

# RESOLUÇÃO DO PROBLEMA DE RODÍZIO DE TRIPULAÇÕES DE ÔNIBUS URBANO VIA *SIMULATED ANNEALING* E *ITERATED LOCAL SEARCH*

**Túlio Ângelo Machado Toffolo**  
**Marcone Jamilson Freitas Souza**  
**Gustavo Peixoto Silva**

Departamento de Computação  
Universidade Federal de Ouro Preto

## RESUMO

Este trabalho aborda o Problema de Rodízio de Tripulações de Ônibus Urbano, mais conhecido na literatura como *Crew Rostering Problem*. Tal problema consiste em atribuir as jornadas de trabalho dos dias úteis, sábados e domingos/feriados aos respectivos tripulantes da empresa, de forma a que a carga total de trabalho de cada tripulação seja a mais equilibrada possível dentro de um horizonte de planejamento. Na resolução deste problema, devem ser levadas em consideração restrições legais e operacionais tais como o tempo mínimo de descanso entre o final de uma jornada e o início da jornada do dia seguinte, o dia de descanso da tripulação após um determinado número de dias de trabalho, pelo menos um dia de descanso no mês ocorrendo no domingo, entre outras. Neste trabalho o problema foi resolvido pelos métodos *Simulated Annealing* e *Iterated Local Search*, os quais operam com movimentos baseados na troca de jornadas entre tripulantes. Cada solução gerada é avaliada por uma função baseada em penalidades, a qual possui componentes que avaliam o cumprimento da legislação trabalhista, o respeito às regras operacionais da empresa, bem como se a escala mensal tem a carga de trabalho entre os tripulantes equânime. Resultados computacionais mostram que os métodos são capazes de gerar soluções viáveis rapidamente, sendo o método *Iterated Local Search* capaz de produzir soluções finais de melhor qualidade.

## ABSTRACT

This work deals with the Bus Crews' Rostering Problem (CRP) related to a company operating in a public mass transit. Such problem consists in assigning the set of a company duties related to work days, Saturdays and Sundays, of a given period, to a set of drivers in order to have an even distribution of the workload and extra payment in the period. A solution to the CRP have to consider labor and operational constraints such as: the minimum rest time between the end of a duty and the beginning of the duty in the next day, how the rest day have to be interspersed between work days, at least one rest day have to be in the Sunday, and so on. In this work the CRP was solved through the Simulated Annealing and Iterated Local Search metaheuristics, processing with movements based on the change of duties between a pair of crews. 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. Computational results show that the methods produce feasible solutions quickly and the Iterated Local Search can produce better final solutions.

## 1. INTRODUÇÃO

O Problema de Programação Mensal da Tripulação, ou Problema de Rodízio da Tripulação tem como ponto de partida as jornadas a serem executadas nos dias úteis, sábados, domingos e feriados de um dado período, que normalmente corresponde a um mês. As jornadas dos dias úteis são todas iguais, assim como são coincidentes as jornadas dos domingos e dos feriados. Entretanto, as jornadas dos dias úteis, dos sábados e dos domingos/feriados diferem umas das outras. Tais jornadas diárias, ditas programação das tripulações, são consideradas dados de entrada fornecidos pela empresa ou por algum método de otimização que resolva o Problema de Programação Diário da Tripulação (PPDT). De qualquer maneira, as programações das tripulações são suficientes para a realização de todas as viagens sob a responsabilidade da empresa. Assim, para cada tipo de dia é conhecido exatamente o número necessário de tripulações para a execução das respectivas jornadas. A resolução do Problema de Rodízio da Tripulação deve especificar a sequência das jornadas diárias a serem executadas por cada tripulação da empresa ao longo do mês. Tal sequência de jornada é denominada, na prática, de Escala Mensal dos Motoristas e Cobradores e na literatura especializada de *Crew Rostering*,

ou Rodízio da Tripulação. Neste trabalho o problema será doravante denominado Problema de Rodízio da Tripulação (PRT).

A programação da tripulação normalmente cobre um intervalo de tempo restrito. No sistema de transporte público, este período é tipicamente de um dia e uma tripulação executa a mesma programação todos os dias da semana, com exceção dos sábados e domingos. Este sistema é preferido tanto pelas tripulações quanto pelas empresas, uma vez que permite que as tripulações se familiarizem com as respectivas linhas nas quais atuarão. Entretanto, cada jornada tem seu custo e dificuldades particulares. Assim, as jornadas são atribuídas às tripulações usando como prioridade o critério de “tempo de casa”. Desta maneira, é construído um rodízio com desigualdades entre o conjunto de jornadas de cada tripulação. Normalmente as tripulações recebem salários equivalentes e, portanto, é necessário garantir que todas as escalas mensais tenham as mesmas dificuldades e atratividades. No caso de haver horas extras, estas devem estar igualmente distribuídas entre as tripulações. Isso pode ser obtido intercalando-se as escalas diárias das tripulações para que os rodízios das tripulações tenham basicamente as mesmas características.

Na resolução do PRT devem ser levadas em conta restrições legais e operacionais tais como: a existência de um tempo mínimo de descanso entre o final de uma jornada e o início da jornada no dia seguinte; a ocorrência de um dia de descanso após um determinado período contínuo de trabalho; a ocorrência de pelo menos um domingo por mês de descanso para cada tripulação; o total de horas trabalhadas deve estar entre valores mínimos e máximos, entre outros.

Os Problemas de Programação Diária e de Rodízio da Tripulação são bastante estudados com aplicações na área de transportes, seja aeroviário, ferroviário, transporte urbano e interurbano. Devido à magnitude e impacto nos custos, a programação diária e o rodízio da tripulação do setor de transporte aéreo é sem dúvida nenhuma aquela que tem recebido maior atenção por parte dos pesquisadores, dentre todos os sistemas de transporte. Recentemente surgiram diversas aplicações de modelos ao setor de transporte público ferroviário, que assim como no setor aeroviário, dividindo o problema nos seguintes subproblemas: geração de jornadas viáveis, otimização das jornadas e rodízio da tripulação (Ernst *et al.*, 2004a).

