



BUSCA TABU APLICADA AO PROBLEMA DE PROGRAMAÇÃO DE TRIPULAÇÕES DE ÔNIBUS URBANO

Euler Horta Marinho, Luiz Satoru Ochi, Lúcia M. A. Drummond

Instituto de Computação
Universidade Federal Fluminense
R. Passo da Pátria, 156 -Bloco E- São Domingos
24.210-240 Niterói, RJ
{emarinho, satoru, lucia}@ic.uff.br

Marcone Jamilson Freitas Souza

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

Gustavo Peixoto Silva

Departamento de Engenharia de Produção
Universidade Federal de Ouro Preto
35400-000 Ouro Preto, MG
gustavo@depro.em.ufop.br

Resumo

Este trabalho aborda o Problema da Programação de Tripulações de ônibus urbano. O objetivo é gerar escalas de trabalho para motoristas e cobradores de uma empresa, satisfazendo a um conjunto de regras trabalhistas e regras operacionais da empresa, com o melhor aproveitamento possível da mão-de-obra. O problema é resolvido por um método de Busca Tabu, o qual parte de uma solução inicial inteligentemente construída e faz uso de duas diferentes estruturas de vizinhança e de uma estratégia de diversificação para explorar o espaço de soluções. O método foi testado com dados reais de uma empresa brasileira que opera no sistema de transporte público da cidade de Belo Horizonte. Os resultados obtidos mostram a superioridade do método, tanto em relação à qualidade da solução final, quanto em relação à sua rapidez na produção de soluções de boa qualidade, quando comparado a um outro método existente na literatura.

Palavras-chave: Programação de Tripulações de Ônibus, Busca Tabu, Metaheurística.

Abstract

This work deals with the Bus Driver Scheduling problem. The goal is the generation of schedules for drivers and conductors of a company, satisfying a set of labor agreement and operational rules of the company, as well as to optimize the workforce. The problem is solved by a Tabu Search heuristic, which starts from a initial solution intelligently constructed and uses two distinct neighborhood structures and diversification strategy for explore the search space. The method was tested with real data provided by a Brazilian company of public transport from Belo Horizonte city. The results obtained show the superiority of the method, both regarding the quality of final solution and the speed in producing high quality solutions, when compared with another method found in the literature.

Keywords: Bus Driver Scheduling, Tabu Search, Metaheuristic.

1. Introdução

O Problema de Programação de Tripulações (PPT) também conhecido como *Crew Scheduling Problem* (Wren & Rousseau, 1995) ou *Bus driver Scheduling Problem* (Lourenço, Paixão & Portugal, 2001) trata da geração de escalas de motoristas e cobradores de empresas de transporte público. Este problema tem sido objeto de estudo desde a década de 60 (Wren & Rousseau, 1995). Essa constante investigação deve-se às contínuas transformações ocorridas nos sistemas de transporte público onde as empresas deste setor buscam a redução de custos operacionais e o aprimoramento de seus processos produtivos, mantendo a qualidade dos seus produtos e/ou serviços. Segundo a Associação Nacional dos Transportes Públicos (ANTP), para que ocorra a democratização do acesso aos sistemas de transporte público, deve haver a garantia de tarifas toleráveis para os usuários, o que depende em grande parte do aumento da eficiência e redução de custos por parte das organizações do setor. Como a mão-de-obra operacional é uma das componentes que mais pesam na planilha de custos, uma pequena redução neste item pode significar um ganho considerável no custo total (Bouzada, 2002), o que justifica qualquer trabalho no sentido de minimizar os custos com a mão-de-obra. O interesse por esse problema nas últimas décadas é demonstrado pela publicação de uma série de volumes intitulados *Computer-Aided Scheduling of Public Transport* (Wren, 1981), (Rousseau, 1985), (Daduna & Wren, 1988), (Desrochers & Rousseau, 1992), (Daduna, Branco & Paixão, 1995), (Wilson, 1999) e (Daduna & Voss, 2001).

Embora o PPT tenha sido largamente estudado e aplicado em outros países (Blais, Lamont & Rousseau, 1990), (Fores, Proll & Wren, 1999) e (Lourenço, Paixão & Portugal, 2001), suas técnicas de resolução são pouco difundidas e raramente aplicadas à nossa realidade. Isso se deve, em parte, pelo estágio primário em que se encontram as empresas de transporte público no Brasil, onde as empresas desse setor normalmente geram manualmente as escalas de trabalho.

As principais restrições presentes neste problema estão relacionadas ao cumprimento das legislações trabalhistas e das normas operacionais vigentes nas empresas que atuam no sistema. Tais fatores impedem que um sistema desenvolvido no Reino Unido, por exemplo, seja aplicado no Brasil.

