



## **MÉTODO HEURÍSTICO DE SOLUÇÃO PARA O PROBLEMA DE CRIAÇÃO DE LINHAS DE TRABALHO PARA EMPRESAS DE ÔNIBUS INTERURBANO**

**Fábio Gavião Avelino de Mélo**

Faculdade de Administração e Informática – FAI  
Av. Antônio de Cássia, 472 – Santa Rita do Sapucaí, MG  
gaviao\_tsi@uol.com.br

**Edson Luiz França Senne**

Universidade Estadual Paulista – UNESP/FEG  
Av. Ariberto Pereira da Cunha, 333 – Guaratinguetá, SP  
elfsenne@feg.unesp.br

### **RESUMO**

Este artigo apresenta uma nova maneira de formular e resolver o problema de escalonamento de motoristas para empresas de transporte rodoviário interurbano. Trata-se de um problema de otimização combinatória de grande porte para o qual é difícil obter uma solução ótima. O método proposto divide o problema em três subproblemas: construção do espaço de solução do problema; obtenção de um bom conjunto de linhas de trabalho; e determinação de um conjunto adequado de padrões de folga. O objetivo é determinar o melhor conjunto possível de linhas de trabalho que satisfaça a todas as restrições operacionais e trabalhistas do problema e minimize o número necessário de motoristas. Testes computacionais para os dois primeiros subproblemas são apresentados e os resultados obtidos são promissores.

**PALAVRAS-CHAVE.** Problemas Multiobjetivos. Escalonamento de pessoal. Escalas de trabalho. Linhas de trabalho. Heurística. Algoritmo genético.

**Área principal:** Otimização Combinatória.

### **ABSTRACT**

This paper presents a new way to formulate and solve the problem of personnel scheduling for the case of intercity bus drivers. This is, in general, a combinatorial optimization problem of large size for which it is difficult to obtain an optimal solution. The proposed method divides the problem in three subproblems: construction of the search space of the problem; searching for a good set of lines of work; and determination of a suited set of shift scheduling. The aim is to determine the best possible set of lines of work that satisfies all operational and labor constraints and minimizes the required number of drivers. Computational experiments for the first two subproblems are presented and the preliminary results are promising.

**KEYWORDS.** Multi-objective problems. Rostering. Working scales. Lines of work. Heuristic. Genetic algorithm.

**Main area:** Combinatorial Optimization.

## 1. Introdução

O problema da construção de linhas de trabalho para empresas de transporte consiste na elaboração de conjuntos de viagens diárias para um determinado período de tempo para posterior atribuição a motoristas. As viagens diárias que fazem parte destes conjuntos devem pertencer a dias distintos dentro do período de programação e representarem uma sequência possível de trabalho a ser realizado pelo motorista tendo em vistas algumas restrições operacionais e trabalhistas. Uma linha de trabalho é uma escala de serviço de um motorista. O conjunto final de linhas de trabalho deve ser otimizado para atender aos objetivos de: 1) distribuir as cargas de trabalho entre as linhas o mais uniformemente possível; 2) reduzir a dispersão na duração das viagens dentro de cada linha; e 3) minimizar o número de motoristas envolvidos na operação.

Este problema, embora tipicamente observado em empresas de transporte rodoviário interurbano, ocorre também em outros tipos de transporte e em várias outras áreas de aplicação relacionadas a problemas de escalonamento de pessoal (MESQUITA; MOZ; PAIAS, 2011), (VANHOUCHE; MAENHOUT, 2010), (SHODI; NORRIS, 2004). Trata-se de um problema de otimização combinatória envolvendo três conjuntos  $X$ ,  $F$  e  $S$ .  $X$  é o conjunto de todas as possíveis linhas de trabalho do espaço de solução do problema.  $F \subseteq X$  é o subconjunto de linhas de  $X$  que são viáveis e que satisfazem as restrições operacionais e trabalhistas do problema.  $S \subseteq F$  é o subconjunto de  $F$  que otimiza os objetivos do problema.

Os problemas de escalonamento de pessoal têm sido estudados desde a década de 1950, (ERNST *et al.*, 2004), (LOURENÇO; PAIXÃO; PORTUGAL, 2001), (MASON; NIELSEN, 1999), (CAPRARA *et al.*, 1997), e são considerados de difícil solução devido à sua natureza combinatória e por serem, em geral, problemas de grande porte. Além disso, por serem multiobjetivos e terem restrições difíceis de formular, muitas vezes tais problemas não são resolvidos por meio de métodos exatos. Por isto, muitos problemas desta natureza usam técnicas heurísticas de solução (RESPICIO; MOZ; PATO, 2013), (BURKE; KENDALL; SOUBEIGA, 2003), (MEISELS; SCHAERF, 2003), (GOODALE; THOMPSON, 2004).

Este trabalho propõe representar o espaço de soluções do problema por meio de um grafo  $G$ , a partir do qual um algoritmo de percorrimento encontra um subconjunto  $F$  de linhas de trabalho viáveis e de boa qualidade, isto é, que cobre todas as viagens diárias da empresa. Sobre este conjunto  $F$  aplica-se um procedimento evolutivo de busca, com o propósito de obter um conjunto de linhas que corresponde a uma boa solução do problema.

Para aplicação deste método, o problema é subdividido em três subproblemas os quais são resolvidos de forma independente. O primeiro é o da criação do espaço de busca e, conseqüentemente, a criação das linhas de trabalho possíveis. O segundo é o da busca por um conjunto de linhas de boa qualidade e viáveis. O terceiro é o da criação de padrões de folga para os motoristas. Um padrão de folga é uma escala especial constituída apenas de dias de expediente intercalados por dias de folga. Para os testes computacionais com o método de solução proposto foram empregados dados reais de uma empresa de transporte rodoviário.