Em relação ao transporte público por ônibus urbano, os trabalhos se concentraram principalmente na resolução da programação diária da tripulação, deixando o rodízio da tripulação para ser resolvido manualmente. Ernst *et al.* (2004a) destacam Bodin *et al.* (1983) e Carraresi e Gallo (1984) como os principais trabalhos que abordam o PRT no sistema de transporte público. O interesse pela resolução apenas da etapa diária do problema prevalece ainda hoje, como pode ser confirmado pelas revisões de Ernst *et al.* (2004a) e Ernst *et al.* (2004b) e nas últimas publicações do principal congresso que trata do assunto, ou seja, em Desrochers e Rousseau (1992), Daduna *et al.* (1995), Wilson (1999) e Voss e Daduna (2001). O modelo de escalonamento cíclico é usado para a classe de problemas de trabalhadores com localização fixa e pode ser aplicado a determinados problemas de rodízio (Baker 1974, Apud Bodin *et al.* 1983). Neste caso, o intervalo de tempo  $t$  é de um dia e a demanda  $dt$  é o número de tripulações requeridas para o dia  $t$ . Na forma simplificada do problema, o horizonte de programação é de uma semana e os  $dt$ 's são iguais para os dias úteis. Casos particulares de problemas de programação de trabalhadores com localização fixa com todos os  $c_j = 1$  podem ser facilmente resolvidos em tempo polinomial simplesmente resolvendo um sistema linear

(Baker 1974, apud Bodin *et al.* 1983). Pequenas variações deste problema também podem ser resolvidas eficientemente. O modelo de escalonamento cíclico considera todas as jornadas de um dado dia com pesos iguais, o que em determinadas situações não ocorre. O objetivo típico para o rodízio no sistema de transporte público é balancear o trabalho requerido entre todas as tripulações, uma vez que cada tripulação recebe o mesmo pagamento independentemente do trabalho executado. Neste caso, o objetivo é produzir rodízios individuais resultando em um plano geral de trabalho o mais equânime possível em termos do total de horas trabalhadas ao longo do período de programação.

No modelo apresentado por Carraresi e Gallo (1984), a cada jornada é atribuído um peso, que representa uma medida do custo da jornada para a tripulação. Tal peso pode ser dado pela duração da jornada ou pode ser em função da atratividade da jornada, do tempo total da jornada em caso de dupla-pegada, o tempo gasto pela tripulação para se deslocar da garagem até o ponto onde a jornada se inicia, entre outros. O problema de se encontrar um conjunto de jornadas balanceadas ao longo de um dado intervalo de tempo é formulado por meio do seguinte problema de gargalo: “minimizar a soma dos pesos das jornadas designadas à tripulação, que tem a pior configuração”. Para este problema os autores apresentam uma heurística que encontra uma solução sub-ótima. É apresentado um algoritmo para o problema de designação com gargalo, o qual é utilizado como sub-rotina para resolver o problema de rodízio da tripulação.

Para encontrar  $m$  seqüências de jornadas tal que a carga máxima de trabalho seja minimizada Carraresi e Gallo (1984) definem uma rede  $n$ -níveis  $G = (N, A)$ , onde  $n$  é o número de dias no período sobre o qual o rodízio deve se estender e  $m$  é o número máximo de tarefas por dia. Os autores assumem que o mesmo conjunto de jornadas deve se repetir a cada dia. Um nó na rede, denotado por  $\langle i, k \rangle$  representa a jornada  $i$  no dia (nível)  $k$ . E um arco do nó  $\langle i, k \rangle$  para o nó  $\langle j, k+1 \rangle$  é incluído na rede se as jornadas  $i$  e  $j$  puderem ser executadas pela mesma tripulação, em dias consecutivos. Denota-se tais arcos por  $(i, j; k)$  com fluxo  $x_{ij}^k$  assumindo valores zero ou um. Um peso  $w_i$  é associado a cada jornada  $i$ , representando o seu custo para a tripulação que o executará. Assim, o problema se resume em encontrar na rede  $G$ ,  $m$  caminhos disjuntos com origem no primeiro conjunto de nós  $\{\langle 1, 1 \rangle, \langle 2, 1 \rangle, \dots, \langle m, 1 \rangle\}$  e destino no último conjunto de nós  $\{\langle 1, n \rangle, \langle 2, n \rangle, \dots, \langle m, n \rangle\}$ , tal que o caminho mais longo tenha um comprimento mínimo. Para  $m = 2$  tem-se o Problema Padrão de Designação com Gargalo. Para  $m > 2$ , o problema é denominado Problema de Designação Multi-nível com Gargalo. Os autores mostram ainda que o PRT é um problema NP-difícil e apresentam uma heurística que resolve iterativamente vários problemas de assinalamento com gargalo sobre os nós nos níveis  $k$  e  $k+1$ .

Deve ficar claro que nenhuma das abordagens descritas anteriormente fornece uma solução completa para o problema de rodízio da tripulação. A função objetivo usada no modelo de Baker (1974) não é realista e o modelo de Carraresi e Gallo (1984) mantém as folgas fixas no final de semana, não prevendo atividades para os sábados, domingos e feriados. Os modelos apresentados não são facilmente ampliáveis para horizontes mais amplos, mas podem ser úteis quando combinados com procedimentos que tratam de outros aspectos do problema.

Bianco *et al.* (1992) formulam o PRT através de um modelo de programação linear inteira e descrevem uma heurística que utiliza o limitante inferior proveniente da formulação

matemática para reduzir as dimensões do problema. A cada iteração o algoritmo resolve um problema de designação multi-nível com gargalo, para o qual é proposto um novo procedimento que produz soluções que convergem assintoticamente para o ótimo.

No Brasil, Yunes (2000) abordou o PPDT e o PRT utilizando técnicas de Programação Matemática e de Programação por Restrições. Para o PPDT foi desenvolvido um algoritmo híbrido de geração de colunas combinando às duas técnicas mencionadas. O PRT foi tratado pelo autor de forma parecida. Foram utilizadas duas metodologias para resolver esse problema. Na primeira abordagem o problema é modelado como um problema de programação linear inteira, e resolvido com a técnica de *branch and bound*. A segunda abordagem se baseia em programação por restrições. A primeira abordagem só é aplicável a problemas de pequeno porte, e ainda se mostrou menos eficaz do que a técnica Programação por Restrições.