A dificuldade em se resolver o PPT é devido ao grande número de combinações, o que inviabiliza uma resolução por enumeração de todas as possíveis soluções. Essa intratabilidade computacional se deve ao fato de o problema ser NP-difícil, uma vez que se reduz ao problema de recobrimento de conjuntos ou ao problema de particionamento de conjuntos, que são NP-difíceis (Garey & Johnson, 1979). Muitos trabalhos presentes na literatura para a resolução desse problema apresentam abordagens por metaheurísticas, as quais podem facilmente tratar modelos com muitas restrições (Wren, Kwan & Kwan, 2000), (Shen & Kwan, 2001) e (Lourenço, Paixão & Portugal, 2001). Outros descrevem abordagens por técnicas de programação matemática (Desrochers & Soumis, 1989) e (Fores, Proll & Wren, 1999). Para uma revisão dos principais métodos de solução do PPT referencia-se Ernst et al. (2004a, 2004b).

O desenvolvimento de um método de solução eficiente para o PPT é, portanto, de grande importância. Neste trabalho aborda-se o PPT através do método de Busca Tabu. Foram desenvolvidas várias versões do método, a partir de diferentes critérios de exploração das vizinhanças. Essas diferentes versões foram comparadas com um método da literatura aplicado na resolução de um caso de uma empresa brasileira de transporte público. Todas as versões foram capazes de produzir soluções finais de qualidade melhor que aquelas propostas pela empresa e pelo referido trabalho.

Este artigo está organizado como segue. Na seção 2 descreve-se o problema abordado. A seção 3 apresenta a metodologia abordada, incluindo a forma de representação adotada e de geração de uma solução inicial, a estrutura de vizinhança, a função de avaliação e a adaptação da técnica heurística Busca Tabu ao problema. A seção 4 mostra os resultados encontrados e a seção 5 conclui o trabalho e aponta os trabalhos futuros.

2. Descrição do Problema Abordado

No transporte público, a Programação de tripulações é feita a partir da Programação de Veículos. Nesta, as viagens são reunidas em *Blocos*. Um bloco apresenta a seqüência de viagens que um determinado veículo deve realizar em um dia, começando e terminando na garagem. Cada bloco apresenta as *Oportunidades de troca (OT)*. A OT é um intervalo de tempo suficiente para que ocorra a troca de tripulações. A cada oportunidade de troca está associado um *Horário de Troca (HT)* e um

Ponto de Troca (PT), os quais representam respectivamente a hora e um local onde poderá haver uma troca de tripulação .

A partir do bloco de um veículo são criadas as *Tarefas*. Cada tarefa é um conjunto de viagens reunidas de maneira que haja apenas duas OT: uma no início e outra no final da tarefa. 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 elas são menores que duas horas. A duração normal de uma jornada do tipo *Pegada Simples* é de 7:10 horas. Caso ocorra um intervalo maior ou igual a duas horas a jornada é classificada como dupla pegada. Este intervalo não é contabilizado na remuneração da tripulação. A duração normal de uma jornada do tipo *Dupla Pegada* é de 6:40 horas.

Ao se reunir as tarefas formando as jornadas deve-se levar em conta as seguintes restrições operacionais e trabalhistas:

- a) Uma tripulação não pode realizar mais de uma tarefa ao mesmo tempo (*Sobreposição de Tarefas*);
- b) Uma tripulação não pode realizar duas tarefas consecutivas na qual o ponto de troca do final da primeira seja diferente do ponto de troca do início da segunda, e o intervalo entre elas seja menor do que duas horas (*Troca de Pontos Proibida*);
- c) Por determinações trabalhistas uma tripulação não pode exercer mais que 2 (duas) horas de atividades diárias além da duração normal de sua jornada (*Horas Excedentes*);
- d) Durante a realização de uma jornada de trabalho a tripulação tem direito a uma folga de 30 minutos destinada à refeição. Por acordo feito pelos trabalhadores, esta folga pode ser dividida e distribuída no decorrer do dia, conquanto que uma das frações seja de pelo menos 15 minutos. O método desenvolvido contempla automaticamente esse quesito, sendo que este não é contabilizado no cálculo da função de avaliação (*Folga Acumulada Deficiente*);
- e) O tempo mínimo entre jornadas de trabalho é de 11:00 horas. A jornada da tripulação que não atingir esse valor será penalizada pelo intervalo de tempo necessário para se atingir o tempo mínimo (*Tempo entre Jornadas*);
- f) Uma tripulação não pode realizar duas tarefas consecutivas na qual o grupo da linha final da primeira seja diferente do grupo da linha inicial da segunda, isto é, um veículo que faz um certo itinerário não pode fazer logo a seguir outro itinerário que não esteja em uma mesma região definida pela empresa (*Troca de Linhas Proibida*).
- g) O número de tripulações que têm jornadas do tipo dupla pegada não pode superar um certo número (*Número de duplas pegadas*);
- h) O número de horas que superam a duração normal de uma jornada é considerado hora-extra. Este montante deve ser minimizado (*Horas Extras*);
- i) O número de trocas de pontos nos intervalos que caracterizam jornadas do tipo dupla pegada deve ser minimizado. Isto é, evitar sempre que possível que a última tarefa da primeira pegada de uma tripulação ocorra em um ponto diferente da primeira tarefa da segunda pegada (*Troca de Pontos Permitida*);
- j) O número de trocas de veículos deve ser minimizado. Isto é, reduzir o número de vezes em que uma tripulação realiza duas tarefas consecutivas em veículos diferentes (*Troca de Veículos*);
- k) O número total de jornadas de trabalho deve ser minimizado (*Número de Jornadas*);
- l) Reduzir, tanto quanto possível, a ociosidade nas jornadas (*Ociosidade*);
- m) O número de vezes que uma tripulação troca de linha dentro de um mesmo grupo deve ser minimizado (*Troca de Linhas Permitida*).

