

MÉTODO INTERATIVO PARA RESOLUÇÃO DO PROBLEMA DE ESCALONAMENTO DE TRIPULAÇÕES

Geraldo Regis Mauri

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

Luiz Antonio Nogueira Lorena

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

Resumo

Este trabalho descreve uma metodologia interativa baseada na aplicação do Algoritmo de Treinamento Populacional (ATP) juntamente com programação linear (PL) para a geração de escalas para tripulações de um sistema de transporte coletivo. Estes métodos são aplicados de maneira interativa, onde o ATP, através de informações da relaxação da PL, é responsável pela geração de boas colunas (baixo custo e boa cobertura das tarefas), e a PL pela resolução de um problema de particionamento de conjuntos formado por essas colunas. Os resultados obtidos são comparados com os da metaheurística *Simulated Annealing*, usando instâncias geradas aleatoriamente, porém baseadas em problemas reais.

Palavras-chave: Escalonamento de Tripulações, Algoritmo de Treinamento Populacional, Geração de Colunas.

Abstract

This paper describes an interactive methodology based on the application of the Population Training Algorithm (PTA) together with linear programming (LP) for the generation of scales for crews of a public transportation system. These methods are applied in an interactive way, where PTA, through information of the relaxation of LP, is responsible for the generation of good columns (low cost and good covering of the tasks), and LP for the resolution of a set partitioning problem formed by these columns. The computational results are compared against a Simulated Annealing metaheuristic, using randomly generated instances based on real problems.

Keywords: Crew scheduling, Population Training Algorithm, Column Generation.

1. Introdução

Desde a década de 60, vários autores vêm se dedicando ao desenvolvimento de modelos capazes de gerar escalas de tripulações que satisfaçam às restrições impostas, e que possuam o menor custo possível. Este é um problema permanentemente estudado uma vez que as realidades dos sistemas de transporte estão em contínua transformação e exigem cada vez mais uma gerência eficiente dos recursos disponíveis. Prova disso é a existência de congressos especializados sobre o tema nas últimas décadas, como por exemplo, a *International Conference on Computer-Aided Scheduling of Public Transport* (Daduna et al., 1995; Desrochers e Rousseau, 1992; Wilson, 1999; Vof e Daduna, 2001; Shen e Kwan, 2001).

Entretanto, pouco tem sido feito no Brasil, talvez pelo fato de que aqui além do grande número de restrições determinadas por lei, algumas empresas ainda impõem suas próprias regras, baseadas em políticas de trabalho adotadas, tornando esse problema mais complexo do que na maioria dos países desenvolvidos.

Além disso, a democratização do acesso ao sistema de transporte público no Brasil depende da garantia de tarifas suportáveis para os usuários, e esta depende tanto do aumento da eficiência e da redução dos custos operacionais, quanto da concessão de subsídios em situações específicas, sob controle social (ANTP, 2004).

E, como é sabido que nas empresas de transporte público urbano a mão-de-obra operacional representa um dos maiores custos (Bouzada, 2002), uma pequena redução neste item pode significar um ganho considerável no custo total, podendo reduzir significativamente o valor das tarifas para o usuário final, justificando assim qualquer trabalho no sentido de minimizar os custos com a mão-de-obra.

Várias metodologias adotadas para resolver o problema de escalonamento de tripulações (PET) são encontradas na literatura. O problema é conhecido por ser *NP-difícil*. Metaheurísticas como *Algoritmos Genéticos*, *Busca Tabu*, *Simulated Annealing*, entre outras, permitem incluir vários tipos de condições de trabalho facilmente (Shen e Kwan, 2001; Lourenço et al., 2001; Mauri, 2003). Técnicas de Geração de Colunas também têm mostrado resultados muito bons (Desrochers e Soumis, 1989; Wilhelm, 2001).

Este trabalho apresenta uma nova alternativa para gerar boas escalas para tripulações. O método evolutivo chamado Algoritmo de Treinamento Populacional (ATP) é utilizado para gerar colunas de alta qualidade para formar um problema de particionamento de conjuntos, que é resolvido através da programação linear (PL). O artigo está organizado como segue. A seção 2 apresenta a definição e a modelagem do problema. O ATP é detalhado na seção 3, enquanto a seção 4 descreve o processo de geração de colunas e a interação ATP/PL. Os resultados computacionais obtidos são apresentados na seção 5, e as conclusões são resumidas na seção 6.

2. Descrição do Problema

O Problema de Escalonamento de Tripulações - PET, também conhecido nos países europeus como *Crew Scheduling Problem – CSP* (Wren e Rousseau, 1993) ou *Bus Driver Scheduling Problem – BDS* (Lourenço et al., 2001), consiste na atribuição da tarefa de condução dos veículos aos motoristas e cobradores (tripulações), de tal forma que as viagens das diferentes linhas atendidas pela empresa sejam executadas com o menor custo possível.