Embora poucos trabalhos sobre o PRT no sistema de transporte por ônibus urbano tenham sido desenvolvidos, a prática de manter as tripulações nas mesmas jornadas ao longo do período, deve ser revista com o objetivo de gerar seqüências de jornadas capazes de equilibrar a carga de trabalho entre todas as tripulações e diminuir a quantidade de horas extras pagas pela empresa ao final do período. Nesse sentido é que se torna importante estudar o PRT e propor métodos para a sua resolução.

O que se espera com a realização de novos estudos sobre o rodízio da tripulação, é verificar se pequenas variações nas jornadas diárias das tripulações não possibilitam obter uma carga mais justa de trabalho, remuneração e descanso para o conjunto dos trabalhadores da empresa, assim como a redução dos custos com a mão de obra. Este pode ser um fator de motivação para o desenvolvimento de um número maior de trabalhos sobre esta última etapa do planejamento do sistema de transporte público que é o rodízio da tripulação.

Este trabalho está organizado como segue. A seção seguinte apresenta, em detalhes, o problema abordado. A seção 3 detalha a metodologia usada para resolvê-lo. A seção 4 apresenta os resultados obtidos e a seção 5 conclui o trabalho.

## **2. O PROBLEMA DE RODÍZIO DA TRIPULAÇÃO**

O Problema de Rodízio da Tripulação (PRT) é feita após a programação diária e tendo esta como dado de entrada. As jornadas diárias são alocadas aos tripulantes compondo uma escala mensal. Como nesta escala pode haver mudança nas jornadas executadas ao longo do período, justifica-se a denominação de rodízio da tripulação para o problema. O PRT se refere ao seqüenciamento das jornadas diárias ao longo de um período, em geral de um mês, para formar o plano de trabalho e descanso de cada tripulação, durante tal período. Um bom rodízio é aquele que distribui a carga de trabalho e a remuneração extra da forma mais equilibrada possível dentro do mês. As restrições do problema se referem às regras de alocação das folgas e o tempo de descanso entre duas jornadas consecutivas de trabalho, contidas nos acordos coletivos da categoria. Normalmente, as tripulações se mantêm no mesmo turno de trabalho (manhã, tarde e noite) ao longo da semana, sendo que a escolha dos turnos segue, via de regra, uma ordem de “tempo de casa” das tripulações. Assim, os motoristas e cobradores mais velhos opinam por trabalhar nos primeiros horários do dia e os novatos ficam com os horários restantes.

Na programação mensal da tripulação, as jornadas diárias terão características próprias de acordo com o tipo do dia que está sendo programado. As jornadas dos sábados, domingos e feriados devem ser do tipo Pegada Simples, ou seja, não pode haver uma interrupção não remunerada de pelo menos duas horas durante o período. As jornadas do tipo Pegada Dupla são aquelas que apresentam uma interrupção conforme descrito acima.

Ao se resolver o PRT deve-se levar em conta as seguintes restrições e objetivos:

- O intervalo de tempo entre jornadas deve ser maior ou igual a 11 horas;
- Durante todos os dias úteis de um mês de trabalho, uma tripulação faz somente um tipo de pegada: Simples ou Dupla;
- Uma tripulação não pode trabalhar mais do que seis dias consecutivos sem uma folga;
- As tripulações que fazem Pegadas Duplas têm direito a folgar no domingo;
- As tripulações que fazem Pegadas Simples têm direito a pelo menos uma folga no domingo durante o mês;
- Deve haver uma uniformidade na construção da escala mensal das tripulações de forma que exista um baixo desvio das horas trabalhadas no mês, por uma tripulação específica, em relação à média das horas totais trabalhadas pelas tripulações;
- As regras operacionais da empresa devem ser atendidas;

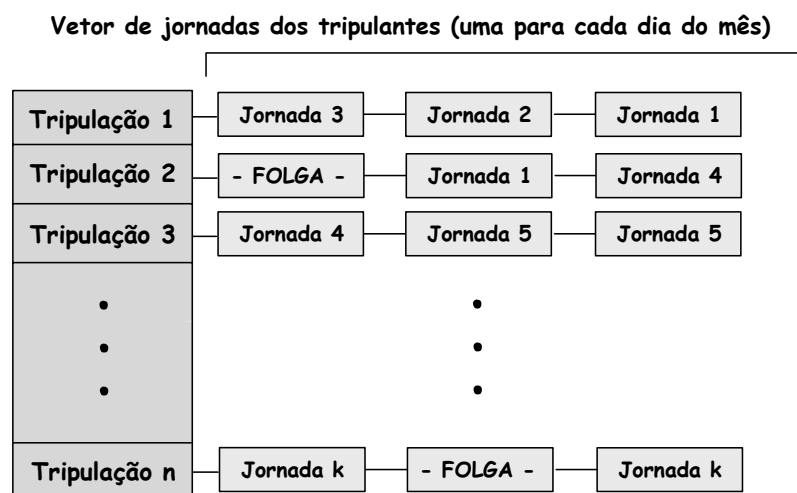
Devido à complexidade da resolução, na otimalidade, de casos reais do problema, uma alternativa é a utilização de metaheurísticas, como as que se propõe na próxima seção.

### 3. METODOLOGIA

Nas subseções seguintes são descritas a forma de representação do problema, o procedimento de geração da solução inicial, o movimento considerado para explorar o espaço de soluções, a função de avaliação usada para guiar a busca e a adaptação de duas metaheurísticas ao PRT.

#### 3.1. Representação

Uma solução do PRT pode ser vista como uma matriz, onde as linhas são os tripulantes e as colunas são as jornadas. Embora a melhor maneira de visualizar uma solução seja pensar numa matriz, a implementação computacional é feita como uma lista de listas. A Figura 1 ilustra a representação de uma solução.