As sete primeiras restrições ((a)...(g)) são consideradas essenciais, ou seja, se não forem atendidas a solução obtida é infactível. As demais restrições ((h)...(m)) são consideradas não essenciais, ou seja, devem ser atendidas sempre que possível.

3. Metodologia

Este trabalho aborda o PPT utilizando a Busca Tabu. Assim, na seção 3.1 será descrita a forma de representação da solução do problema; na seção 3.2, o procedimento de geração da solução inicial; na seção 3.3, o estabelecimento das estruturas de vizinhança; na seção 3.4, a função de avaliação e na seção 3.5, a aplicação da Busca Tabu ao PPT.

3.1. Representação

Os elementos básicos para a formação das jornadas são as tarefas (Figura 1). Essas agregam as informações: número da tarefa, que é um identificador único da mesma, o grupo a qual ela pertence, horário inicial, horário final, linha inicial, linha final, ponto inicial, ponto final, folga acumulada (isto é, o período de tempo ocioso entre as viagens agrupadas na tarefa) e número do veículo.

TAREFA	
Número	Grupo
Horário Inicial	Horário Final
Linha Inicial	Linha Final
Ponto Inicial	Ponto Final
Folga Acumulada	Número do Veículo

Figura 1: Modelagem de uma tarefa

As jornadas, conforme ilustrado na Figura 2, são constituídas por uma lista de tarefas. Para cada tarefa contabiliza-se: (a) Sobreposição, que representa a quantidade de tempo, em minutos, em que duas ou mais tarefas são realizadas simultaneamente; (b) Horas extras, que representa a quantidade de tempo, em minutos, que excedem a duração normal de uma jornada; (c) Horas excedentes, que representa a quantidade de tempo, em minutos, que excedem a duração máxima de uma jornada; (d) Ociosidade, que representa a quantidade de tempo, em minutos, relativamente à duração de uma jornada, em que não há realização de viagens; (e) Tempo entre jornadas, que representa a quantidade de tempo, em minutos, que falta para a diferença entre o horário final da jornada em um dia e o horário inicial da mesma no dia subsequente alcançar 11 (onze) horas; (f) Troca de Veículos, que representa o número de vezes em que há troca de veículos na jornada; (g) Troca de Pontos Proibida, que representa o número de vezes em que ocorreu troca de pontos proibida. As informações Troca de Pontos Permitida, Troca de Linhas Proibida e Troca de Linhas Permitida são calculadas como no item (g) e de acordo com as restrições homônimas definidas na seção 2.

JORNADA	
Número	
Sobreposição	Horas Extras
Horas Excedentes	Ociosidade
Tempo entre Jornadas	Troca de Veículos
Troca de Pontos Proibida	Troca de Pontos Permitida
Troca de Linhas Proibida	Troca de Linhas Permitida

Figura 2: Modelagem de uma jornada

Uma solução para o problema é formada por uma lista de jornadas (Figura 3).

SOLUÇÃO	
Número de Jornadas	Número de duplas pegadas

Figura 3: Representação da solução do PPT

3.2. Geração de uma solução inicial

A solução inicial é obtida a partir do particionamento do bloco de veículos nas oportunidades de troca, o que irá gerar um conjunto de tarefas. Na Figura 4 é ilustrado parte de um bloco de um veículo particionado em tarefas.

Viagem	Ponto de Partida	Horário de Partida	Ponto de Chegada	Horário de Chegada	Viagem Vazia	Tempo de Terminal	Linha	
00	0	04:50	1	05:00	10	00	101	} Tarefa 1
01	1	05:00	1	05:45	00	15	101	
02	1	06:00	1	06:45	00	06	101	➔ Tarefa 2
03	1	06:51	1	07:36	00	06	101	➔ Tarefa 3
04	1	07:42	1	08:27	00	06	101	➔ Tarefa 4
05	1	08:33	1	09:15	00	00	101	} Tarefa 5
06	1	09:15	1	10:00	00	05	101	

Figura 4: Bloco de um veículo particionado em tarefas

O procedimento começa pela primeira viagem do bloco do veículo e procede pelas demais até encontrar um intervalo entre viagens de no mínimo 5 minutos, necessário para a troca de tripulações. Então, as viagens analisadas nesse passo são agrupadas em tarefas. O procedimento é repetido até que sejam analisados todos os blocos de veículos. Ao final desse procedimento, tem-se um conjunto de tarefas. A partir desse conjunto de tarefas serão formadas as jornadas, conforme explicado a seguir.

