

RESOLUÇÃO DO PROBLEMA DE ALOCAÇÃO DE BERÇOS ATRAVÉS DE UMA TÉCNICA DE GERAÇÃO DE COLUNAS

Geraldo Regis Mauri*

Dep. de Engenharia Rural / Centro de Ciências Agrárias
Universidade Federal do Espírito Santo (UFES)
Alegre – ES
mauri@cca.ufes.br

Alexandre César Muniz de Oliveira

Departamento de Informática
Universidade Federal do Maranhão (UFMA)
São Luís – MA
acmo@deinf.ufma.br

Luiz Antonio Nogueira Lorena

Lab. Associado de Computação e Matemática Aplicada
Instituto Nacional de Pesquisas Espaciais (INPE)
São José dos Campos – SP
lorena@lac.inpe.br

* *Corresponding author* / autor para quem as correspondências devem ser encaminhadas

Recebido em 08/2007; aceito em 12/2009 após 1 revisão
Received August 2007; accepted December 2009 after one revision

Resumo

O Problema de Alocação de Berços (PAB) aborda a programação e a alocação de navios às áreas de atracação ao longo de um cais. Neste trabalho, o PAB é modelado como um Problema de Roteamento de Veículos com Múltiplas Garagens e Janelas de Tempo, e para resolvê-lo, é utilizado um método proposto recentemente denominado ATP/PL. Este método consiste na aplicação do Algoritmo de Treinamento Populacional (ATP) juntamente com a Programação Linear (PL) para Geração de Colunas. Estes métodos são aplicados de maneira interativa, onde o ATP, através de informações da relaxação da PL, é responsável pela geração de boas colunas, e a PL pela resolução de um Problema de Particionamento de Conjuntos, com uma restrição adicional (PPC⁺), formado por essas colunas. Os resultados computacionais são obtidos através de instâncias utilizadas em um trabalho recente sobre o problema e comparados com os resultados obtidos pelo CPLEX e por outro método encontrado na literatura.

Palavras-chave: problema de alocação de berços; algoritmo de treinamento populacional; geração de colunas.

Abstract

The Berth Allocation Problem (PAB) consists of ships programming and allocation to the mooring areas along a quay. PAB is modeled as a Multi-Depot Vehicle Routing Problem with Time Windows. A recently proposed method denominated ATP/PL is used to solve it. This method consists of the application of the Population Training Algorithm (ATP) with Linear Programming (PL) for Column Generation. These methods are applied in an interactive way, where ATP uses PL relaxation information for the generation of good columns, and PL is used for solving a Set Partitioning Problem, with an additional restriction (PPC⁺), formed by these columns. The computational results are obtained through instances used in a recent work and compared against results from CPLEX and other method found in literature.

Keywords: berth allocation problem; population training algorithm; column generation.

1. Introdução

O constante crescimento econômico, o volume crescente do comércio entre os diversos países, e os aumentos dos avanços na tecnologia de construção de grandes navios, induzem ao aumento da frequência de transportes entre portos, e consequentemente ao aumento da importância desses portos na economia de um país.

Segundo Hansen *et al.* (2008), um porto deve operar de forma eficiente para competir neste ambiente. Além disso, de acordo com Imai *et al.* (2003), a alocação e a programação de navios a berços têm um impacto primário na eficiência dessas operações. Uma discussão a respeito dos problemas de decisão que surgem em um porto é apresentada em Vis & Koster (2003). Steenken *et al.* (2004) apresentam uma classificação dos principais processos e operações em um porto. O problema crucial no gerenciamento de um porto é a otimização do equilíbrio entre os proprietários dos navios, que exigem serviços rápidos, e o uso econômico dos recursos disponíveis no porto (Dragovic *et al.*, 2005).

O Problema de Alocação de Berços (PAB) reflete as principais decisões referentes aos recursos de custo mais importante no gerenciamento de portos de cargas. Assim, uma boa distribuição dos navios aos berços aumentará a satisfação dos proprietários dos navios e aumentará a produtividade do porto, conduzindo a rendas mais altas para ambas as partes (Imai *et al.*, 2003).

