
- CARRARESI, P. & GALLO, G. (1984). Network models for vehicle and crew scheduling. *European Journal of Operational Research*, v.16, 139-151.
- CLEMENT & WREN, A. (1995) Greedy genetic algorithms, optimizing mutantis and bus driver scheduling. In: *Computer-Aided Transit Scheduling*, Daduna, J. R.; Branco, I.; Paixão, J. M. P. (eds.), Springer-Verlag, p. 213-235.
- DESROCHERS, M & SOUMIS, F. (1989) A column generation approach to the urban transit crew scheduling problem. *Transportation Science*, v.23, n.1, p.1-13.
- ELIAS, S. E. G. (1964) - *The use of digital computers in the economic scheduling for both man and machine in public transport*. Technical Report 49, Kansas State University Bulletin, Kansas, EUA.
- FEO, T. A. & RESENDE, M. G. C. (1995). Greedy randomized adaptive search procedures, *Journal of Global Optimization*, v. 6, p. 109-133.
- GLOVER, F. (1986), Future paths for integer programming and links to artificial intelligence, *Computers and Operations Research*, v. 13, p. 533-549.

- GLOVER, F. (1989) - Tabu Search: Part I. *ORSA Journal on Computing*, 1, p.190-206.
- GOLDBERG, D. E. (1989) - *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Berkeley.
- KIRKPATRICK, S.; GELLAT, D.C. & VECCHI, M.P. (1983) - Optimization by Simulated Annealing. *Science*, p. 671-680.
- KWAN, A. S. K.; KWAN, R. K.; WREN, A. (1999) Driver scheduling using genetic algorithms with embedded combinatorial traits. In: *Computer-Aided Transit Scheduling*, Wilson, N. H. M. (ed.), Springer-Verlag, Berlin, p. 81-102.
- MANINGTON, B. & WREN, A. (1975) - *A general computer method for bus crew scheduling*. Preprints of the Workshop on Automated Techniques for Scheduling of Vehicle operators for Urban Public Transportation Services, Chicago.
- SHEN, Y & KWAN, R. S. K. (2000) – *Tabu Search for Time Windowed Public Transport Driver Scheduling*. Report 2000.14, Research Report Series, School of Computer Studies, University of Leeds.
- SHEN, Y. & KWAN, R. S. K. (2001) Tabu Search for driver scheduling. In: *Computer-Aided Scheduling of Public Transport*, Voß, S; Daduna, J. (eds.), Springer-Verlag, p. 121-135.
- SIQUEIRA, P. H. (1999) *Aplicação do algoritmo do matching no problema da construção de escalas de motoristas e cobradores de ônibus*. Dissertação de mestrado, Setor de Tecnologia e de Ciências Exatas, Universidade Federal do Paraná.
- SILVA, G. P.; SOUZA, M. J. F. & ALVES, J. M. C. B. (2002a) *Resolução do Problema de Programação Diária da Tripulação de Ônibus Urbano via Simulated Annealing*. In XVI Congresso de Pesquisa e Ensino em Transportes. Panorama Nacional de Pesquisa em Transportes. Rio de Janeiro: ANPET, 2002, v. 2, p. 95-104.
- SILVA, G. P.; SOUZA, M. J. F. & ALVES, J. M. C. B. (2002b) *Simulated Annealing Approach to Solve the Daily Crew Scheduling Problem*. In Proceedings of the IV ALIO/EURO Workshop on Applied Combinatorial Optimization. Pucon: Universidad de Concepción, 2002, v.1, p. 127-129.
- SMITH, B. M. & WREN, A. (1988) A bus crew scheduling system using a set covering formulation. *Transportation Research*, v. 22A, p. 97-108.
- WREN, A. & ROUSSEAU, J. M. (1995) *Bus driver scheduling – An overview*. In: Daduna, J. R.; Branco, I.; Paixão, J. M. P. (eds.), p. 173-187.
- WREN, A.; WREN, D. O. (1995) A genetic algorithm for public transport driver scheduling. *Computer and Operations Research*, v. 22, n.1, p.101-110.