**Figura 1:** Representação de uma solução do PRT

### 3.2. Geração de uma solução inicial

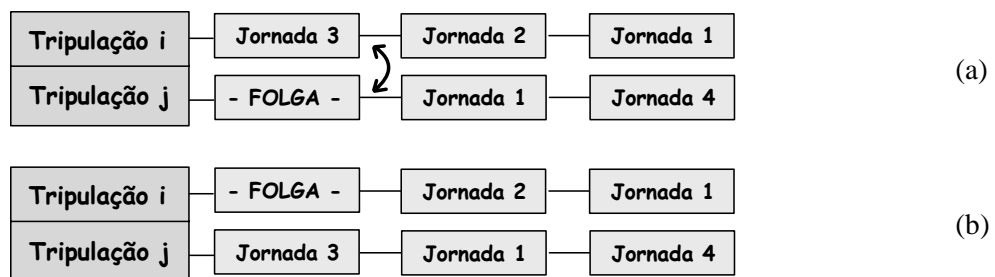
Uma solução inicial para o PRT segundo a abordagem deste trabalho é produzida atribuindo-se, inicialmente, as  $d$  jornadas diárias de pegadas duplas aos  $d$  primeiros tripulantes e as seguintes  $u$  jornadas diárias de pegadas simples aos demais tripulantes. A seguir, é feito o mesmo procedimento para distribuir as jornadas dos sábados, domingos e feriados.

Uma vez que todas as jornadas estão alocadas, são criados outros  $t$  tripulantes, cada qual com todas as jornadas diárias sendo folgas, de forma a possibilitar a alocação de folgas a todos os tripulantes e satisfazer as restrições trabalhistas. Como são conhecidos a quantidade de jornadas e o número de folgas necessárias para cada tripulante, pode-se estimar o valor de  $t$ . Esse valor deve ser grande o suficiente para permitir que todos os tripulantes folguem em pelo menos um domingo. Para dar maior flexibilidade aos métodos de refinamento, é criado um total de  $d + u + t + x$  tripulantes, onde  $x$  é um parâmetro que indica um número de tripulantes adicionais, com jornadas do tipo folga, a serem criados. Para resultados satisfatórios, o valor de  $x$  não deve ser grande, podendo inclusive ser nulo, dependendo dos dados de entrada.

### 3.3. Estrutura de Vizinhança

Para gerar um vizinho no espaço de soluções do PRT, o algoritmo trabalha com o movimento Trocar Jornada, o qual consiste em trocar uma jornada diária de uma tripulação  $i$  com a jornada correspondente de uma tripulação  $j$ . O movimento é aplicado sob certas condições. Como existem dois grupos de tripulantes: os de pegadas simples e os de pegadas duplas, as trocas apenas envolverão duas jornadas de pegada dupla ou duas jornadas de pegada simples. Portanto, uma jornada de pegada simples jamais será trocada por uma de pegada dupla e vice-versa. Para gerar esse movimento são analisadas duas tripulações,  $i$  e  $j$ . A seguir, dada uma jornada de um dia  $k$ , efetua-se a troca dessa jornada do dia  $k$  da tripulação  $i$  pela jornada do dia  $k$  da tripulação  $j$ . Essa troca ocorre mesmo se uma das jornadas envolvidas for uma folga.

No exemplo da Figura 2 o movimento é ilustrado. Em (b) mostra-se que a solução vizinha é gerada pela troca da jornada 3 da tripulação  $i$  com a folga da tripulação  $j$  no primeiro dia do período considerado.



**Figura 2:** Movimento de troca de jornadas no PRT

### 3.4. Função de Avaliação

Uma escala mensal (solução)  $s$  é avaliada por uma função de avaliação (FO), que pode ser expressa de acordo com a fórmula (1):

$$FO(s) = F_{RE} + F_{RNE} \quad (1)$$

sendo  $F_{RE}$ : função que avalia o não-atendimento aos requisitos essenciais do rodízio e  $F_{RNE}$ : função que avalia o não-atendimento aos requisitos não essenciais.

i) Determinação de  $F_{RE}$  – Esta parte da função avalia, pela fórmula (2), o não atendimento aos requisitos considerados essenciais no problema do rodízio da tripulação.

$$F_{RE} = P_1 I_1 + P_2 I_2 + P_3 I_3 \quad (2)$$

em que:

- $I_1$  : Tempo entre jornadas: É o tempo que falta, em minutos, para completar 11:00h entre o horário de fim da jornada de um dia e o horário de início da jornada do dia seguinte;
  - $I_2$  : Número de folgas: É o número de dias que ultrapassa o tempo máximo de trabalho sem folga, no caso, de seis dias;
  - $I_3$  : Número de tripulantes sem folga no domingo: É o número de tripulantes que fazem pegada simples e não têm folgas aos domingos dentro do mês de trabalho;
- $P_1, P_2$  e  $P_3$  são os pesos referentes, respectivamente, a  $I_1, I_2$  e  $I_3$ .

ii) Determinação de  $F_{RNE}$  – Esta segunda parte da função de avaliação tem como finalidade minimizar os custos operacionais e aproveitar ao máximo o tempo de cada tripulação. Além disso, visa também a um equilíbrio das jornadas, com relação à duração, entre as tripulações dentro do período. A fórmula (3) avalia  $F_{RNE}$ :

$$F_{RNE} = p_1 R_1 + p_2 R_2 + p_3 R_3 + p_4 R_4 + p_5 R_5 + \dots + p_{11} R_{11} \quad (3)$$

sendo:

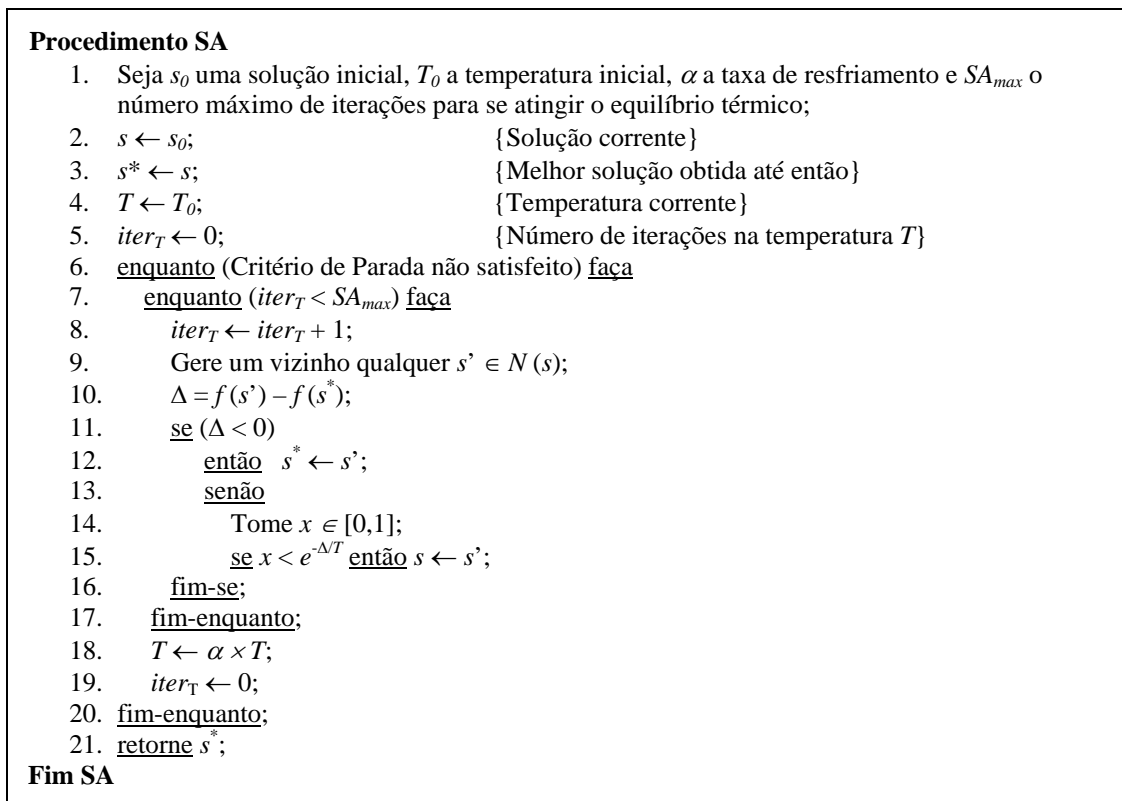
- $R_1$  = *Distribuição de folgas*: É a soma, para cada tripulante, da quantidade de folgas em falta ou excesso em relação ao número mínimo  $NF_{\min}$  e número máximo  $NF_{\max}$  de folgas estipuladas, no caso, 4 e 6. É usado para garantir uma distribuição mais equilibrado das folgas;
  - $R_2$  = *Nº de trocas de linhas*: É a soma do número de vezes que cada tripulação troca de linha;
  - $R_3$  = *Nº de linhas diferentes*: É a soma do número de linhas diferentes contidas na jornada mensal de cada tripulação;
  - $R_4$  = *Nº de trocas do período de trabalho*: Este item contabiliza quantas vezes no rodízio houve troca de período de trabalho para cada tripulação. Tem por objetivo uniformizar os períodos de trabalho das tripulações;
  - $R_5$  = *Trocas de jornadas nos dias úteis*: É a soma do número de vezes que cada tripulação troca de jornada durante os dias úteis;
  - $R_6$  = *Trocas de jornadas nos sábados*: É a soma do número de vezes que cada tripulação troca de jornada contando apenas os sábados;
  - $R_7$  = *Trocas de jornadas nos domingos*: É a soma do número de vezes que cada tripulação troca de jornada considerando apenas os domingos;
  - $R_8$  = *Jornadas diferentes nos dias úteis*: É a soma do número de jornadas de dias úteis diferentes para cada tripulação;
  - $R_9$  = *Jornadas diferentes nos sábados*: É a soma do número de jornadas diferentes de sábados para cada tripulação;
  - $R_{10}$  = *Jornadas diferentes nos domingos*: É a soma do número de jornadas diferentes de domingos para cada tripulação;
  - $R_{11}$  = *Diferença do tempo médio de trabalho*: É a soma da diferença entre o tempo de trabalho de cada tripulação e o tempo médio de trabalho de todas as tripulações;
- $p_1, p_2, p_3, \dots, p_{11}$  são os pesos referentes, respectivamente, aos requisitos  $R_1, R_2, R_3, \dots, R_{11}$ .

### 3.5. Técnicas metaheurísticas aplicadas ao PRT

O PRT foi abordado através das metodologias *Simulated Annealing* e *Iterated Local Search* (ILS), esta última utilizando como método de busca local o método Descida Randômica (DR).

#### 3.5.1. *Simulated Annealing* (SA) aplicado ao PRT

*Simulated Annealing* é uma metaheurística proposta por Kirkpatrick *et al.* (1983), sendo uma técnica de busca probabilística que se fundamenta em uma analogia com a termodinâmica, ao simular o resfriamento de um conjunto de átomos aquecidos. O pseudocódigo do algoritmo é apresentado pela Figura 3.