No transporte coletivo rodoviário, usualmente o escalonamento de tripulações é feito após a programação dos veículos. Nesta, as viagens são reunidas em *Blocos* (ver Figura 1), onde um bloco apresenta a seqüência de viagens distribuídas sucessivamente a um ônibus começando e terminando na garagem.

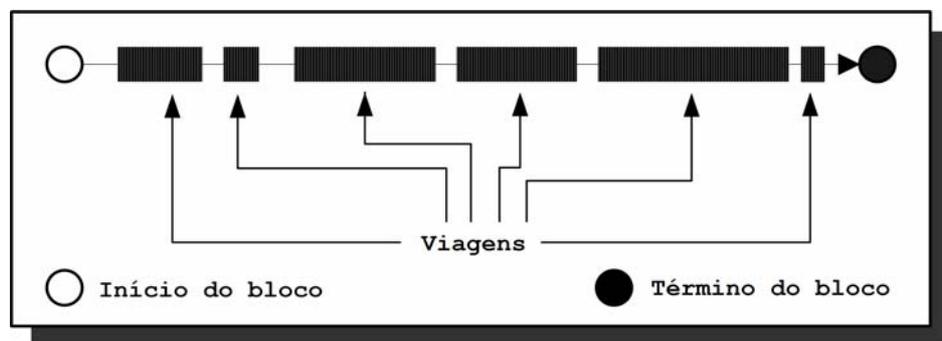


Figura 1 - Representação do bloco de um veículo.

Cada bloco mostra também as *Oportunidades de Troca - OT's* (*Relief Opportunities* segundo Lourenço et al. (2001) ou *Relief Points* segundo Desrochers e Soumis (1989)), que são intervalos de

tempo suficiente para haver a troca das tripulações (ver Figura 2). A partir das viagens dos blocos dos veículos são formadas as *Tarefas* (ver Figura 2), a partir das quais são criados os *Turnos* (ver Figura 3), que por sua vez são atribuídos às tripulações, as quais irão executá-los durante um dia de trabalho.

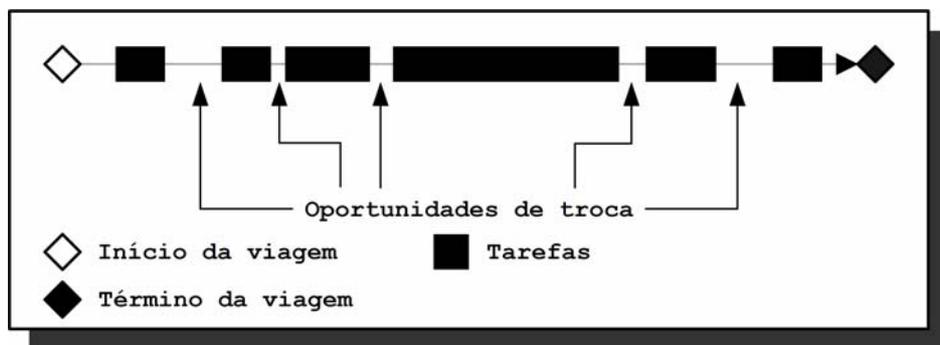


Figura 2 - Representação de uma viagem.

O processo de formação desses turnos está sujeito a um conjunto de regras que são específicas a uma organização. Estas regras são geralmente derivadas de regras nacionais e locais, podendo ser obrigatórias ou não. Tipicamente, há restrições no tempo total trabalhado, na extensão total do turno (duração entre o início e o fim do turno) etc.