Para minimizar o número de inviabilidades na solução inicial, as tarefas a serem agrupadas em jornadas pertencem ao mesmo veículo. Para essa finalidade é calculada a duração do bloco do veículo e se o mesmo é do tipo pegada simples ou dupla pegada. Se o bloco do veículo for do tipo pegada simples e a duração do mesmo for igual ou inferior a 9:10 horas (duração máxima da jornada de trabalho do tipo pegada simples) é formada uma única jornada. Caso a duração seja superior a 9:10 horas e inferior a 18:20 horas são formadas duas jornadas. Se a duração for superior a 18:20 forma-se três jornadas. Dessa forma o requisito *horas excedentes* não é violado. No caso de o bloco do veículo ser do tipo dupla pegada a duração máxima da jornada é de 8:40 horas e o procedimento de agrupamento é semelhante ao descrito para o caso de pegada simples.

3.3. Estrutura de Vizinhaça

Foram definidos dois tipos de movimentos para definir a vizinhaça $N(s)$ de uma dada solução s (Figura 5). O primeiro caracteriza-se pela realocação de uma tarefa de uma determinada jornada para outra jornada. O conjunto de todas as soluções s' geradas a partir de s através de movimentos de realocação define a vizinhaça $N^{(R)}(s)$. Como pode ser observado em (a) a tarefa “C” foi realocada da jornada i para a jornada j .

O segundo movimento é caracterizado pela troca de tarefas entre duas jornadas. O conjunto de todas as soluções s' geradas a partir de s através de movimentos de troca define a vizinhaça $N^{(T)}(s)$. Como pode ser observado em (b), as jornadas i e j trocaram entre si as tarefas “D” e “G”.

Uma solução s' , é dita vizinha de s se for obtida desta a partir de um movimento de realocação de tarefas ou de troca de tarefas, isto é, $s' \in N(s) = N^{(R)}(s) \cup N^{(T)}(s)$.

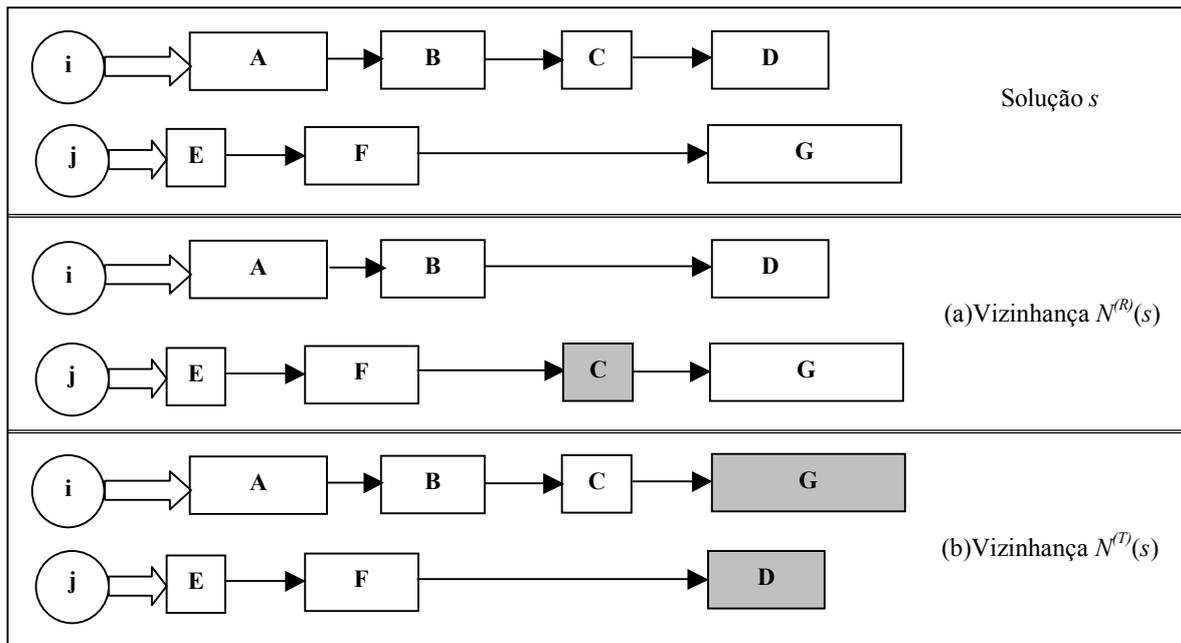


Figura 5: Estruturas de Vizinhança

3.4. Função de Avaliação

A função de avaliação considerada neste trabalho, e que deve ser minimizada, é baseada na penalização dos requisitos não atendidos, conferindo aos requisitos essenciais um peso maior que aos requisitos não essenciais. Desta forma, uma solução s é avaliada com base na seguinte função f :

$$f(s) = \sum_{i=1}^N (\alpha_1 \times \text{quantTempoOcioso}_i + \alpha_2 \times \text{quantHoraExtra}_i + \alpha_3 \times \text{numTrPPerm}_i + \alpha_4 \times \text{numTrLPerm}_i + \alpha_5 \times \text{numTrVeiculos}_i + \beta_1 \times \text{quantExcTempoTrab}_i + \beta_2 \times \text{quantSobrep}_i + \beta_3 \times \text{numTrPPr oib}_i + \beta_4 \times \text{numTrLPr oib}_i + \beta_5 \times \text{quantTempoEJornInsuf}_i) + \alpha_6 \times \text{TotalJorn} + \beta_6 \times \text{TotalDuplaPeg} + \beta_7 \times \text{TotalJornInfactiveis}$$