## 2. Conceitos básicos

Sejam  $N$ , o número de viagens diárias que uma empresa deve cumprir e  $P$ , o número de dias de programação ou tamanho da escala/linha de trabalho. Uma viagem é denotada por  $v_i^k$ , onde  $i$  é o dia no qual a viagem ocorre e  $k$  designa uma viagem daquele dia  $i$  dentro do conjunto de viagens diárias da empresa. Uma viagem é função de 5 variáveis:  $v_i^k = f(h_i, h_f, co, cd, dv)$ ,  $\forall i \in \{1, \dots, P\}$ ,  $\forall k \in \{1, \dots, N\}$ , onde:  $h_i$  é o horário de início da viagem,  $h_f$  é o horário de término da viagem,  $co$  é a cidade origem da viagem,  $cd$  é a cidade destino da viagem e  $dv$  é a data da viagem dentro dos  $P$  dias da programação. Uma linha de trabalho é uma sequência possível destas viagens, representada por  $T = \{(v_1^{j_1}, v_2^{j_2}), (v_2^{j_2}, v_3^{j_3}), \dots, (v_{p-1}^{j_{p-1}}, v_p^{j_p})\}$  onde  $j_1, j_2, \dots, j_n$  representam os índices de cada uma das  $N$  viagens diárias da empresa e cada par  $(v_i^{j_k}, v_{i+1}^{j_{k+1}})$  representa uma troca entre viagens consecutivas que o motorista pode fazer respeitando as regras operacionais e trabalhistas da

empresa. Estas regras foram obtidas a partir de informações da empresa pesquisada e agrupadas num conjunto  $R$  constituído por:

- $r_1$ : Deverá haver um período mínimo de descanso de 11 horas entre o horário de término de uma viagem num dia e o horário de partida da viagem consecutiva no próximo dia;
- $r_2$ : A soma das durações das viagens em qualquer linha de trabalho não poderá ser inferior a um mínimo e nem ultrapassar a um valor máximo;
- $r_3$ : A dispersão da duração das viagens dentro das linhas deve estar num nível satisfatório pré-fixado;
- $r_4$ : Todas as viagens da empresa deverão estar cobertas por pelo menos uma das linhas de trabalho do conjunto  $S \subseteq F$ ;
- $r_5$ : Uma viagem diária não deverá ser incluída em mais de uma linha de trabalho do conjunto  $S \subseteq F$ ;
- $r_6$ : O número de dias de trabalho consecutivos entre duas folgas consecutivas não deverá exceder a um limite máximo;
- $r_7$ : Não poderá haver mais folgas do que um valor máximo dentro do período de programação.

O que se deseja é encontrar o conjunto  $S$  de linhas de trabalho que cubra, no máximo uma vez, cada uma das viagens diárias da empresa, sem deixar nenhuma viagem a descoberto, e que ao mesmo tempo, atenda aos três objetivos de minimizar a diferença de duração entre as linhas de trabalho, minimizar a dispersão das viagens dentro de cada linha, e minimizar o número de motoristas.

### 3. O método de solução proposto

O método proposto divide o problema em três subproblemas: (1) construção do espaço de solução; (2) obtenção de um bom conjunto de linhas de trabalho; (3) determinação de um conjunto adequado de padrões de folga para o conjunto de linhas de trabalho obtido. Para a resolução destes três subproblemas propõe-se o método heurístico representado pelo seguinte algoritmo:

1. **Criar** o espaço de busca para o problema, respeitando o conjunto de restrições  $R$  para obtenção do conjunto  $X$  de linhas de trabalho;
2. **Repetir**
  3. **Percorrer** o espaço de busca gerando linhas de trabalho **viáveis**, para obtenção do conjunto  $F: F \subseteq X$ ;
  4. **Buscar** em  $F$  o subconjunto de linhas de trabalho  $S \subset F$  que melhora o valor da função objetivo do problema;
  5. **Otimizar**  $S$ , de modo a obedecer a restrição  $r_5$ ;
6. **Até** que o critério de parada seja atingido;
7. **Resolver** o problema dos padrões de folga;
8. **Associar** cada elemento do conjunto final  $S$  aos padrões de folga encontrados no passo 7.

Os passos deste algoritmo são detalhados nas seções a seguir.

#### 3.1 A construção do espaço de solução

O espaço de busca  $X$  é representado por um grafo  $G=(V,A)$ , cujos vértices  $V$  são as viagens diárias da empresa e o conjunto de arcos  $A$  é constituído pelas possíveis sequências de viagens  $T$ , como definido anteriormente. A Figura 2 ilustra o grafo  $G$  para o caso P3N3, ou seja, uma programação de três dias ( $P = 3$ ) para uma empresa com três viagens por dia ( $N = 3$ ). Uma linha de trabalho (escala de serviço) neste grafo é, por exemplo,  $T = \{(v_1^1, v_2^2), (v_2^2, v_3^1)\}$ .