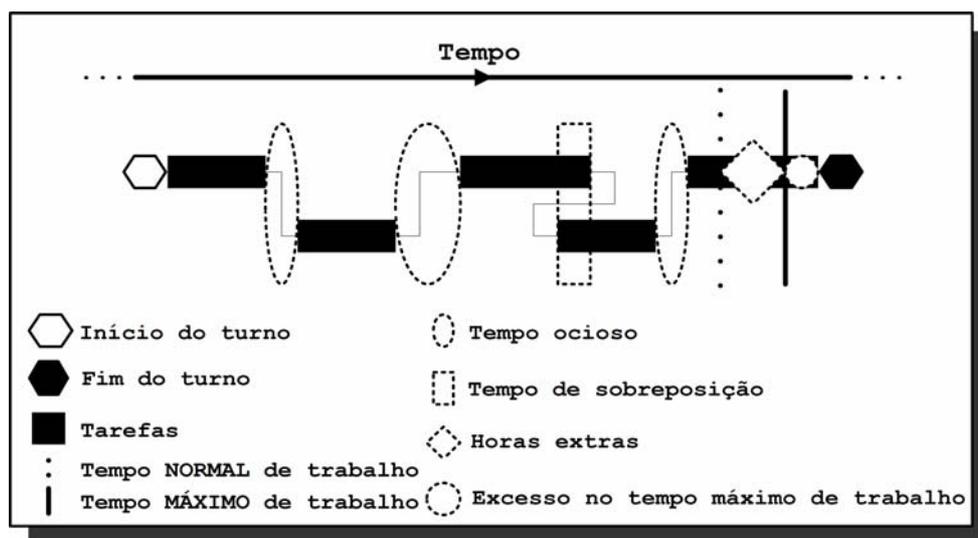


Figura 3 - Representação de um turno.

Enfim, este processo de atribuir tarefas às tripulações, também chamado de escalonamento de motoristas, é a construção de um conjunto de turnos legais que cubram todos os blocos do horário de um veículo em particular, o que em uma escala para vários veículos pode refletir toda a operação de uma organização, assim como uma parte específica contida nesta operação (Wren e Rousseau, 1993).

A modelagem abordada neste trabalho toma como base as entidades propostas por Mauri (2003) devido à sua identificação com outras encontradas na literatura, à sua clareza e ao resultado obtido em seu trabalho para problemas reais. Essa modelagem trata o problema através das entidades: *Tarefas*, *Tripulações* e *Turnos* (*Jornadas diárias* segundo ele).

Assim, o problema é descrito como o de formação de uma matriz, onde as colunas representam as tripulações, e as linhas as tarefas (ver Figura 4). Cada elemento $A_{ij} \in \{0,1\}$, sendo $i \in M = \{1..m\}$ e $j \in N = \{1..n\}$, m é o número de tarefas (linhas), e n o número de tripulações (colunas), onde $a_{ij} = 1$ se a tarefa i pertence ao turno da tripulação j , e 0 caso contrário.

$$A = \begin{bmatrix} a_{11} & a_{12} & \rightarrow & a_{1n} \\ a_{21} & a_{22} & \rightarrow & a_{2n} \\ \downarrow & \downarrow & \rightarrow & \downarrow \\ \vdots & \vdots & \rightarrow & \vdots \\ a_{m1} & a_{m2} & \rightarrow & a_{mn} \end{bmatrix}$$

Figura 4 - Representação do problema.

Essa matriz é usada para resolver o seguinte problema de particionamento de conjuntos (PPC):

Minimizar:

$$\sum_{j=1}^n c_j x_j \quad (1)$$

Sujeito a:

$$\sum_{j=1}^n a_{ij} x_j = 1 \quad i = 1, \dots, m \quad (2)$$

$$x_j \in \{0, 1\}; \quad j = 1, \dots, n \quad (3)$$

Onde, c_j representa o custo da coluna j e x_j é igual a 1 se a coluna (turno) j pertencer a solução do problema e 0 caso contrário. Esta é uma formulação clássica e constantemente utilizada em diversos trabalhos encontrados na literatura (Desrochers e Soumis, 1989; Wilhelm, 2001).

Os custos atribuídos às colunas, considerados neste trabalho, penalizam horas extras, tempo ocioso, tempos de sobreposição e excesso no tempo trabalhado. Esses custos são mostrados na Figura 3, onde *Tempo NORMAL de trabalho* refere-se ao tempo “normal” de duração de um turno (sem horas extras), *Tempo MÁXIMO de trabalho* refere-se ao tempo máximo de duração de um turno (com horas extras), *Tempo ocioso* indica o tempo em que uma tripulação não está trabalhando durante seu turno, *Tempo de sobreposição* indica o tempo em que uma tripulação está, teoricamente, realizando duas tarefas ao mesmo tempo, *Horas extras* indica o tempo em que o turno de uma tripulação excede o tempo normal de trabalho, e *Excesso no tempo total de trabalho* refere-se ao tempo em que um turno excede o tempo máximo de trabalho.

Para as companhias do setor, é essencial (obrigatório) evitar excesso no tempo trabalhado e sobreposições, mas o tempo ocioso e as horas extras são toleráveis (não essenciais). Colunas com custo zero são as melhores, mas podem não representar um bom turno. O PPC requer que todos os custos sejam minimizados enquanto todas as tarefas sejam executadas.