Onde:

- (i) N é o número de jornadas;
- (ii) α_i e β_i são as penalidades aplicadas aos requisitos não-essenciais e essenciais, respectivamente;
- (iii) $\text{quantTempoOcioso}_i$ é a quantidade total de ociosidade, em minutos, na jornada i e α_1 a penalização desse quesito;
- (iv) quantHoraExtra_i é a quantidade de horas extras, em minutos, na jornada i e α_2 a penalização correspondente;
- (v) numTrPPerm_i é o número de vezes em que ocorreu troca de pontos permitida durante a jornada i e α_3 a penalização desse quesito;
- (vi) numTrLPerm_i é o número de vezes em que ocorreu troca de linhas permitida durante a jornada i e α_4 a penalização correspondente;
- (vii) numTrVeiculos_i é o número de vezes em que ocorreu troca de veículos durante a jornada i e α_5 a penalização desse quesito;
- (viii) $\text{quantExcTempoTrab}_i$ é a quantidade de horas excedentes, em minutos, na jornada i e β_1 a penalização correspondente;

(ix) $quantSobrep_i$ é a quantidade de tempo, em minutos, em que duas ou mais tarefas são realizadas simultaneamente na jornada i e β_2 a penalização para esse quesito;

(x) $numTrPProib_i$ é o número de vezes em que ocorreu troca de pontos proibida durante a jornada i e β_3 a penalização desse quesito;

(xi) $numTrLProib_i$ é o número de vezes em que ocorreu troca de linhas proibida durante a jornada i e β_4 a penalização desse quesito;

(xii) $quantTempoEJornInsuf_i$ é a quantidade de tempo, em minutos, que falta para que a diferença entre o horário final da jornada i em um dia e o horário inicial no dia subsequente atinja o tempo mínimo entre jornadas de 11 horas. β_5 é a penalização correspondente;

(xiii) $TotalJorn$ é o número total de jornadas e α_6 a penalização correspondente;

(xiv) $TotalDuplaPeg$ é a quantidade de jornadas do tipo Dupla Pegada que supera um certo número, previamente definido pela empresa. β_6 é a penalização correspondente;

(xv) $TotalJornInfactiveis$ é o número de jornadas infactíveis e β_7 a penalização desse quesito.

3.5. Busca Tabu aplicada ao PPT

O método de Busca Tabu, proposto independentemente por Glover (1986) e Hansen (1986), é um procedimento iterativo para a solução de problemas de otimização combinatoria que aceita movimentos de piora para escapar de ótimos locais.

Começando com uma solução inicial s_0 , o método explora a cada iteração, um subconjunto V da vizinhança $N(s)$ da solução corrente s . O membro s' de V com menor valor nessa região segundo a função $f(\cdot)$ torna-se a nova solução corrente mesmo que s' seja pior que s , isto é que $f(s') > f(s)$.

O critério de escolha do melhor vizinho é utilizado para escapar de um mínimo local. Esta estratégia, entretanto, pode fazer com que o algoritmo cicle, isto é, que retorne a uma solução já gerada anteriormente.

De forma a evitar que isto ocorra, existe um mecanismo chamado memória de curto prazo ou lista tabu. O objetivo dessa lista é tentar evitar movimentos que levem à regiões já visitadas do espaço de soluções, o que usualmente é alcançado pela proibição dos últimos movimentos realizados. Esses movimentos são armazenados nesta lista e permanecem proibidos (com *status* tabu), por um dado número de iterações, chamado *tabu tenure*. Uma vez que esta estratégia pode ser muito restritiva, de forma a não desconsiderar soluções de alta qualidade, movimentos com *status* tabu podem ser aceitos se a nova solução produzida satisfaz um critério de aspiração, no caso, o critério de aspiração por objetivo, isto é, se a solução ainda que tabu seja melhor que a melhor solução gerada até então. De forma a reduzir ainda mais a probabilidade de ciclagem, foi utilizada no método proposto, uma lista tabu de tamanho dinâmico, no caso, sorteando o número de iterações em que o movimento permanecerá tabu dentro de uma faixa ($minTabuTenure - maxTabuTenure$). Na Figura 6 é apresentado o pseudocódigo do método de Busca Tabu clássico.

```

procedimento BuscaTabu( $f(\cdot)$ ,  $N(\cdot)$ ,  $|V|$ ,  $|T|$ ,  $BTmax$ ,  $s$ )
1  $s^* \leftarrow s$ ;           {Melhor solução obtida até então}
2  $Iter \leftarrow 0$ ;       {Contador do número de iterações}
3  $MelhorIter \leftarrow 0$ ; {Iteração mais recente que forneceu  $s^*$ }
4  $T \leftarrow \emptyset$ ;   {Lista Tabu}
5 enquanto (Critério de parada não satisfeito) faça
6    $Iter \leftarrow Iter + 1$ ;
7   Seja  $s' \leftarrow s \oplus m$  o melhor elemento de  $V \subset N(s)$  tal que
   o movimento  $m$  não seja tabu ( $m \notin T$ ) ou  $f(s') < f(s^*)$ ;
8   Atualize a lista tabu;
9    $s \leftarrow s'$ ;
10  se ( $f(s) < f(s^*)$ ) então
11     $s^* \leftarrow s$ ;
12     $MelhorIter \leftarrow Iter$ ;
13  fim-se;
14  fim-enquanto;
15   $s \leftarrow s^*$ ;
16  Retorne  $s$ ;
fim BuscaTabu;