Neste artigo, o problema é tratado em sua forma discreta, considerando como objetivo principal a minimização do tempo total gasto pelos navios dentro do porto, que segundo Hansen *et al.* (2008), é uma função objetivo apropriada para o PAB. Essa abordagem é baseada nos trabalhos apresentados por Imai *et al.* (1997, 2001, 2003) e Cordeau *et al.* (2005).

Para resolver o PAB, aplica-se um método proposto recentemente, denominado ATP/PL (ver Mauri & Lorena, 2004), que é baseado na técnica de geração de colunas. É utilizado um modelo matemático baseado em um problema de roteamento de veículos para representar o problema, e o ATP/PL é utilizado para tratá-lo, ou seja, alocar e programar as atracções dos navios aos berços de forma a reduzir o tempo de permanência dos navios no porto.

O artigo está organizado como segue. A Seção 2 apresenta uma breve descrição do problema. A modelagem do problema é apresentada na Seção 3. Já a Seção 4 descreve o método utilizado para resolver o PAB. Os resultados computacionais obtidos são apresentados na Seção 5, e as conclusões são resumidas na Seção 6.

2. Descrição do problema

O Problema de Alocação de Berços (PAB) consiste em atribuir os navios que chegam a um determinado porto para as “posições” de atracção disponíveis ao longo de um cais (berços). As principais decisões a serem tomadas neste processo envolvem a escolha de onde e quando os navios deverão atracar (Cordeau *et al.*, 2005).

Em relação ao local de atracção, existem restrições relativas à profundidade da água, à distância máxima em relação ao local mais favorável ao longo do cais, e também ao tamanho dos navios. Já em relação ao horário de atracção dos navios, as restrições são expressas como janelas de tempo para conclusão de seu atendimento (Cordeau *et al.*, 2005).

O tempo de atendimento de um navio depende de seu ponto de atracção (berço), e é uma função da distância do berço até a área de carga e descarga de containeres no pátio do porto.

Como mencionado em Cordeau *et al.* (2005), esta dependência afeta fortemente o desempenho das operações no porto.

Os dados referentes ao tempo de serviço dependem de outra decisão, que é o número de guindastes no cais que estão disponíveis aos navios que chegam. Essa decisão é conhecida como *Problema de Atribuição de Guindastes – PAG* (ver Lee *et al.*, 2008) e afeta o tempo de serviço dos navios. Consequentemente, o PAG tem um impacto no PAB. Em um sistema complexo, como um porto de transferência de cargas, o processo de tomada de decisão é frequentemente hierárquico, e o PAG é resolvido antes do PAB.

O objetivo então é minimizar os custos referentes ao porto e ao navio, que é relacionado ao tempo de serviço dos navios. O objetivo do PAB normalmente é minimizar o tempo de serviço total de todos os navios. Caso os navios não tenham a mesma importância, uma soma dos tempos de serviços dos navios, considerando uma penalização para indicar a sua devida importância, pode refletir melhor a prática de gerenciamento de alguns portos. Os pesos nesta soma podem representar um esquema do valor estimado da carga ou do número de containeres movimentados. Em algumas variantes do problema também podem ser incluídas outras condições de penalidade na função objetivo, como prioridade de atendimento, valor de carga movimentada, e outras (ver Imai *et al.*, 2008).

O PAB pode ser modelado como um problema discreto se o cais for visto como um conjunto finito de berços. Neste caso, os berços podem ser descritos como segmentos de comprimento fixos, ou, se a dimensão de espaço for ignorada, como pontos. Já os modelos contínuos consideram que os navios podem atracar em qualquer lugar ao longo do cais. No caso discreto, o PAB pode ser modelado como um *Problema de Programação de Máquinas Paralela Sem Conexão* (Pinedo, 1995), onde um navio é tratado como um trabalho e um berço como uma máquina. O PAB ainda pode ser modelado como um *Problema de Roteamento de Veículos com Múltiplas Garagens e Janelas de Tempo* (Legato *et al.*, 2001), onde os navios são tratados como clientes e os berços como garagens. Já no caso contínuo, o PAB pode ser representado como um *Problema de Corte e Estoque Bidimensional* com algumas restrições adicionais. Em ambos os casos, o PAB é *NP-Hard* (Garey & Johnson, 1979).