A partir desses custos, os turnos são avaliados de acordo com a seguinte expressão:

$$c_j = ET_j + TS_j + HE_j + TO_j \quad (4)$$

onde

$$ET_j = \max(0, [HT(t_{ult}) - HI(t_{pri})] - TMT)$$

$$TS_j = \sum_{i=1}^{p-1} \max(0, [HT(t_i) - HI(t_{i+1})])$$

$$HE_j = \max(0, [HT(t_{ult}) - HI(t_{pri})] - TNT)$$

$$TO_j = \max(0, TNT - [HT(t_{ult}) - HI(t_{pri})]) + \sum_{i=1}^{p-1} \max(0, [HI(t_{i+1}) - HT(t_i)])$$

Onde, c_j é o custo da tripulação j ; ET_j é o excesso no tempo máximo de trabalho da tripulação j ; TS_j é o tempo de sobreposição da tripulação j (tempo em que a tripulação está, teoricamente, realizando duas tarefas ao mesmo tempo); HE_j é o tempo que excede o tempo normal de trabalho da tripulação j (horas extras); TO_j é o tempo ocioso da tripulação j (tempo em que a tripulação não está trabalhando durante seu turno); TMT é o tempo máximo de trabalho (8hs + 2hs); TNT é o tempo normal de trabalho (8hs); p é o número de tarefas do turno da tripulação j ; $HT(t_i)$ é o horário de término da tarefa i ; $HI(t_i)$ é o horário de início da tarefa i ; t_{pri} é a primeira tarefa do turno da tripulação j ; t_{ult} é a última tarefa do turno da tripulação j .

3. Algoritmo de Treinamento Populacional

O Algoritmo de Treinamento Populacional (ATP) é um tipo de técnica evolutiva empregado inicialmente por Oliveira (2002), e derivada do Algoritmo Genético Construtivo (AGC). O AGC foi proposto por Lorena e Furtado (2001) e possui várias características inovadoras comparadas aos algoritmos genéticos tradicionais. Uma dessas características é a utilização de uma população “ranqueada” de tamanho dinâmico composta de “esquemas” e “estruturas”. Os esquemas e as estruturas são avaliados diretamente em uma base comum, usando um processo duplo de aptidão, chamado *aptidão-fg*.

Os esquemas não são usados no ATP, e a *aptidão-fg* é executada através de heurísticas. Um indivíduo é considerado bem adaptado se ele não melhorar considerando a heurística de treinamento utilizada. A adaptação no treinamento da população é, então, usada para guiar a busca para áreas promissoras.

Dessa forma, a representação de um turno é dada por: $S_k = (a_{1k}, a_{2k}, \dots, a_{mk})$, ou, por exemplo, $S_k = (1, 0, 1, 1, \dots, 0)$, significando que as tarefas 1, 3, 4, ..., fazem parte do turno da tripulação k .

As duas funções usadas no treinamento evolutivo são definidas através de $g(S_k)$ = “qualidade” do turno k (usado durante a geração de colunas - ver expressão (7)), e $f(S_k)$ = Melhor $g(S'_k) \mid S'_k \in \text{Vizinhança}(S_k)$ (a heurística de treinamento é detalhada na Figura 3).

O processo evolutivo é desenvolvido privilegiando os indivíduos de menor diferença $[g(S_k) - f(S_k)]$ e menor $g(S_k)$ (custo), atribuindo a eles os seguintes *ranks*:

$$\delta(S_k) = d \times [g_{\max} - g(S_k)] - [g(S_k) - f(S_k)] \quad (5)$$

onde g_{\max} é um turno aleatório de custo alto e d uma percentagem constante de g_{\max} . A população é controlada dinamicamente por um parâmetro de evolução, denominado α , que é atualizado por

$$\alpha = \alpha + \text{Step} \times PS \times \frac{\delta_{bst} - \delta_{wst}}{RG} \quad (6)$$

onde Step é uma constante que controla a velocidade do processo evolutivo, PS é o tamanho da população corrente, $(\delta_{bst} - \delta_{wst})$ é a variação, no momento, entre os *ranks* do melhor e do pior indivíduo, respectivamente, e RG é o número de gerações que faltam para terminar o processo.

O parâmetro α é comparado aos *ranks* (ver expressão (5)), e se $\alpha \geq \delta(S_k)$ então o turno S_k é eliminado da população. A população no momento de evolução α é dinâmica em tamanho e pode ser esvaziada durante o processo.

3.1. Operadores do ATP

Para a aplicação do ATP na geração de colunas, é definida como heurística direcionadora, para o cálculo da função f , uma busca local simples, onde são avaliados vários (tamanho da vizinhança definido como parâmetro do ATP) indivíduos alternativos em uma vizinhança. Essa busca é mostrada na Figura 5.

```

1.  seja  $S_k$  um turno qualquer;
2.   $f(S_k) \leftarrow g(S_k)$ ;
3.   $S'_k \leftarrow \text{clone}(S_k)$ ;
4.  por (tamanho_da_vizinhança vezes)
5.      $i \leftarrow \text{selecionar}$  (uma tarefa de  $S'_k$ );
6.      $j \leftarrow \text{selecionar}$  (uma tarefa qualquer);
7.     remover (a tarefa  $i$  de  $S'_k$ );
8.     adicionar (a tarefa  $j$  a  $S'_k$ );
9.     calcular ( $g(S'_k)$ );
10.   se ( $g(S'_k) < f(S_k)$ )
11.      $f(S_k) \leftarrow g(S'_k)$ ;
12.   fim-se
13. fim-por

```

Figura 5 - Heurística de treinamento.