**Figura 3:** Algoritmo *Simulated Annealing*

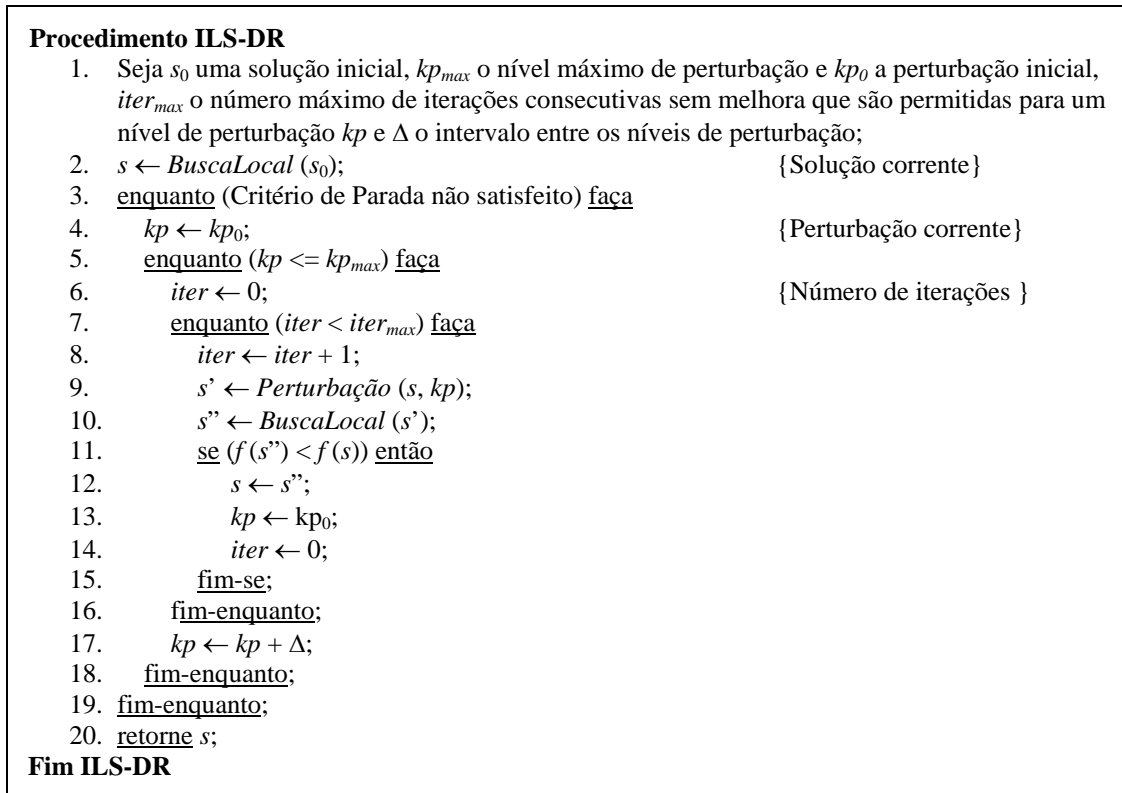
Para gerar um vizinho aleatório pelo método (linha 9 da Figura 3) procede-se como segue. Sorteiam-se dois tripulantes diferentes  $i$  e  $j$  e um dia  $k$  do mês e efetua-se a troca de jornadas desse dia entre eles. No caso de um dos tripulantes, por exemplo, o tripulante  $i$ , tiver problemas com a alocação de folgas (no caso, estar trabalhando a mais de seis dias consecutivos), então procura-se aleatoriamente no outro tripulante um dia  $l$  do mês cuja jornada seja uma folga. Esta procura ocorre por no máximo  $n_{dias}$ , sendo  $n_{dias}$  o número de dias do mês. Não encontrando uma folga no tripulante  $j$ , então são sorteados dois novos tripulantes, repetindo-se o processo. Encontrando-se uma folga no tripulante  $j$ , faz-se a troca de jornadas dos tripulantes  $i$  e  $j$  relativas ao dia  $l$ .

Os parâmetros de controle do procedimento são a razão de resfriamento  $\alpha$ , o número de iterações para cada temperatura ( $SA_{max}$ ) e a temperatura inicial  $T_0$ . O último parâmetro é obtido por simulação, sendo a temperatura na qual 50% dos vizinhos são aceitos.

Originalmente, o método pára quando a temperatura aproxima-se de zero, evidenciando o encontro de um ótimo local. No método proposto, considera-se como critério de parada o máximo tempo de processamento. No caso de o sistema congelar antes de atingir o tempo máximo de processamento, a temperatura é reaquecida conforme procedimento a seguir. A cada reaquecimento, a temperatura final é acrescida em  $\eta \times \varphi$ , onde  $\eta$  é o número de reaquecimentos e  $\varphi$  a temperatura na qual 5% dos vizinhos são aceitos, sendo este último parâmetro determinado por simulação no início da aplicação do algoritmo.

#### 3.5.4. Iterated Local Search (ILS) e Descida Randômica (DR) aplicado ao PRT

Para refinar uma solução gerada pelo método descrito na seção 3.2, foi proposto um método híbrido, denominado ILS-DR, composto pela metaheurística ILS (*Iterated Local Search*) (Lourenço *et al.*, 2002) e pelo método de Descida Randômica (DR), sendo este usado como um mecanismo de busca local para o ILS. O pseudocódigo do procedimento ILS-DR adaptado ao PRT é apresentado na Figura 4.



**Figura 4:** Algoritmo ILS-DR

Esse algoritmo inicia-se a partir de uma solução inicial  $s_0$  gerada pela metodologia descrita na seção 3.2. Em seguida, aplica-se um mecanismo de busca local, o método Descida Randômica (DR). Este método se comporta como um método de descida clássico, porém analisa somente um único vizinho a cada iteração, gerado de forma análoga à do método *Simulated Annealing*. O método DR pára quando um número máximo  $iter_{DR_{max}}$  de iterações sem melhora for atingido. A cada iteração do método ILS-DR, gera-se uma perturbação na solução corrente. Essa perturbação consiste em realizar  $k$  movimentos aleatórios de troca de jornadas, sendo  $k$  um número arbitrário compreendido entre um e  $kp$ . A essa nova solução vizinha  $s'$ , aplica-se o

DR, gerando uma solução vizinha ótima local,  $s''$ . Se a solução  $s''$  for melhor que a solução  $s$  corrente, então  $s''$  é aceita como a nova solução corrente e reinicia-se o valor de  $kp$ . Essa repetição é executada até que  $iter_{max}$  iterações sem melhora sejam realizadas. Quando isso ocorrer, incrementa-se o grau de perturbação,  $kp$ , em um fator  $\Delta$ , até que  $kp$  atinja seu valor máximo ( $kp_{max}$ ).