3. Modelagem do problema

Neste trabalho, o *Problema de Alocação de Berços* (PAB) é tratado em sua forma discreta, onde o cais é dividido em um conjunto finito de berços, e a dimensão espacial é ignorada. Assim, deve-se observar, para cada navio, os tempos descritos na Figura 1.

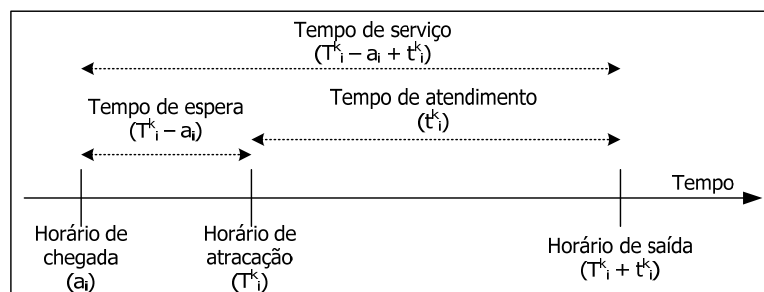


Figura 1 – Variáveis referentes ao tempo.

Como observado por Legato *et al.* (2001), o PAB pode ser modelado como um *Problema de Roteamento de Veículos com Garagens Múltiplas e Janelas de Tempo* (ver Cordeau *et al.*, 2001). Neste trabalho, o PAB é representado inicialmente através do modelo matemático proposto por Cordeau *et al.* (2005).

Nesse modelo, os navios são tratados como clientes e os berços como garagens (cada uma com seu veículo específico). Existem então m veículos “fictícios” (um para cada garagem), sendo que cada um inicia e termina sua “rota” na sua própria garagem. Os navios são modelados como vértices em um multigrafo, onde cada garagem (berço) ainda é dividida em um vértice de origem e um de destino. Nos vértices de origem e destino, as janelas de tempo correspondem ao período de funcionamento dos berços.

O modelo então é dado por um multigrafo $G^k = (V^k, A^k)$, $\forall k \in M$, onde $V^k = N \cup \{o(k), d(k)\}$ e $A^k \subseteq V^k \times V^k$. As variáveis e parâmetros usados para representar o problema são:

Parâmetros:

- N : conjunto de navios, $n = |N|$;
- M : conjunto de berços, $m = |M|$;
- t_i^k : duração do atendimento do navio i no berço k ;
- a_i : horário de chegada do navio i ;
- s^k : horário de abertura do berço k ;
- e^k : horário de fechamento do berço k ;
- b_i : horário de término da janela de tempo para serviço do navio i ;
- v_i : valor (custo) do tempo de serviço do navio i ;
- $M_{ij}^k = \max\{b_i + t_i^k - a_j, 0\}$, $\forall k \in M, \forall (i,j) \in N$.

Variáveis:

- $x_{ij}^k \in \{0,1\}$ $\forall k \in M, \forall (i,j) \in A^k$, $x_{ij}^k = 1$ se o navio j é atendido pelo berço k após o navio i ;
- T_i^k $\forall k \in M, i \in N$ é o horário que o navio i atracou no berço k ;
- $T_{o(k)}^k$ $\forall k \in M$ é o horário em que o primeiro navio atracou no berço k ;
- $T_{d(k)}^k$ $\forall k \in M$ é o horário em que o último navio saiu do berço k ;

O modelo do PAB proposto por Cordeau *et al.* (2005) é escrito da seguinte forma:

Minimizar:

$$Z = \sum_{i \in N} \sum_{k \in M} v_i \left[T_i^k - a_i + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{ij}^k \right] \quad (1)$$