A mutação também é baseada em uma busca local, implementada de uma maneira bem simplificada, na qual uma tarefa do turno de uma determinada tripulação é selecionada aleatoriamente, e trocada por outra, também selecionada aleatoriamente. Esse processo é repetido enquanto o novo turno for inválido, isto é, não atender as restrições essenciais. Esse processo pode ser visto na Figura 6.

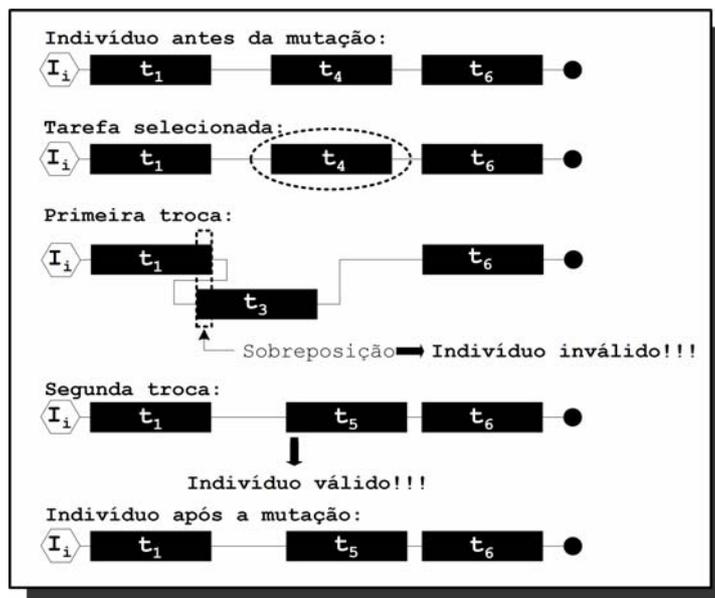


Figura 6 - Mutação.

Já o cruzamento, é responsável por gerar novos indivíduos (sempre válidos) da seguinte maneira: dois indivíduos são selecionados (base e guia) e seus cromossomos são fundidos. Como a população é ordenada de maneira decrescente através dos valores dos *ranks* (como na expressão (5)) dos indivíduos, um indivíduo base é selecionado da primeira parte da população, enquanto o guia será selecionado de toda ela. Feito isso, para garantir que o indivíduo filho seja válido, uma posição do cromossomo fundido é selecionada (aleatoriamente) e as tarefas posicionadas antes dessa posição são removidas, e as tarefas subsequentes são inseridas de uma forma seqüencial enquanto o novo indivíduo representar um turno válido. Esse processo é descrito na Figura 7.

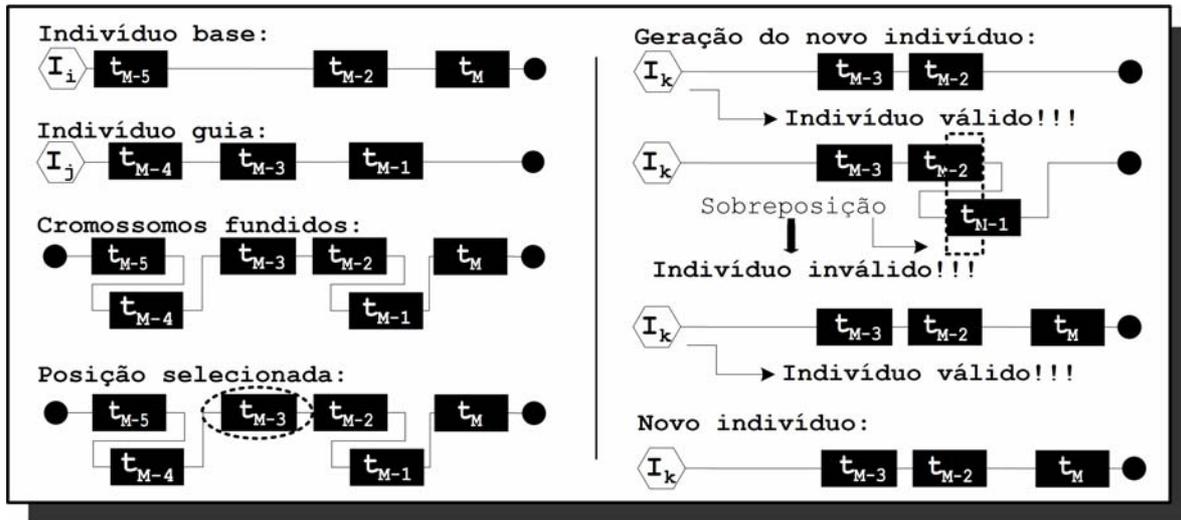


Figura 7 - Cruzamento.