```

Figura 6: Algoritmo clássico de Busca Tabu

No método desenvolvido a busca pelo melhor elemento (linha 7 do pseudocódigo) é feita somente na vizinhança $N^{(R)}(s)$ ou somente na vizinhança $N^{(D)}(s)$, conforme definido na seção 3.3. A escolha da estrutura de vizinhança a ser analisada a cada iteração é feita de acordo com uma dada probabilidade, sendo p a probabilidade de se escolher a vizinhança $N^{(R)}(s)$ e, conseqüentemente, $1-p$ a probabilidade de se escolher a vizinhança $N^{(D)}(s)$.

Foram desenvolvidas três versões do método para resolver o PPT, a saber: Busca Tabu com primeiro de melhora em PercViz% da vizinhança (BT-FF-IC), Busca Tabu com melhor vizinho em vizinhança variável (BT-BF-IC) e Busca Tabu com primeiro de melhora em PercViz% da vizinhança, com diversificação (BTR-FF-IC). Todas essas versões são justificadas devido ao fato de que a análise da vizinhança completa no PPT é de elevado custo computacional. As seções seguintes descrevem melhor essas três versões.

3.5.1 Busca Tabu com primeiro de melhora em PercViz% da vizinhança

Nesta estratégia, denominada BT-FF-IC, o melhor vizinho é procurado dentro de um percentual da vizinhança (50%), interrompendo-se a busca quando um movimento de melhora em relação à solução corrente é encontrado.

3.5.2 Busca Tabu com melhor vizinho em vizinhança variável

Nesta estratégia, denominada BT-BF-IC, procura-se o melhor vizinho começando com a exploração de um percentual reduzido da vizinhança (5%) e quando ocorre $BTMax$ iterações sem melhora, esse percentual é aumentado por um fator de 10%. É importante destacar que todo o percentual dessa vizinhança é analisado.

3.5.3 Busca Tabu com primeiro de melhora em PercViz% da vizinhança, com diversificação

Com o intuito de melhorar a robustez da variante BT-FF-IC, que nos testes realizados se mostrou a melhor, foi incorporada à mesma uma estratégia de diversificação. Esta estratégia consiste em reiniciar o procedimento de Busca Tabu sempre que não se alcance uma solução viável em um determinado número limiar de iterações, no caso, 200.

4. Resultados

O algoritmo desenvolvido foi implementado na linguagem C++ usando o compilador gcc 3.2.2 usando o *flag* -O3 e testado em um microcomputador PC Pentium 4, 2,6 GHz, com 512 MB de RAM sob o sistema operacional Linux.

Inicialmente, foram realizados testes para calibrar os parâmetros das diversas versões dos métodos implementados. Em todas elas, os parâmetros adotados foram: $BTMax = 500$ (número máximo de iterações sem melhora no valor da melhor solução), $minTabuTenure = 55$ (número mínimo de iterações na lista tabu), $maxTabuTenure = 60$ (número máximo de iterações na lista tabu) e $p = 0.8$ (probabilidade de escolha da vizinhança $N^{(R)}(s)$, que trabalha com movimentos de realocação). As penalizações para os requisitos não essenciais encontram-se na Tabela 1 e para os requisitos essenciais, na Tabela 2.

Tabela 1: Penalizações dos requisitos não essenciais

α_1	α_2	α_3	α_4	α_5	α_6
40	60	300	300	5000	1000

Tabela 2: Penalizações dos requisitos essenciais.

β_1	β_2	β_3	β_4	β_5	β_6	β_7
9000	5000	13000	13000	5000	9000	13000

A seguir, as diversas versões do método de Busca Tabu foram comparadas com o método VNS-RTL proposto por Souza et al. (2004), baseado no Método de Pesquisa em Vizinhança Variável. Para tal finalidade foram realizados 20 testes para cada método, cada qual partindo de uma semente diferente de números aleatórios, com um tempo de execução de 1 hora. Uma nova versão, BT-FF, foi desenvolvida utilizando o mesmo procedimento de construção de uma solução inicial do método VNS-RTL. Este último método foi executado em um microcomputador com a mesma configuração acima descrita, porém sob o sistema operacional Windows XP.