Sujeito a:

$$\sum_{k \in M} \sum_{j \in N \cup \{d(k)\}} x_{ij}^k = 1 \quad \forall i \in N \quad (2)$$

$$\sum_{j \in N \cup \{d(k)\}} x_{o(k)j}^k = 1 \quad \forall k \in M \quad (3)$$

$$\sum_{i \in N \cup \{o(k)\}} x_{i,d(k)}^k = 1 \quad \forall k \in M \quad (4)$$

$$\sum_{j \in N \cup \{d(k)\}} x_{i,j}^k - \sum_{j \in N \cup \{o(k)\}} x_{j,i}^k = 0 \quad \forall k \in M, \forall i \in N \quad (5)$$

$$T_i^k + t_i^k - T_j^k \leq (1 - x_{i,j}^k) M_{i,j}^k \quad \forall k \in M, \forall (i, j) \in A^k \quad (6)$$

$$T_i^k \geq a_i \quad \forall k \in M, \forall i \in N \quad (7)$$

$$T_i^k + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{j,i}^k \leq b_i \quad \forall k \in M, \forall i \in N \quad (8)$$

$$T_{o(k)}^k \geq s^k \quad \forall k \in M \quad (9)$$

$$T_{d(k)}^k \leq e^k \quad \forall k \in M \quad (10)$$

$$x_{i,j}^k \in \{0,1\} \quad \forall k \in M, \forall (i, j) \in A^k \quad (11)$$

A função objetivo Z minimiza o tempo decorrido desde o momento em que os navios chegam, atracam e são atendidos, considerando um custo de serviço para esse tempo. A restrição (2) garante que cada navio é atendido por apenas um berço. As restrições (3) e (4) garantem, respectivamente, que um navio será o primeiro a ser atendido em cada berço, e outro será o último. A restrição (5) garante a conservação do fluxo (atendimento) para os demais navios. A restrição (6) faz o cálculo do horário de atracação dos navios. Nessa restrição são considerados apenas os arcos A^k válidos para cada berço k , ou seja, alguns navios não podem ser atendidos em determinados berços, pois por exemplo, o tipo de equipamento disponível no berço pode não ser apropriado para o atendimento de determinados tipos de carga. A possibilidade de atendimento ou não dos navios pelos berços é determinada através dos dados encontrados nas instâncias utilizadas (o tempo de atendimento é zero). As restrições (7) e (8) garantem, respectivamente, que o horário de atracação seja após a chegada do navio, e que o horário do término do atendimento do navio seja anterior ao horário limite do navio (janela de tempo). As restrições (9) e (10) garantem a não violação das janelas de tempo nos berços. E por fim, a restrição (11) garante que as variáveis de decisão sejam binárias.

4. ATP/PL

Proposto inicialmente por Mauri & Lorena (2004), o ATP/PL é baseado na aplicação do Algoritmo de Treinamento Populacional (ATP) juntamente com a Programação Linear (PL) através da técnica de Geração de Colunas. O ATP e a PL são aplicados de maneira interativa, sendo o ATP, através de informações da relaxação PL, responsável pela geração de boas colunas (baixo custo e boa cobertura dos navios), e a PL pela resolução de um Problema de Particionamento de Conjuntos (PPC) formado por essas colunas. O PPC é formulado da seguinte maneira:

Minimizar:

$$Z^* = \sum_{j=1}^p c_j x_j \quad (12)$$

Sujeito a:

$$\sum_{j=1}^p a_{ij} x_j = 1 \quad i = 1, \dots, n \quad (13)$$

$$x_j \in \{0,1\}; \quad j = 1, \dots, p \quad (14)$$

Na formulação acima, o problema é descrito como o de formação de uma matriz, onde as colunas representam os berços, e as linhas os navios. Cada elemento $a_{ij} \in \{0,1\}$, sendo $i \in N = \{1..n\}$ e $j \in P = \{1..p\}$, n é o número de navios (linhas) e p o número de colunas geradas, onde $a_{ij} = 1$ se a coluna j atender o navio i , e 0 caso contrário. Esta é uma formulação clássica e constantemente utilizada em diversos trabalhos encontrados na literatura (Desrochers & Soumis, 1989; Wilhelm, 2001). c_j representa o custo da coluna j (eq. 16) e x_j é igual a 1 se a coluna j pertencer à solução do problema e 0 caso contrário.