Dessa forma, esses operadores são incorporados ao ATP, cujo pseudocódigo é mostrado na Figura 8.

```

1.  inicializar (P, alpha, t);
2.  enquanto (o critério de parada não for satisfeito)
3.      t ← t+1;
4.      selecionar (P, base, percentual base);
5.      selecionar (P, guia, toda a população);
6.      novo ← recombinar (base, guia);
7.      avaliar (novo);
8.      se (rand() < probabilidade de mutação)
9.          mutação (novo, tamanho da vizinhança);
10.     fim-se
11.     avaliar (vizinhança (novo));
12.     calcular (delta (novo));
13.     se (delta (novo) > alpha)
14.         atualizar (população (P, novo));
15.     fim-se
16.     alpha ← alpha + incrementar (Step);
17.     enquanto (existir delta (S) < alpha)
18.         eliminar (S);
19.     fim-enquanto
20. fim-enquanto

```

Figura 8 - Algoritmo ATP.

É interessante notar que, utilizando esses processos, o ATP formará populações de vários tamanhos, guiados pelo objetivo de selecionar colunas (tripulações) de baixo custo e com uma cobertura suficiente das tarefas. As melhores colunas deverão incluir um número de tarefas diversificado, caracterizado pela heurística de treinamento (função de treinamento) que guiará o processo evolutivo.

4. Interação ATP/PL

Resolver o problema de particionamento de conjuntos (expressões (1)-(3)) pode ser um desafio. Assim, a relaxação de PL é utilizada juntamente com o ATP para gerar um conjunto de colunas mais “tratável” por softwares comerciais (Ilog, 2001). O algoritmo descrito na Figura 9 resume a interação ATP/PL.

```

1. criar (população direcionada ao problema);
2. resolver (PL);
3. enquanto (houver melhora na PL)
4.     executar (ATP);
5.     remover (colunas inválidas);
6.     calcular (custo reduzido das novas colunas);
7.     se (PL não estabilizar)
8.         adicionar (colunas com custo reduzido negativo);
9.     senão
10.        adicionar (todas as colunas geradas);
11.     fim-se
12. resolver (PL);
13. fim-enquanto;
14. converter (PL para PLI);
15. resolver (PLI);

```

Figura 9 - Algoritmo ATP/PL.

Nesse algoritmo, uma população inicial é gerada aleatoriamente contendo apenas colunas válidas (colunas que atendam as restrições essenciais) e cobrindo todas as tarefas. A PL é resolvida e novas colunas são geradas através do ATP, considerando os valores das variáveis duais da PL para construir as funções de aptidão f e g . A função $g(S_k)$ é definida por

$$g(S_k) = \frac{c(S_k)}{\sum_{i=1}^m \lambda_i a_{ik}} \quad (7)$$

$$\theta_k = c(S_k) - \sum_{i=1}^m \lambda_i a_{ik} \quad (8)$$

onde

$c(S_k)$: custo real da coluna k (descrito pela expressão (4));
 θ_k : custo reduzido da coluna k ;
 λ_i : variável dual para a tarefa i .

Pode-se observar através das expressões (7) e (8) que, para custos reduzidos negativos, o valor da função g estará no intervalo $[0,1]$. Então, a heurística de treinamento que define a função f correspondente (melhor g em uma vizinhança), atribuirá uma pequena diferença ($g - f$) para as colunas que possuírem custos reduzidos negativos. Dessa forma, a população estará então sendo treinada, indiretamente, para geração de indivíduos com custo reduzido negativo (melhorando as colunas para a PL), evitando assim a geração de um número excessivo de colunas, e conseqüentemente, acelerando o processo.

As colunas com custo reduzido negativo ($\theta_k < 0$) são adicionadas ao problema corrente, e o novo PPC é resolvido novamente através da PL. Esse processo é repetido por um certo número de iterações, e quando a solução da PL estabilizar (o mesmo valor objetivo for obtido por, por exemplo, cinco iterações consecutivas), ou um certo número de iterações for alcançado, o processo é interrompido.

Entretanto, o valor da solução da PL poderá estabilizar muito cedo, resultando em soluções pobres quando o PPC for resolvido. Então, para evitar que isso aconteça, quando a PL estabilizar, ao invés de adicionar à nova PL apenas as colunas com custo reduzido negativo, todas as colunas geradas pelo ATP são adicionadas. Essa correção renova a geração de colunas e o processo continua convergindo.