#### 4. RESULTADOS

Os algoritmos foram desenvolvidos na linguagem C++ usando o compilador do Borland C++ Builder 6.0 e testados em um microcomputador com processador Pentium IV 2,5 MHz, com 384 MB de memória RAM, sob sistema operacional Windows XP Service Pack 2. Os testes foram realizados com dados de uma empresa de transporte público que opera em Belo Horizonte, MG. Considerou-se, para os testes, o mês de maio de 2005, contendo um total de 5 domingos, 4 sábados, 1 feriado e 21 dias úteis. A Tabela 1 mostra o valor dos pesos usados na função de avaliação.

**Tabela 1:** Pesos usados na função de avaliação.

| Requisito                                 | Peso | Requisito                                  | Peso |
|---|------|--|------|
| Tripulante não folga em domingo/feriado   | 250  | Cada folga em falta/excesso do tripulante  | 100  |
| Cada falta/excesso de dias entre 2 folgas | 5000 | Caso haja falta de tempo entre 2 jornadas  | 100  |
| Cada hora além de 5h da média de trabalho | 25   | Cada troca de período de trabalho          | 50   |
| Cada troca de linha                       | 20   | Cada linha diferente                       | 20   |
| Cada jornada diferente em dias úteis      | 3    | Cada troca de jornada em dias úteis        | 5    |
| Cada jornada diferente em sábados         | 1    | Cada troca de jornada em dias úteis        | 1    |
| Cada jornada diferente em dom./feriados   | 1    | Cada troca de jornada em domingos/feriados | 1    |

Os parâmetros adotados no método *Simulated Annealing* foram: Temperatura inicial obtida por simulação (temperatura na qual 50% dos vizinhos são aceitos),  $SA_{max} = 10000$  e Razão de resfriamento  $\alpha = 0,95$ . Para o método ILS-DR os parâmetros usados foram:  $kp_0 = 1$ ,  $kp_{max} = 7$ ,  $iter_{max} = 25$ ,  $\Delta = 3$  e  $iterDR_{max} = 1000$ . O critério de parada adotado em ambos os métodos foi o tempo total de processamento, restrito a 45 minutos.

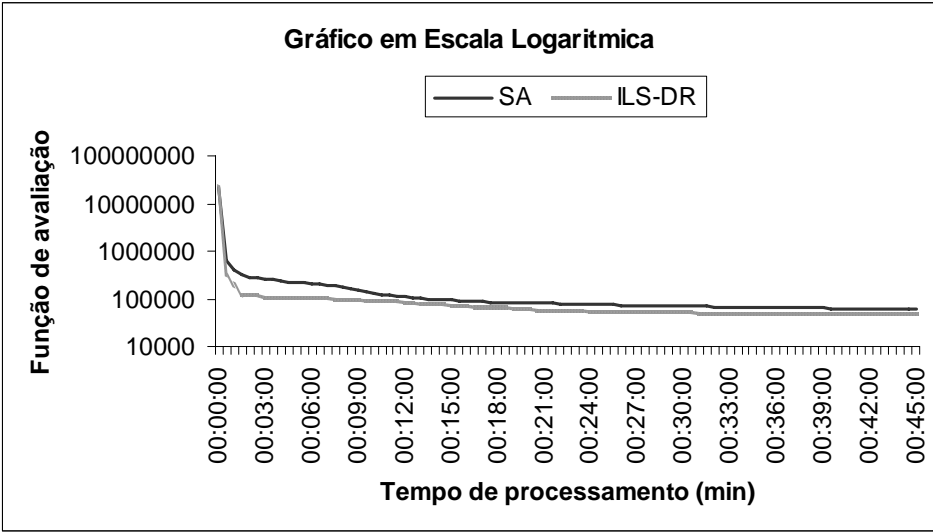
Cada um dos métodos foi executado 10 vezes, cada qual partindo de uma semente diferente de números aleatórios. A Tabela 2 apresenta, para cada algoritmo, o melhor valor obtido, a média entre os valores obtidos e o desvio do valor médio em relação ao melhor valor encontrado para a instância, sendo desvio = (Média – Melhor Valor) / (Melhor Valor).

**Tabela 2:** Desempenho dos algoritmos

| Método       | SA     | ILS-DR |
|--------------|--------|--------|
| Melhor Valor | 61.024 | 48.859 |
| Média        | 64.834 | 51.293 |
| Desvio       | 32,6%  | 4,9%   |

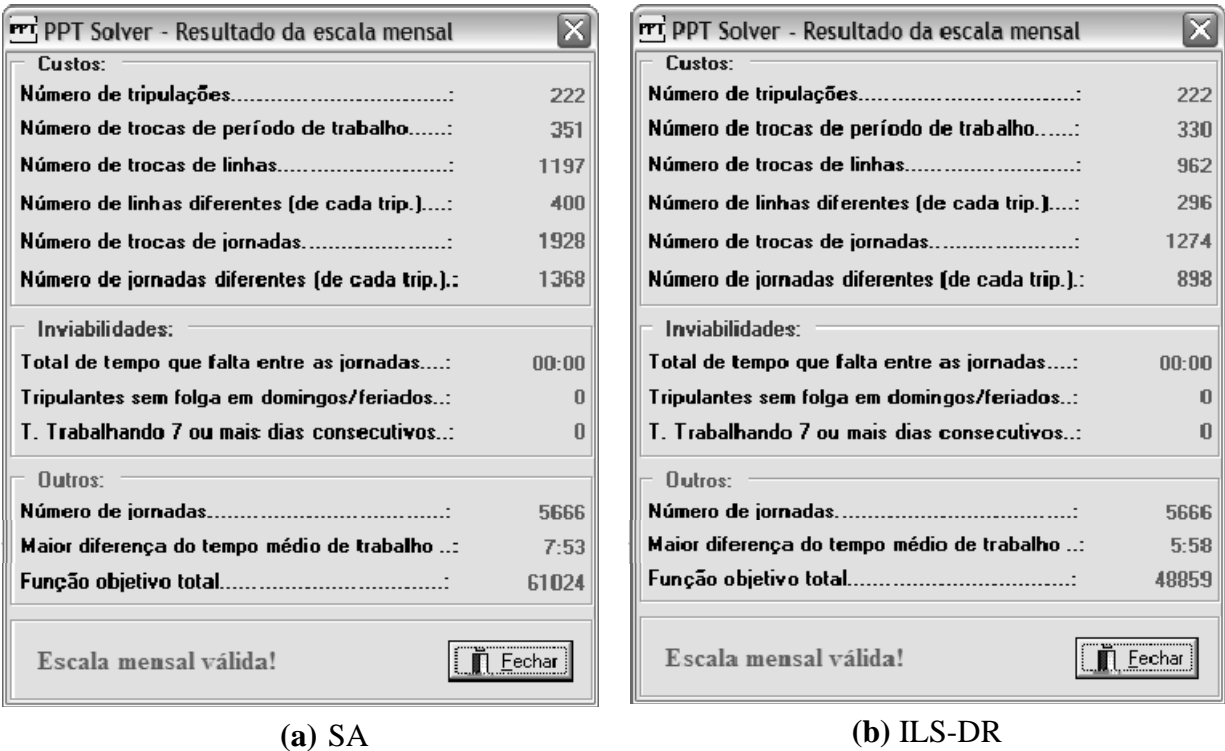
A Tabela 2 mostra que o método ILS-DR, para os parâmetros considerados, foi consideravelmente superior ao método *Simulated Annealing* (SA) na resolução do PRT. Ambos os métodos geraram soluções viáveis rapidamente (cerca de 5 minutos de execução).