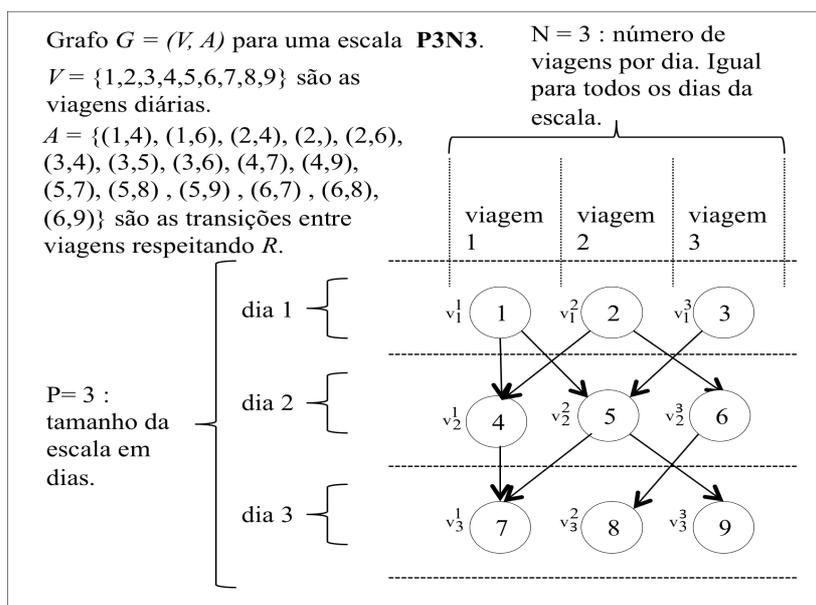


Figura 2 – Grafo  $G$  para o caso P3N3

Uma vez definido o espaço total de busca para o problema, executa-se um algoritmo de percorrimento em profundidade que utiliza uma estratégia de pesos para selecionar o próximo vértice a ser visitado. Durante o percorrimento, um procedimento de filtragem elimina as linhas que não atendem às restrições  $r_2$  e  $r_3$  e procura satisfazer a condição de que o tamanho da linha seja igual a  $P$ . Linhas com tamanho menor do que  $P$  são descartadas. A duração de uma linha  $j$ , dada por:  $\Delta_j = \sum_{i=1}^P \delta(v_i^k)$ , onde  $\delta(v_i^k)$  é a duração da viagem  $v_i^k$ , deve ser tal que:  $LI < \Delta_j < LS$ , onde  $LI$  e  $LS$  são valores que a empresa considera como limites inferior e superior, respectivamente, para que uma escala possa ser atribuída a um motorista. Finalmente, o cálculo de dispersão é feito levando em consideração o número viagens com duração dentro do intervalo  $[\mu_j - \sigma_j, \mu_j + \sigma_j]$  onde  $\mu_j$  é a média das durações das viagens da linha  $j$  e pode ser calculada através da fórmula  $\mu_j = \Delta_j/P$ . O desvio padrão das durações das viagens da linha  $j$  é calculado por:  $\sigma_j = \sqrt{(\sum_{i=1}^P (\delta(v_i^k) - \mu_j)^2)/P}$ . Caso 70% dos valores de duração das viagens de uma linha de trabalho estiverem dentro deste limite a linha é aceita; caso contrário, é rejeitada.

As linhas selecionadas constituem o conjunto  $F$ . O custo da linha  $j$ , é calculado por:  $c_j = (|VI - \Delta_j| \cdot \sigma_j)/VI$ , onde  $VI$  é a duração ideal de uma linha de trabalho, conforme definido pela empresa. Se, por exemplo, define-se que uma viagem ideal tem duração de 6 horas, então  $VI = 6P$ .

### 3.2 A determinação do conjunto de linhas de trabalho

Uma vez gerado o conjunto  $F$ , visando melhorar o valor da função-objetivo, torna-se necessário resolver um novo problema: determinar o subconjunto  $S \subseteq F$ . Este novo problema é formulado como um problema de cobertura de conjuntos. Para resolver o problema, o conjunto  $F$  é transformado na matriz binária  $A_{i,j}$ , cujas linhas representam as viagens diárias da empresa e as colunas as linhas de trabalho de  $F$ . Assim,  $a_{ij} = 1$  se a viagem diária equivalente a  $i$  está contida na linha de trabalho equivalente a  $j$ . Caso contrário,  $a_{ij} = 0$  a viagem diária  $i$  não está presente na linha de trabalho  $j$ . Portanto, o problema é formulado como:

$$\begin{aligned} \min z &= \sum_{j=1}^{\text{card}(A)} c_j x_j, \\ \text{sujeito a:} \\ r_i &= \sum_{j=1}^{\text{card}(A)} a_{ij} x_j \geq 1, \text{ para } i = 1, \dots, P \cdot N, \\ x_j &\in \{0,1\}. \end{aligned}$$

Para resolver este problema desenvolveu-se um procedimento baseado no algoritmo genético proposto por Beasley e Chu (1996). Nele, a solução  $S \subseteq F$  será representada por um indivíduo  $l$  binário de comprimento igual à  $card(F)$  (número de colunas de  $A$ ) e cada posição (gene) de  $l$  indica se a correspondente linha de trabalho de  $F$  está ou não presente em  $S$ . Um valor 1 em uma posição de  $l$  indica que a linha de trabalho respectiva de  $F$  está presente em  $S$ ; caso o valor seja 0 a respectiva linha de trabalho de  $F$  não está presente em  $S$ . A Figura 3 ilustra um indivíduo para o caso de teste P3N3 já levando em consideração a matriz  $A$ .