Finalmente, terminada essa interação, a PL é convertida em PLI e resolvida pelo software de otimização CPLEX (Ilog, 2001). Porém, o problema final ainda pode ser grande para ser resolvido de uma forma exata, então após algum tempo de execução ou um *gap* limite (porcentagem da diferença dos valores da melhor PLI e PL) a melhor solução encontrada é tomada como solução para o PET.

5. Resultados Computacionais

Para validação do método implementado, foram realizados alguns testes nos quais adotaram-se instâncias geradas aleatoriamente, porém baseadas em problemas reais de uma empresa brasileira de transporte público. O número de tarefas nessas instâncias foi: 25, 50, 100, 250 e 500. Essas instâncias foram formadas por tarefas (horários de início e fim) selecionadas aleatoriamente de um problema real.

Todos os testes foram realizados em um *laptop Toshiba A10 S127* com processador Intel Celeron® de 2.0 GHz e 256Mb de memória RAM. Toda a implementação foi desenvolvida na linguagem C++ com chamadas ao software CPLEX (ver Ilog, 2001).

Os parâmetros utilizados pelo ATP são mostrados na Tabela 1. Além disso, em todos os testes, os valores de g_{max} utilizados foram obtidos a partir da maior função g dos indivíduos gerados na população inicial. Já o valor de d , também em todos os testes, foi igual a $1/g_{max}$, e α , inicialmente, foi igual a 0.

Tabela 1 - Parâmetros do ATP.

| Número de tarefas | 25 | 50 | 100 | 250 | 500 |
|--------------------------|-----------|-----------|------------|------------|------------|
| Tamanho da pop. Inicial | 10^3 | 10^3 | 10^3 | $3*10^3$ | $6*10^3$ |
| Step | 10^{-5} | 10^{-5} | 10^{-5} | 10^{-7} | 10^{-10} |
| N. máximo de gerações | $2*10^3$ | $2*10^3$ | $2*10^3$ | $5*10^3$ | $8*10^3$ |
| Percertual base | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |
| Prob. de mutação | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Tamanho da vizinhança | 7 | 7 | 7 | 7 | 7 |

A Tabela 2 mostra o número de iterações e o número de colunas geradas pelo método ATP/PL nos testes realizados. Deve-se notar que nos últimos dois experimentos (250 e 500 tarefas) descritos na Tabela 3, o PPC foi interrompido após 150 e 6000 segundos de execução, respectivamente.

Tabela 2 - Detalhes do ATP/PL.

| Número de tarefas | 25 | 50 | 100 | 250 | 500 |
|--------------------------|-----------|-----------|------------|------------|------------|
| Número de iterações | 5 | 21 | 9 | 121 | 501 |
| Núm. de colunas geradas | 522 | 11614 | 2315 | 18825 | 125991 |

Os resultados obtidos são mostrados na Tabela 3. A metodologia ATP/PL foi capaz de gerar escalas de boa qualidade para todas as instâncias em um tempo computacional razoável. Todas as restrições essenciais foram atendidas e as não essenciais minimizadas. Os resultados são comparados com uma abordagem que utiliza a metaheurística *Simulated Annealing* (SA) aplicada a um conjunto inicial de colunas geradas aleatoriamente. As penalizações no custo das colunas foram as mesmas definidas na expressão (4). Duas soluções com o SA foram obtidas, sendo que em uma (SA-20), o SA foi executado 20 vezes para diferentes conjuntos iniciais de colunas, e a outra (SA), o SA foi executado apenas uma vez.

Tabela 3 - Resultados computacionais.

| Número de tarefas | Técnica de solução | Número de tripulações | Horas extras (min) | Tempo ocioso (min) | Custo total da solução | Tempo de execução (seg) |
|-------------------|--------------------|-----------------------|--------------------|--------------------|------------------------|-------------------------|
| 25 | ATP/PL | 12 | 15 | 2356 | 2371 | 0.30 |
| | SA | 12 | 15 | 2356 | 2371 | 1.98 |
| | SA-20 | 12 | 15 | 2356 | 2371 | ≈ 39.60 |
| 50 | ATP/PL | 20 | 0 | 2600 | 2600 | 41.64 |
| | SA | 20 | 7 | 2607 | 2014 | 95.79 |
| | SA-20 | 20 | 0 | 2600 | 2600 | ≈ 1915.80 |
| 100 | ATP/PL | 40 | 0 | 7395 | 7395 | 4.98 |
| | SA | 40 | 0 | 7395 | 7395 | 67.19 |
| | SA-20 | 40 | 0 | 7395 | 7395 | ≈ 1343.80 |
| 250 | ATP/PL | 82 | 1058 | 8186 | 9244 | 229.68 |
| | SA | 84 | 907 | 8995 | 9902 | 434.80 |
| | SA-20 | 82 | 1216 | 8344 | 9560 | ≈ 8696.00 |
| 500 | ATP/PL | 151 | 1567 | 10961 | 12528 | 6797.40 |
| | SA | 153 | 1246 | 11600 | 12846 | 12132.34 |
| | SA-20 | 153 | 1198 | 11552 | 12750 | ≈ 242646.80 |