A Tabela 3 apresenta o melhor valor encontrado para a função de avaliação e o desvio médio em relação ao valor da melhor solução encontrada em todos os testes, isto é,

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

Tabela 3: Desempenho dos algoritmos

Método	VNS-RTL	BT-BF-IC	BT-FF-IC	BT-FF	BTR-FF-IC
Melhor valor	1.188.980	996.740	986.440	1.012.940	976.880
Desvio	26,33%	38,16%	20,04%	153,55%	3,45%

Pela Tabela 1, observa-se que o método BTR-FF-IC foi o que produziu a solução de melhor qualidade e com menor desvio, reduzindo o valor da melhor solução produzida pelo método VNS-RTL em 21,7%. Observa-se, também, que o procedimento de construção teve uma forte influência na qualidade das soluções finais encontradas, uma vez que dentre os algoritmos baseados em Busca Tabu, os que utilizaram o processo de construção descrito na seção 3.2 (BT-BF-IC, BT-FF-IC e BTR-FF-IC) tiveram melhor desempenho que o método BT-FF, o qual utiliza o processo de construção do método VNS-RTL.

A Tabela 4 apresenta algumas características da melhor solução produzida por cada método, comparando-as com a solução obtida pela empresa.

Tabela 4: Características das melhores soluções geradas pelos algoritmos

	Empresa	VNS-RTL	BT-BF-IC	BT-FF-IC	BT-FF	BTR-FF-IC
#Jornadas	219	219	219	219	219	218
Hora-extra (hh:mm)	116:00	87:13	84:39	87:49	86:49	87:18
Trocas de veículos	0	28	32	27	33	29
Trocas de linhas	0	13	24	13	23	15

Por esta tabela, observa-se que, em relação à empresa, todos os métodos heurísticos produziram soluções com redução no número de horas-extras. Adicionalmente, o método BTR-FF-IC conseguiu reduzir o número de jornadas e, conseqüentemente, de tripulações necessárias.

A Figura 7 ilustra a evolução típica da melhor solução produzida pelos métodos VNS-RTL e BTR-FF-IC nos instantes iniciais da execução dos mesmos.

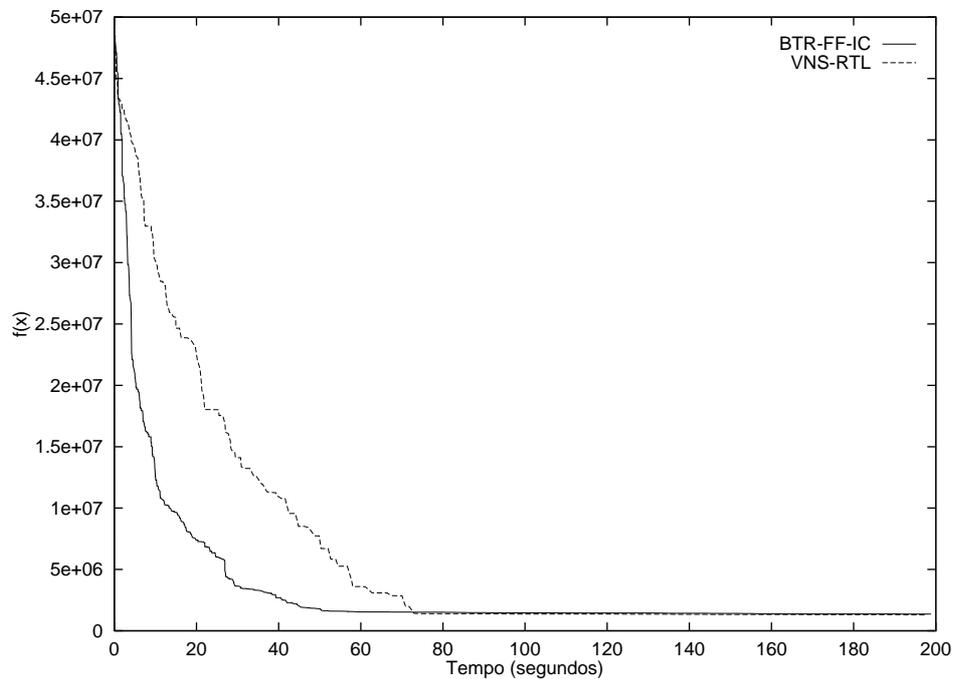


Figura 7 – Evolução dos algoritmos BTR-FF-IC e VNS-RTL nos instantes iniciais do refinamento

Como se observa, o método BTR-FF-IC é capaz de produzir soluções de melhor qualidade mais rapidamente que o método VNS-RTL.

5. Conclusão e Trabalhos Futuros

Este trabalho apresenta uma aplicação da metaheurística Busca Tabu na resolução do Problema de Programação de Tripulações de Ônibus Urbano. Foram desenvolvidas várias versões deste método a partir de diferentes critérios de exploração da vizinhança. Todas as versões desenvolvidas foram comparadas com o método VNS-RTL, existente na literatura, tendo produzido soluções de melhor qualidade que este último. Mostrou-se, ainda, que a melhor das versões do método de Busca Tabu, BTR-FF-IC, é capaz de gerar soluções de melhor qualidade mais rapidamente que o método VNS-RTL. Ademais, é o método mais robusto, pois foi o que apresentou o menor desvio em relação ao valor da melhor solução encontrada. Comparando a solução obtida pelo método BTR-FF-IC com a da empresa, esse método reduziu o número de horas-extras e o de jornadas e, conseqüentemente, de tripulações necessárias, contribuindo para a redução dos custos com a mão-de-obra.