		Índice das colunas de A (ou linhas de F)								
		1	2	3	4	5	6	7	8	9
Índice das viagens diárias	1	1	1	1	0	0	0	0	0	1
	2	0	0	0	1	1	0	0	1	0
	3	0	0	0	0	0	1	1	0	0
	4	0	0	1	1	0	0	0	0	1
	5	1	1	0	0	0	1	1	1	0
	6	0	0	0	0	1	0	0	0	0
	7	1	0	0	0	0	1	0	1	1
	8	0	0	1	1	1	0	0	0	0
	9	0	1	0	0	0	0	1	0	0
		<b>Indivíduo <math>l</math></b>								
		<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>

Figura 3 – Representação de um indivíduo  $l$  para o caso P3N3.

O indivíduo  $l$  mostrado na Figura 3 indica que apenas as linhas de trabalho 5, 7 e 9 de  $F$  fazem parte da solução  $S$ . Um conjunto de indivíduos  $l$  forma a população inicial  $\Pi$ . A adaptação de um indivíduo reflete o quanto a correspondente solução  $S$  é uma boa solução, no sentido da função objetivo do problema. O tamanho da população inicial é obtido mediante a fórmula  $\rho = \phi n \mu$ , onde  $\phi$  é a densidade da matriz de cobertura  $A$  (número de 1s dividido pelo número total de elementos da matriz),  $n$  é a cardinalidade do conjunto  $F$  (igual ao número de colunas de  $A$ ) e  $\mu$  é um fator redutor ajustado empiricamente.

Para gerar a população inicial, para cada viagem diária  $i$ , sorteia-se aleatoriamente uma linha de trabalho  $j$  de  $F$  que cobre esta viagem (equivalente a uma coluna  $j$  de  $A$  tal que  $a_{ij} = 1$ ) e faz-se a posição  $j$  equivalente no indivíduo  $l$  igual a 1. Uma vez consideradas todas as viagens diárias, o indivíduo correspondente representará uma solução  $S$ . Como uma linha de trabalho cobre, em geral, várias viagens diárias, ao final é necessário otimizar o processo eliminando-se linhas de trabalho que podem ser consideradas redundantes na solução, isto é, linhas de trabalho que se retiradas da solução (fazendo o correspondente gene igual a zero em  $l$ ) nenhuma viagem diária da empresa ficará descoberta.

A partir da população inicial o procedimento evolutivo executa operações de cruzamento e mutação para gerar as novas gerações até esgotar o tempo de evolução pré-fixado. Para o cruzamento, primeiramente, são sorteados aleatoriamente 4 pais. Em seguida, os 2 pais melhor adaptados (menor valor da função-objetivo) são escolhidos para o cruzamento. A operação de cruzamento consiste em combinar os genes dos pais para gerar um novo filho. Os genes iguais nos 2 pais são mantidos no filho. Os genes diferentes são copiados, primeiramente, do pai melhor adaptado, na proporção dos fatores de adaptação dos dois pais, e por fim do pai menos adaptado.

Encerrada a operação de cruzamento aplica-se o operador de mutação ao novo filho, com o objetivo de trocar aleatoriamente alguns de seus genes. Neste trabalho, adotou-se a estratégia de variar a quantidade de genes a serem modificados em função do tempo de evolução decorrido. Três parâmetros,  $m_f$ ,  $m_c$  e  $m_g$  são usados para controlar esta variação. Estes parâmetros são utilizados na função  $f(t) = \eta = \left[ m_f / 1 + e^{(-4m_g(t-m_c)/m_f)} \right]$ , onde  $\eta$  é a taxa de genes a ser modificada a cada novo filho gerado,  $t$  é o tempo de evolução decorrido, medido pelo número de filhos gerados desde o início do processamento,  $m_f$  é o valor final de variação do número de genes,  $m_c$  é o número de filhos gerados quando a variação de genes atingir metade de  $m_f$ . Finalmente,  $m_g$  é o gradiente da função  $f(t)$  quando  $t = m_c$ . O objetivo desta estratégia é tornar a mutação mais influente no final do processo quando o algoritmo estiver em fase de convergência.

Finalmente, após a aplicação das operações de cruzamento e de mutação, o novo filho é submetido a uma operação de viabilização, pois devido a estas operações o novo filho pode ter se tornado inviável caso alguma viagem diária não esteja coberta. Esta viabilização é feita usando o procedimento a seguir.

Sejam,

$U$  = um conjunto contendo as linhas de  $A$  que ficaram descobertas;

$j^*$  = a melhor coluna  $j$  de  $A$ , isto é, aquela que cobre uma linha de  $U$  com o menor valor da razão  $c_j / (\text{número de linhas de } U \text{ que serão cobertas por } j)$ ;

$w_i$  = o conjunto cujos elementos representam a quantidade de vezes que uma linha de  $U$  será coberta. Inicialmente esses elementos são zerados;

1. Para cada linha  $i$  de  $U$  em ordem crescente de  $i$ ;
  2. Troque a posição de  $l$  correspondente a  $j^*$  de 0 para 1;
  3. Faça  $w_i = w_i + 1$  para todo  $i: A[i][j^*] \neq 0$ ;
  4. Faça  $U = U - \{i\}$ ;
5. Para todo  $j \in \mathbf{1}: \mathbf{1}_j \neq \mathbf{0}$  em ordem decrescente de  $j$ ;
  6. Se  $w_i \geq 2$  para todo  $i: A[i][j] \neq 0$ ;
  7. Faça  $w_i = w_i - 1$ ;
  8. Troque a posição de  $l$  correspondente a  $j$  de 0 para 1;

Uma vez obtido um filho viável, este será inserido na população atual, caso ainda não pertença à população. Caso já exista, o novo filho será descartado. Caso contrário, o novo filho entrará na população no lugar de um indivíduo cujo valor de adaptação seja imediatamente superior ao seu. Os procedimentos de cruzamento, mutação e viabilização são repetidos até que o tempo de evolução pré-fixado seja atingido. Quando isto ocorrer, a população final é percorrida à procura do filho com o maior valor de adaptação (menor valor de custo). Este indivíduo é otimizado através de um procedimento heurístico para se obter o número de 1s em  $l$  igual ao número de viagens diárias da empresa ( $N$ ) e com isto satisfazer a restrição  $r_5$ . Como o procedimento heurístico não garante que isto ocorra, este problema será tratado posteriormente quando da associação de linhas de trabalho a padrões de folga.