6. Conclusões

Este trabalho descreveu uma metodologia interativa para gerar boas escalas para tripulações do setor de transporte público. O ATP, integrado com uma técnica tradicional de geração de colunas foi capaz de resolver o subproblema que gera novas colunas de uma forma implícita. Isso foi feito definindo propriamente a *aplicação-fg* com informações dos valores duais. Dessa forma, esse método, em geral, mostrou ser adequado para outros problemas para os quais a geração de colunas seja indicada. Os resultados computacionais do ATP/PL foram muito bons e obtidos em tempos razoáveis de execução, comparados à metaheurística *Simulated Annealing*.

Enfim, a partir deste trabalho tem-se um grande campo de pesquisa a ser explorado, o que pode melhorar ainda mais a eficiência do método apresentado para a resolução do PET, assim como de outros problemas correlatos.

Agradecimentos

Os autores agradecem à Fundação de Amparo à Pesquisa do Estado de São Paulo – FAPESP (processo 03/04547-2) e ao Conselho Nacional de Pesquisas – CNPq (processo 304598/2003-8) pelo apoio financeiro dado ao desenvolvimento deste trabalho.

Referências

ANTP (2004), Associação Nacional de Transporte Público. [online]. Disponível em: <<http://www.antp.org.br>>.

ILOG (2001), Ilog cplex 7.5 Reference Manual. ©Copyright 2001 by Ilog.

BOUZADA, C. F. (2002), **Análise das despesas administrativas no custo do transporte coletivo por ônibus no município de Belo Horizonte**. Dissertação de Mestrado. Escola de Governo, Fundação João Pinheiro, Belo Horizonte.

DADUNA, J. R.; BRANCO, I.; PAIXÃO, J. M. P. (1995), **Computer-Aided Transit Scheduling, Proceedings of the Sixth International Workshop on Computer-Aided Scheduling of Public Transport**. Springer-Verlag.

DESROCHERS, M.; ROUSSEAU, J. M. (1992), **Computer-Aided Transit Scheduling, Proceedings of the Fifth International Workshop on Computer-Aided Scheduling of Public Transport**. Springer-Verlag.

DESROCHERS, M.; SOUMIS, F. (1989), A Column Generation Approach to the Urban Transit Crew Scheduling Problem. **Transportation Science**, v. 23, p. 1–13.

LORENA, L. A. N.; FURTADO, J. C. (2001), Constructive genetic algorithm for clustering problems. **Evolutionary Computation**, v. 9, n. 3, p. 309–327.

LOURENÇO, H. R.; PAIXÃO, J. P.; PORTUGAL, R. (2001), Multiobjective metaheuristics for the bus-driver scheduling problem. **Transportation Science**, v. 35, p. 331–343.

MAURI, G. R. (2003), **Resolução do Problema de Programação de Tripulações de um Sistema de Transporte Público via Simulated Annealing**. Relatório Técnico 02/2003, Departamento de Ciência da Computação - Universidade Federal de Ouro Preto. Disponível em: <<http://www.lac.inpe.br/~mauri/relTec0203.pdf>>.

OLIVEIRA, A. C. M. (2002), **Treinamento Populacional em Heurísticas - Aplicações em otimização**. Proposta de tese de Doutorado. Instituto Nacional de Pesquisas Espaciais (INPE).

SHEN, Y.; KWAN, R. S. K. (2001), Tabu Search for driver scheduling. **Computer-Aided Scheduling of Public Transport**. Berlin: Springer-Verlag. p. 121–135.

VOß, S.; DADUNA, J. (2001), **Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems**. Berlin: Springer-Verlag. v. 505, 73–90 p.

WILHELM, W. E. (2001), A Technical Review of Column Generation in Integer Programming. **Optimization and Engineering**, v. 2, p. 159–200.

WILSON, N. H. M. (1999), **Computer-Aided Transit Scheduling, Proceedings of the Seventh International Workshop on Computer-Aided Scheduling of Public Transport**. Berlin: Springer-Verlag.

WREN, A.; ROUSSEAU, J. M. (1993), Bus Driver Scheduling - An Overview. **6th International Workshop on Computer Aided Scheduling of Public Transport**.