No caso específico do PAB, ainda deve-se considerar que cada berço possui suas próprias características, podendo não ter a capacidade de atender a um determinado navio. Nesse caso, deve-se garantir que seja utilizado nenhum ou apenas um berço de cada “tipo” disponível no cais, ou seja, cada coluna pertencente à solução final do problema deverá representar um berço distinto, e sem repetições. Para isso, deve-se inserir uma nova restrição no PPC (eq. 15), formando assim um problema de particionamento de conjuntos com uma restrição adicional (PPC⁺).

$$\sum_{j=1}^p b_{ij} x_j \leq 1 \quad i = 1, \dots, m \quad (15)$$

Cada elemento $b_{ij} \in \{0,1\}$, sendo $i \in M = \{1..m\}$ e $j \in P = \{1..p\}$, m é o número de berços disponíveis, onde $b_{ij} = 1$ se a coluna j representar o berço i . Cada coluna é representada através de um “indivíduo” formado por números inteiros, onde a primeira posição indica o berço referente à coluna, e as demais posições representam os navios atendidos por esse berço (coluna). A sequência de atendimento de um berço k qualquer é dada por:

$$B^k = \begin{array}{|c|c|c|c|c|} \hline k & 10 & 3 & 1 & 5 \\ \hline \end{array}$$

O indivíduo representado acima indica que o berço k atenderá os navios 10, 3, 1 e 5, nessa ordem. O custo de cada coluna (indivíduo) é dado pela equação (16). Para calcular o custo das colunas, as restrições referentes às janelas de tempo (7-10) são alocadas na função objetivo, juntamente com fatores de penalização (vetor $w = [w_0, w_1, w_2]$). Dessa forma, as colunas são avaliadas de forma relaxada, e nos casos de violações nas janelas de tempo, o custo da coluna receberá uma penalização elevada.

$$c_k = w_0 \sum_{i \in B^k} v_i (T_i^k - a_i + t_i^k) + w_1 \sum_{i \in B^k} (\max(0, a_i - T_i^k) + \max(0, T_i^k + t_i^k - b_i)) + w_2 (\max(0, s^k - T_{o(k)}^k) + \max(0, T_{d(k)}^k - e^k)) \quad (16)$$

A geração das colunas necessárias para resolver o PPC⁺ (eq. 12-15) 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. Maiores detalhes sobre o ATP/PL podem ser vistos em Mauri & Lorena (2007).

4.1 Algoritmo de Treinamento Populacional

O Algoritmo de Treinamento Populacional (ATP) é um tipo de técnica evolutiva empregado inicialmente por Oliveira (2002) e Oliveira & Lorena (2002), e foi derivado do Algoritmo Genético Construtivo (AGC). O AGC foi proposto por Lorena & Furtado (2001) e possui várias características inovadoras comparadas aos algoritmos genéticos tradicionais. Uma dessas características é a utilização de uma população ordenada, de tamanho dinâmico, e composta de “esquemas” (soluções incompletas) e “estruturas” (soluções completas). O aspecto construtivo do AGC está relacionado ao fato de que, ao longo do processo evolutivo, uma população de esquemas é utilizada como base para a construção de uma população de estruturas. Tanto esquemas quanto estruturas competem entre si durante o processo evolutivo, e ambos 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 a heurística de treinamento utilizada não for capaz de melhorá-lo. A adaptação no treinamento da população é usada para guiar a busca para áreas promissoras.

As duas funções usadas no treinamento evolutivo são definidas através de $g(k)$ = “qualidade” do indivíduo (coluna) k (ver eq. 19 e 20), e $f(k)$ = Melhor $g(k')$ | $k' \in$ Vizinhança (k). O valor de $f(k)$ é obtido através da heurística de treinamento (ver Figura 4). O processo evolutivo é desenvolvido privilegiando os indivíduos de menor diferença $[g(k) - f(k)]$ e menor $g(k)$, atribuindo a eles os seguintes *ranks*:

$$\delta(k) = d \times [g_{\max} - g(k)] - [g(k) - f(k)] \quad (17)$$

g_{\max} é o custo (alto) do pior indivíduo criado na população inicial e d é um percentual constante de g_{\max} . A população é controlada dinamicamente por um parâmetro de evolução, denominado α , que é atualizado por:

$$\alpha = \alpha + Step \times PS \times \frac{\delta_{bst} - \delta_{wst}}{RG} \quad (18)$$

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* (eq. 17), e se $\alpha \geq \delta(k)$ o indivíduo k é eliminado da população. A população no momento de evolução α é dinâmica em tamanho e pode ser esvaziada durante o processo.