### 3.3 O subproblema dos padrões de folga

Um padrão de folga  $n \times m$  consiste de uma sequência de  $n$  dias de expediente intercalados por  $m$  dias de folga. A Figura 1 ilustra quatro possíveis variações para um padrão de folga  $3 \times 1$ , em que os círculos preenchidos representam dias de expediente e círculos não preenchidos, dias de folga.

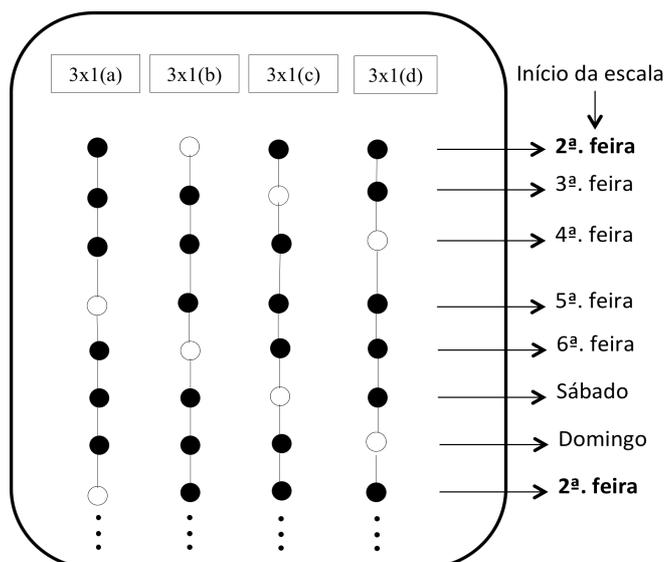


Figura 1 – Quatro variações do padrão de folga 3 x 1.

Neste trabalho, o problema dos padrões de folga é formulado como um problema de particionamento de conjuntos, da seguinte forma:

$$\begin{aligned} \min z &= \sum_{j=1}^M x_j, \\ \text{sujeito a:} \\ \sum_{j=1}^M a_{ij} x_j &= N, \text{ para } i = 1, \dots, P \\ x_j &\geq 0, \\ x_j &\in Z^*. \end{aligned}$$

onde  $i$  representa o número de dias da programação,  $j$  representa cada um dos  $M$  tipos de padrão de folga com os quais a empresa trabalha,  $x_j$  é o número de motoristas a serem alocados em cada padrão de folga  $j$ ,  $a_{ij}$  são os coeficientes da matriz de cobertura do problema, tal que  $a_{ij} = 1$  se  $i$  é dia de expediente no padrão de folga  $j$ , e  $a_{ij} = 0$  caso contrário (ou seja,  $i$  é dia de folga no padrão de folga  $j$ ). As restrições especificam que deve haver exatamente  $N$  motoristas disponíveis diariamente para atender as  $N$  viagens diárias da empresa. O objetivo é minimizar o número total de motoristas alocados para cada padrão de folga.

Este problema pode ser resolvido por otimizadores comerciais, pois em geral não se usam muitos padrões de folga. A solução é um número exato de motoristas que formarão um *pool* e estarão disponíveis todos os dias da escala para serem alocados a cada uma das linhas de trabalho obtidas na solução do subproblema das linhas de trabalho.

#### 4. Finalização da heurística

Na determinação do conjunto  $S$ , devido à operação de otimização realizada no melhor indivíduo  $l$  obtido no fim da evolução, a correspondente solução pode ser tal que  $card(S) = N$  e as restrições  $r_4$  e  $r_5$  são respeitadas. Por outro lado, na resolução do subproblema de padrões de folga a soma de expedientes a cada dia, considerando todos os padrões de folga, é igual a  $N$ . Portanto neste caso, pode-se associar diretamente as linhas de trabalho encontradas aos padrões de folga, ou seja, um padrão de folga associado a uma linha de trabalho e vice-versa. A Figura 4 ilustra esta associação para o caso P3N3. Esta é a melhor solução do ponto de vista de redução do número de motoristas ao mínimo necessário.