A Figura 5 mostra a evolução do valor da melhor solução gerada por cada método ao longo da busca. Ela mostra que, além de gerar soluções de melhor qualidade, o ILS-DR também converge mais rapidamente.



**Figura 5:** Evolução da melhor solução durante a busca nos métodos SA e ILS-DR

A Figura 6 apresenta as principais características das melhores soluções produzidas por cada um dos dois métodos implementados.



**Figura 6:** Características das melhores soluções geradas pelos algoritmos SA e ILS-DR

## 5. CONCLUSÕES

Este trabalho apresenta duas metodologias heurísticas, baseadas em *Simulated Annealing* (SA) e *Iterated Local Search* (ILS), para resolução do Problema de Rodízio de Tripulações. Essas metodologias exploram o espaço de soluções usando movimentos de troca de jornadas. A troca é feita de forma inteligente, tentando minimizar a inviabilidade relativa à quantidade de dias consecutivos sem folga. Os dois métodos são capazes de gerar soluções viáveis rapidamente, sendo o método baseado em ILS capaz de gerar soluções finais de qualidade superior ao do método baseado em SA (considerando os parâmetros adotados para ambos).

Com a flexibilização da operação, permitindo-se a troca de linhas, de jornadas diferentes, de períodos de trabalho etc., obtêm-se uma carga mais justa de trabalho, remuneração e descanso para o conjunto de tripulantes da empresa. Para obter uma escala com tripulantes diferindo em cerca de seis horas mensais da média do conjunto de tripulações da empresa no mês em questão, o método ILS-DR requer que durante o mês ocorra as seguintes flexibilizações para cada tripulante: mudança de uma a duas vezes no turno de trabalho; três trocas de linha; uma troca de linha diferente; seis trocas de jornada e quatro jornadas diferentes.

Observa-se, finalmente, que o sucesso na resolução do PRT depende fortemente da solução do PPDT. Se as escalas diárias de trabalho resultantes da solução do PPDT forem bem distribuídas com relação à carga de trabalho, o esforço na resolução do PRT pode ser substancialmente reduzido.

### Agradecimentos

Os autores agradecem ao CNPq, à FAPEMIG, à BHTRANS e à UFOP pelo apoio ao desenvolvimento deste trabalho.

### REFERÊNCIAS

- BAKER, K. (1974) Scheduling a full-time workforce to meet cyclic staffing requirements. *Management Science*, v. 20, p. 1561-1568.
- BIANCO, L.; BIELLI, M.; MINGOZZI, A.; RICCIARDELLI, S.; SPADONI, M. (1992) A heuristic procedure for the crew rostering problem. *European Journal of Operations Research*, v. 58, n. 2, p. 272-283.
- BODIN, L.; B. GOLDEN; A. ASSAD e M. BALL. (1983) Routing and scheduling of vehicles and crews: The state of the art. *Computers and Operations Research*, v 10, p. 63-211.
- CARRARESI, P.; GALLO, G. (1984) A multi-level bottleneck assignment approach to the bus drivers' rostering problem. *European Journal of Operations Research*, v. 16, p. 163-173.
- DADUNA, J. R.; PAIXÃO, J. M. P. (1995) Vehicle scheduling for public mass transport: an overview. In: Daduna, J. R., I. Branco e J. M. P. Paixão (eds.), *Computer-Aided Transit Scheduling*, Springer, Berlin.
- DESROCHERS, M. e ROUSSEAU, J. M (1992) *Computer-Aided Transit Scheduling*. Springer, Berlin.
- 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.
- KIRKPATRICK, S.; GELLAT, D. C.; VECCHI, M. P. (1983) Optimization by Simulated Annealing. *Science*, v. 220, p. 671-680.
- LOURENCO, H. R., MARTIN, O., STÜTZLE, T. (2002) Iterated Local Search. In F. Glover and G. Kochenberger (eds), *Handbook of Metaheuristics*, p. 321 - 353, Kluwer Academic Publishers.
- VOSS, S.; DADUNA, J. R. (2001) *Computer-Aided Scheduling of Public Transport*. Springer-Verlag, Berlin.
- WILSON, N. H. M. (1999) *Computer-Aided Transit Scheduling*. Springer-Verlag, Berlin.
- YUNES, T. (2000) *Problemas de Escalonamento no Transporte Coletivo: Programação por Restrição e Outras Técnicas*. Dissertação de mestrado. UNICAMP, Campinas.

---

Gustavo Peixoto Silva (gustavo@iceb.ufop.br)  
Departamento de Computação, Universidade Federal de Ouro Preto.  
Campus Universitário, Morro do Cruzeiro.  
35.400.000 – Ouro Preto, MG, Brasil.