A solução inicial é gerada através de duas heurísticas: *heurística de distribuição* e *heurística de programação*. A heurística de distribuição é responsável pela atribuição dos navios aos berços. Essa heurística é baseada na heurística de distribuição apresentada em Mauri e Lorena (2006) e na heurística *FCFS-G*, apresentada em Cordeau *et al.* (2005). Já a heurística de programação determina o horário de atendimento dos navios nos berços. A heurística de distribuição é executada para cada indivíduo da população inicial.

Na heurística de distribuição são criadas m colunas (berços) vazias. Os n navios são organizados por ordem de chegada ao porto, e são distribuídos sequencialmente aos berços, que são selecionados de forma aleatória, porém sempre verificando se este poderá atender o navio em questão. Após essa distribuição, fica garantido que cada navio foi atribuído a um berço que poderá atendê-lo.

O horário em que o navio será atendido ainda poderá ser incoerente, podendo apresentar sobreposição e/ou violações nas janelas de tempo, tanto dos navios quanto do berço. A Figura 2 apresenta a heurística de distribuição.

CRIAR (m berços (colunas) vazios);
CRIAR (uma lista L com todos os navios);
ORDENAR (a lista L pelo horário de chegada dos navios ao porto);
PARA (cada navio j em L, j = 1,2,...,n) FACA
SELECIONAR (um berço i, i = 1,2,...,m);
SE (o berço i não puder atender ao navio j)
VOLTAR (para o passo 5);
SENÃO
ATRIBUIR (o navio j ao berço i);
FIM-PARA;

Figura 2 – Heurística de distribuição.

Na heurística de programação, são efetuados os cálculos do horário de atracação de cada navio, do custo da coluna k (c_k), das funções $g(k)$ e $f(k)$ e do *rank* da coluna $\delta(k)$. A Figura 3 apresenta a heurística de programação.

SEJA (k uma coluna qualquer);
PARA (cada navio i atribuído a k) FACA

$$T_i^k = \begin{cases} \max(a_i, s^k), & i = 1 \\ \max(a_i, T_{i-1}^k + t_{i-1}^k), & i > 1 \end{cases}$$

FIM-PARA;
CALCULAR (c_k , $g(k)$, $f(k)$ e $\delta(k)$);

Figura 3 – Heurística de programação.

Como função de treinamento $f(k)$, é utilizada uma heurística de busca local simples, que é responsável por avaliar vários indivíduos (colunas) alternativos em uma vizinhança. Essa heurística é descrita na Figura 4.

SEJA (k uma coluna qualquer);
 $k' \leftarrow k$;
 $f^* = g(k')$
POR (tamanho da vizinhança vezes)
 $i \leftarrow$ um navio qualquer atendido pela coluna k' ;
 $j \leftarrow$ outro navio qualquer atendido pela coluna k' ;
TROCAR (a ordem de atendimento dos navios i e j);
EXECUTAR (a heurística de programação para a coluna k');
SE ($g(k') < f^*$);
 $f^* \leftarrow g(k')$;
FIM-SE;
FIM-POR;
 $f(k) \leftarrow f^*$;

Figura 4 – Heurística de treinamento.

A mutação utilizada também é baseada em uma busca local, implementada através de uma simples troca das posições de atendimento de dois navios (selecionados aleatoriamente) atendidos por uma coluna (indivíduo). Esse processo é descrito na Figura 5.


```

SEJA (k uma coluna qualquer);
i ← um navio qualquer atendido pela coluna k;
j ← outro navio qualquer atendido pela coluna k;
TROCAR (a ordem de atendimento dos navios i e j);
EXECUTAR (a heurística de programação para a coluna k);
    
```

Figura 5 – Mutação.