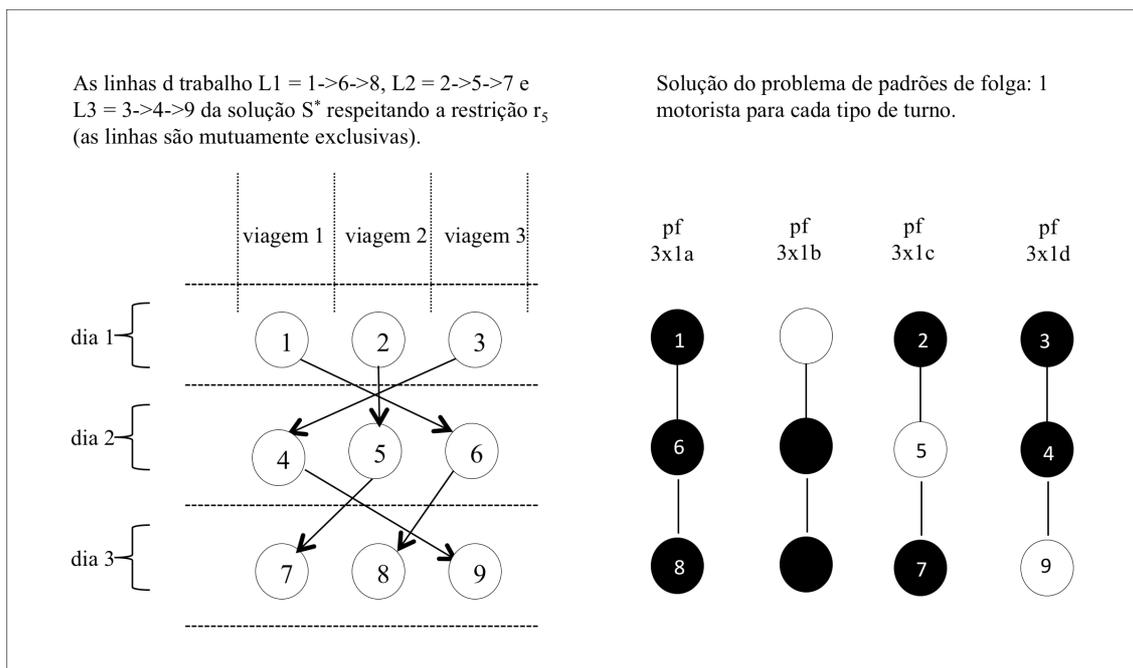


Figura 4 – A associação de linhas de trabalho aos padrões de folga.

Observe nesta figura que o grafo do lado esquerdo mostra uma solução  $S^*$  com três linhas de trabalho, compostas por três viagens diárias cada uma, respeitando a restrição  $r_5$ . São elas:

Linha 1 -  $L_1 = \{(1, 6), (6, 8)\}$

Linha 2 -  $L_2 = \{(2, 5), (5, 7)\}$

Linha 3 -  $L_3 = \{(3, 4), (4, 9)\}$

No lado direito da figura é mostrada a solução do caso P3N3 para o subproblema dos padrões de folga. Esta solução usa apenas três dos quatro padrões disponíveis, ou seja, dos padrões  $3x1a$ ,  $3x1b$ ,  $3x1c$  e  $3x1d$  são necessários apenas os padrões:  $3x1a$ ,  $3x1c$  e  $3x1d$  para atender ao problema de particionamento de conjuntos formulado. Isto resulta em um total de três expedientes por dia, ou seja, o primeiro dia é dia de expediente (círculo preto) nos padrões 'a', 'c' e 'd'. No segundo dia, existe um dia de folga no padrão 'c' (círculo branco) e no terceiro dia existe um dia de folga no padrão 'd'. Como cada padrão de folga é atribuído a um motorista, tem-se uma demanda de três motoristas por dia para cobrir a demanda de viagens diárias que também é de três viagens. A associação de linhas de trabalho aos padrões de folga está mostrada na Figura 4, com a linha  $L_1$  associada ao padrão  $3x1a$ , a linha  $L_2$  associada ao padrão  $3x1c$  e a linha  $L_3$ , associada ao padrão  $3x1d$ . A configuração das linhas e a formulação do problema de padrões com um termo independente igual a N produziu uma lacuna de 1 motorista em cada dia da programação. A coincidência de ter três motoristas disponíveis dos três padrões de folga no primeiro dia, no entanto, não se repete nem no segundo e nem no terceiro dias. No segundo o motorista do padrão 'c' está de folga e no terceiro dia o motorista do padrão 'd' está de folga. Esta configuração é tal que exige apenas 1 motorista cobre-turno para cobrir a folga dos padrões 'c' e 'd', que ocorrem em dias distintos. Este é verdadeiramente o número mínimo de motoristas para este caso P3N3, pois será necessário apenas 1 motorista cobre-turno mais os três motoristas regulares para satisfazer a programação.

A condição necessária para que esta configuração se repita é a seguinte: o melhor indivíduo obtido pelo processo de otimização deve satisfazer as restrições  $r_4$  e  $r_5$ . Caso isto não ocorra, é necessário um processamento posterior, como o sugerido na seção 6.

## 5. Testes computacionais

A heurística proposta foi escrita na linguagem C++ e os testes computacionais foram realizados em um microcomputador com processador Pentium Core 2 Duo, 1.4 Ghz, 32 bits, 2 Gb de memória RAM. Para a resolução do subproblema de padrões de folga foi empregado o otimizador Xpress, versão 7.2.

Inicialmente, foram feitos alguns testes para avaliar a geração de um conjunto de linhas de trabalho consistentes com as restrições do problema (conjunto  $F$ ), a partir das linhas geradas pelo percorrimento do grafo que representa o espaço de soluções do problema (conjunto  $X$ ). A Tabela 1 mostra os resultados destes testes preliminares.

<b>P</b>	<b>N</b>	<b>card(<math>X</math>)</b>	<b>card(<math>F</math>)</b>	<b>Tempo (s)</b>
5	3	1736741824	8225	796
5	4	12417064117	17616	1218
5	4	19717665514	38520	1695
5	4	39886354211	72147	2322

Tabela 1 – Geração dos conjuntos  $X$  e  $F$ .