Como trabalho futuro propõe-se a implementação de estratégias de diversificação com memória, de forma a melhorar o desempenho do método.

Agradecimentos

Este trabalho foi parcialmente financiado pela CAPES e CNPq.

Referências

- ANTP - Associação Nacional dos Transportes Públicos – website. Disponível em: <http://www.antp.org.br>.
- BLAIS, J. -Y.; LAMONT, J.; ROUSSEAU, J. -M (1990) – The HASTUS vehicle and manpower scheduling system at the Societé de transport de la Communauté urbaine de Montréal. *Interfaces*, v. 20, n. 1, p. 26-42.
- BOUZADA, C. F. (2002)– Análise das despesas administrativas no custo do transporte coletivo por ônibus no município de Belo Horizonte. Dissertação de mestrado, Escola do Governo, Fundação João Pinheiro, Belo Horizonte.
- DADUNA, J. R.; BRANCO, I; PAIXÃO, J. M. P. (1995) – Computer-Aided Transit Scheduling. Berlin: Springer-Verlag (Lecture Notes in Economics and Mathematical Systems, v. 430).
- DADUNA, J. R.; VOSS, S. (2001) – Computer-Aided Scheduling of Public Transport. Berlin: Springer Publishers (Lecture Notes in Economics and Mathematical Systems, v. 505).
- DADUNA, J. R.; WREN, A. (1988) – Computer-Aided Scheduling of Public Transport. Berlin: Springer-Verlag. (Lecture Notes in Economics and Mathematical Systems, v. 308).
- DESROCHERS, M.; ROUSSEAU, J. -M (1992) – Computer-Aided Transit Scheduling. Berlin: Springer-Verlag. (Lecture Notes in Economics and Mathematical Systems, v. 386).
- DESROCHERS, M.; SOUMIS, F. (1989) – A column generation approach to the urban transit crew scheduling problem. *Transportation Science*, v. 23, p. 1-13.
- ERNST, A. T.; JIANG, H.; KRISHNAMOORTHY, M; SIER, D. (2004a) – Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, v. 153, p. 3-27.
- ERNST, A. T.; JIANG, H.; KRISHNAMOORTHY, M; OWENS, B.; SIER, D. (2004b) – An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, v. 127, p. 21-144.
- FORES, S.; PROLL, L.; WREN, A. (1999) – An improved ILP system for driver scheduling. In: WILSON, N. H. M. (Ed.). *Computer-Aided Transit Scheduling*. Berlin: Springer-Verlag, p.43-62.
- GAREY, M. R.; JOHNSON, D. S. (1979) – Computers and Intractability - A Guide to the Theory of NP-Completeness, Victor Klee Editor.
- GLOVER, F. (1986) – Future paths for integer programming and artificial intelligence. *Computers & Operations Research*, v. 13, p. 533-549.
- HANSEN, P. (1986) – The steepest ascent mildest descent heuristic for combinatorial programming. Congress on Numerical Methods in Combinatorial Optimization. Capri.
- LOURENÇO, H. R.; PAIXÃO, J. P.; PORTUGAL, R. (2001) – Multiobjective metaheuristics for the bus-driver scheduling problem. *Transportation Science*, v. 35, p. 331-343.
- ROUSSEAU, J. -M (1985) – Computer Scheduling of Public Transport 2. Amsterdam: North-Holland.
- SHEN, Y.; KWAN, R. S. K. (2001) – Tabu Search for Driver Scheduling. In: DADUNA, J. R.; VOSS, S. (Ed.). *Computer-Aided Scheduling of Public Transport*. Berlin: Springer-Verlag, p. 121-135.
- SOUZA, M. J. F.; CARDOSO, L. X. T.; SILVA, G. P.; RODRIGUES, M. M. S.; MAPA, S. M. (2004) – Metaheurísticas Aplicadas ao Problema de Programação de Tripulações no Sistema de Transporte Público, *Tendências em Matemática Aplicada e Computacional*, v. 5, 12p. (no prelo).
- WILSON, N. H. M. (1999) – Computer-Aided Transit Scheduling. Berlin: Springer-Verlag (Lecture Notes in Economics and Mathematical Systems, v. 471)

WREN, A. (Ed.) (1981) – Computer Scheduling of Public Transport. Amsterdam: North-Holland.

WREN, A.; KWAN, R. S. K.; KWAN, A. S. K (2000) – Hybrid genetic algorithms for scheduling bus and train drivers. In: WILSON, N. H. M. (Ed.). *Proceedings of the 2000 Congress on Evolutionary Computation CEC 00*, p. 285-292.

WREN, A.; ROUSSEAU, J. -M. (1995) – Bus driver scheduling - an overview. In: DADUNA, J. R.; BRANCO, I.; PAIXÃO, J. M. P. (Ed.). *Computer-Aided Transit Scheduling*, Berlin: Springer-Verlag, p. 173-187.