Os seguintes requisitos foram analisados para a geração das linhas de trabalho do conjunto  $F$ : diversificação, cobertura total de viagens, não-redundância e atendimento aos critérios de qualidade: duração, dispersão e tamanho.

Conforme pode-se observar na Tabela 1, o algoritmo de percorrimento produz uma quantidade expressiva de linhas de trabalho, sem produzir linhas redundantes. Ao serem produzidas, as linhas de trabalho passam por um processo de verificação quanto aos critérios de qualidade, ou seja, duração total da linha dentro dos limites especificados, dispersão em relação à duração de cada viagem de uma mesma linha dentro do limite especificado e tamanho em termos de número de viagens diárias de cada linha igual a  $P$  (número de dias de programação das escalas).

Em seguida, foram realizados testes para avaliação do algoritmo genético (AG) proposto, considerando 100000 gerações. Foram utilizados exemplares de teste para o problema de cobertura de conjuntos (PCC) disponíveis na biblioteca OR-Library (BEASLEY, 1996). Foram considerados os exemplares 4.1 a 4.10, 5.1 a 5.10, 6.1 a 6.5, A.1 a A.5, B1 e B2, C.1 a C.5 e D.3, num total de 38 exemplares. Cada exemplar de teste foi resolvido 8 vezes. A Tabela 2 mostra os melhores resultados obtidos nestes testes somente para os exemplares em que o algoritmo proposto não alcançou a melhor solução conhecida ( $\text{Gap} > 0$ ). Nesta tabela são mostrados, a identificação do exemplar, a melhor solução conhecida (MSC), a solução obtida pelo algoritmo proposto (Sol), e o desvio percentual da solução obtida em relação à melhor solução conhecida (Gap), calculado como:  $\text{Gap} = 100 * (\text{Sol} - \text{MSC}) / \text{MSC}$ .

<b>Exemplar</b>	<b>MSC</b>	<b>Sol</b>	<b>Gap</b>
4.7	430	432	0,47
4.9	641	644	0,47
C.3	243	246	1,23
<b>Média</b>			<b>0,06</b>

Tabela 2 – Resultados da heurística proposta para solução do PCC

O valor do desvio médio de 0,06%, mostrado na Tabela 2, considera os 35 outros exemplares com  $\text{Gap} = 0$ . A Figura 5 mostra a curva de convergência do algoritmo AG proposto para o exemplar de teste 4.1.

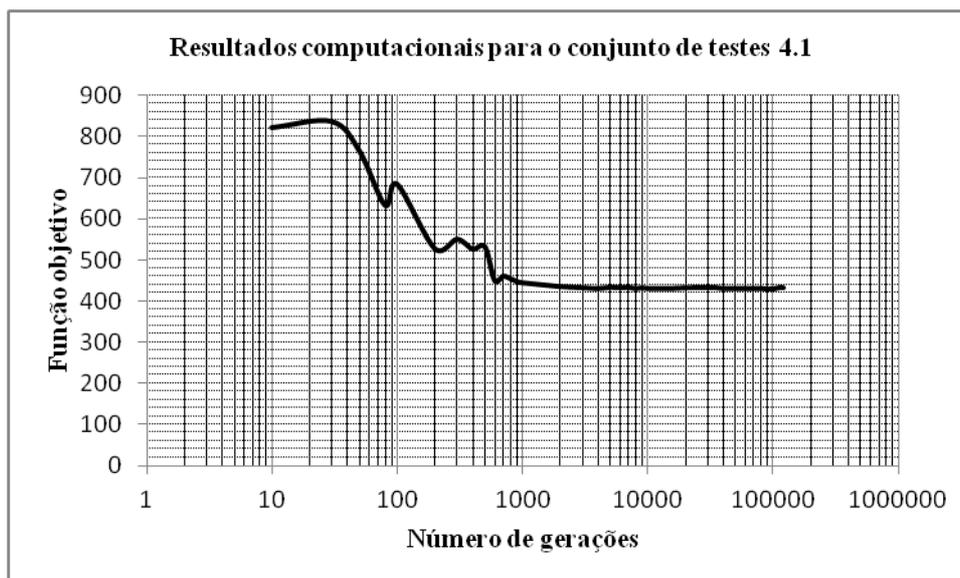


Figura 5 – Convergência da heurística proposta para solução do PCC

A Tabela 3 mostra algumas configurações dos parâmetros (definidos na seção 3.2) do método proposto na seção 4 para a criação de linhas de trabalho, para o caso P10N6, ou seja, programação de 10 dias ( $P = 10$ ) com 6 viagens por dia ( $N = 6$ ). Nesta tabela são mostrados os valores dos parâmetros ( $mf$ ,  $mc$  e  $mg$ ), o valor da função-objetivo (FO), o número de linhas (NL) e o tempo de execução do método proposto (em segundos). Para fins de comparação, este caso foi resolvido até a otimalidade pelo *solver* XPress, resultando em FO = 1800 e NL = 7. A Tabela 3 mostra apenas as melhores configurações dos parâmetros que levaram o método proposto a uma solução com o mesmo valor da função-objetivo da solução ótima obtida pelo XPress.

$mf$	$mc$	$mg$	FO	NL	Tempo (s)
<b>10</b>	<b>200</b>	<b>2,0</b>	<b>1800</b>	<b>7</b>	<b>120</b>
<b>10</b>	<b>100</b>	<b>2,0</b>	<b>1800</b>	<b>7</b>	<b>120</b>
<b>10</b>	<b>200</b>	<b>1,3</b>	<b>1800</b>	<b>6</b>	<b>113</b>

Tabela 3 – Configurações de parâmetros para o caso P10N6

Deve-se observar que o método proposto, para a configuração dos parâmetros [ $mf = 10$ ,  $mc = 200$  e  $mg = 1,3$ ] é capaz de encontrar uma solução sem linhas redundantes (NL = 6), uma vez que a programação considera 6 viagens por dia. Neste teste foi empregado o conjunto  $F$  obtido pelo algoritmo de percorrimento para o caso em questão, o qual produziu 423 linhas de trabalho. Neste caso, conforme observado anteriormente, como as restrições  $r_4$  e  $r_5$  foram obedecidas (pois não existem linhas de trabalho redundantes), existe possibilidade de se encontrar facilmente uma solução para o subproblema dos padrões de folga de modo a minimizar o número de motoristas.

## 6. Conclusão

Este trabalho propõe um novo método heurístico para resolver um problema considerado de difícil solução: o problema do escalonamento de pessoal para empresas de transporte rodoviário, contendo vários objetivos e restrições operacionais e trabalhistas. O método proposto divide o problema em três subproblemas: construção do espaço de solução; obtenção de um bom conjunto de linhas de trabalho; determinação de um conjunto adequado de padrões de folga para o conjunto de linhas de trabalho de modo a minimizar o número necessário

de motoristas. A estrutura de solução para o método proposto ainda não está completamente estabelecida, mas os resultados obtidos para os dois primeiros subproblemas são positivos e incentivam a continuação do algoritmo proposto. Até o momento, a contribuição desta proposta está na representação do problema por meio de um grafo, no algoritmo de percorrimento deste grafo para obtenção de boas linhas de trabalho e no algoritmo genético para minimizar o número de linhas de trabalho necessárias. Não se observa na literatura relacionada nenhum outro trabalho com o mesmo enfoque.

O grande desafio ainda por fazer é encontrar o número mínimo de motoristas para atribuição das escalas de serviço programadas, uma vez que quando na solução da heurística evolutiva proposta o número final de linhas de trabalho é igual ao número de expedientes do subproblema de padrões de folga, o número de motoristas é o mínimo desejado. Porém, quando o conjunto de linhas de trabalho obtido no final do processo evolutivo contém algumas viagens diárias cobertas por mais de uma linha de trabalho, a restrição  $r_5$  não estará sendo atendida. Neste caso, é preciso eliminar esta redundância e ao mesmo tempo manter o número de motoristas ao mínimo possível. Este é um problema difícil e uma heurísticas de solução encontra-se em fase de construção. Espera-se que esta heurística possa levar a boas soluções, embora o número final de motoristas possa ser maior do que o número de motoristas regulares da empresa.

### Referências

- BEASLEY, J.F. (1990). OR-Library: Distributing test problems by electronic mail. *Journal of the Operations Research Society*, v. 41, n. 11, p. 1069-1072.
- BEASLEY, J.F.; CHU, P.C. (1996). A genetic algorithm for the set covering problem. *European Journal of Operational Research*, v. 94, n. 2, p. 392-404.
- BURKE, E.K.; KENDALL, G.; SOUBEIGA, E. (2003). A tabu-search hyperheuristic for timetabling and rostering. *Journal of Heuristics*, v. 9, n. 6, p. 451-470.
- CAPRARA, A.; FISCHETTI, M.; TOTH, P.; VIGO, D.; GUIDA, P.L. (1997). Algorithms for railway crew management. *Mathematical Programming*, v. 79, n. 1-3, p. 125-141.
- ERNST, A.T.; JIANG, H.; KRISHNAMOORTHY, M.; OWENS, B.; SIER, D. (2004). An Annotated Bibliography of Personnel Scheduling and Rostering. *Annals of Operations Research*, v. 127, n. 1-4, p. 21-144.
- GOODALE, J.C.; THOMPSON, G.M. (2004). A Comparison of Heuristics for Assigning Individual Employees to Labor Tour Scheduling. *Annals of Operations Research*, v. 128, n. 1-4, p. 47-63.
- LOURENÇO, H.R.; PAIXÃO, J.P.; PORTUGAL, R. (2001). Multiobjective Metaheuristics for the Bus-Driver Scheduling Problem. *Transportation Science*, v. 35, n. 3, p. 331-343.
- MASON, A.; NIELSEN, D. (1999). *PETRA: A programmable optimisation engine and toolbox for personnel rostering applications*. University of Auckland, New Zealand. Presented at the 15th Triennial International Federation of Operational Research Societies Conference, IFORS 99.
- MEISELS, A.; SCHAERF, A. (2003). Modelling and Solving Employee Timetabling Problems. *Annals of Mathematics and Artificial Intelligence*, v. 39, n. 1-2, p. 41-59.
- MESQUITA, M.; MOZ, M.; PAIAS, A.A. (2011). A new model for the integrated vehicle-crew rostering. *Journal of Scheduling*, v. 14, n. 4, p. 319-334.
- RESPICIO, A.; MOZ, M.; PATO, M.V. (2013). Enhanced genetic algorithms for a bi-objective bus driver rostering problem: a computational study. *International Transactions in Operational Research*, v. 20, n. 4, p. 443-470.



SHODI, M.; NORRIS, S. (2004). A flexible, fast, and optimal modeling approach applied to crew rostering at London Underground. *Annals of Operations Research*, v. 127, n. 1-4, p. 259-281.

VANHOUCKE, M.; MAENHOUT, B.. (2010). A hybrid scatter search heuristic for personalized crew. *European Journal of Operational Research*, v. 206, n. 1, p. 155-167.