O cruzamento é responsável por gerar novos indivíduos da seguinte maneira: dois indivíduos são selecionados (base e guia). A partir de então, é criado um novo indivíduo semelhante à base. Assim, cada navio atendido pelo indivíduo guia é inserido nesse novo indivíduo, caso o berço correspondente ao novo possa atender ao navio em questão. Feito isso, a sequência de atendimento do novo indivíduo é ordenada pelo horário de chegada dos navios ao porto. O cruzamento é descrito na Figura 6.

```

SEJA (k uma coluna base);
SEJA (k' uma coluna guia);
k'' ← clone(k);
PARA (cada navio i atendido pela coluna k') FAÇA
  SE (o berço referente à coluna k'' for capaz de atender ao navio i)
    INSERIR (o navio i na coluna k'');
  FIM-SE;
FIM-PARA;
ORDENAR (a sequência de atendimento da coluna k'');
EXECUTAR (a heurística de programação para a coluna k'');
INSERIR (a coluna k'' na população);
    
```

Figura 6 – Cruzamento.

Esses operadores são incorporados ao ATP, cujo pseudocódigo é mostrado na Figura 7. É interessante notar que, utilizando esses processos, o ATP formará populações de vários tamanhos, guiados pelo objetivo de selecionar colunas de baixo custo e com uma cobertura suficiente dos navios. As melhores colunas deverão incluir um número de navios diversificado, caracterizado pela heurística de treinamento (função de treinamento) que guiará o processo evolutivo.

```

CRIAR (população inicial);
ENQUANTO (número máximo de gerações não for atingido)
  SELECIONAR (indivíduo base);
  SELECIONAR (indivíduo guia);
  k ← CRUZAMENTO (base,guia);
  SE (rand() < probabilidade de mutação)
    MUTACÃO (k);
  FIM-SE;
  CALCULAR ( $\delta(k)$ );
  SE ( $\delta(k) > \alpha$ )
    INSERIR (k na população);
  FIM-SE;
  ORDENAR (a população pelo rank dos indivíduos);
  ATUALIZAR ( $\alpha$ );
  PARA (todo k ∈ população) FAÇA
    SE ( $\delta(k) \leq \alpha$ )
      ELIMINAR (k);
    FIM-SE;
  FIM-PARA;
FIM-ENQUANTO;
    
```

Figura 7 – Algoritmo ATP.

4.2 Interação ATP e PL

Definida a implementação do ATP, tal metodologia é incorporada à ATP/PL. A interação do ATP com a PL é baseada principalmente na definição da função de adaptação (função g) dos indivíduos no ATP. Essa função é definida com o uso de informações das variáveis duais retiradas da PL. A função g é definida da seguinte maneira:

$$g(k) = \frac{c_k}{\sum_{i=1}^n \lambda_i a_{ik} + \sum_{i=1}^m \lambda_i b_{ik}} \quad \text{para} \quad \left(\sum_{i=1}^n \lambda_i a_{ik} + \sum_{i=1}^m \lambda_i b_{ik} > 0 \right) \quad (19)$$

$$g(k) = c_k \quad \text{para} \quad \left(\sum_{i=1}^n \lambda_i a_{ik} + \sum_{i=1}^m \lambda_i b_{ik} \leq 0 \right) \quad (20)$$

c_k é o custo da coluna k (eq. 16); λ_i é a variável dual para a para a restrição i . Através dos conceitos da técnica de geração de colunas, pode-se calcular o custo reduzido da k -ésima coluna inserida no PPC⁺, denotado por θ_k , através da equação:

$$\theta_k = c_k - \left(\sum_{i=1}^n \lambda_i a_{ik} + \sum_{i=1}^m \lambda_i b_{ik} \right) \quad (21)$$

A partir das equações (19), (20) e (21), pode-se notar que para custos reduzidos negativos ($\theta_k < 0$), o valor da função g estará dentro do intervalo $[0,1]$, fazendo com que a heurística de treinamento, que define o valor da função f correspondente (melhor g em uma vizinhança), proporcione uma pequena diferença ($g - f$) para colunas que tenham custo reduzido negativo. Para custos reduzidos positivos ($\theta_k \geq 0$) o valor da função g será igual ao custo, ou seja, um valor “elevado”. Assim, a população é treinada indiretamente para indivíduos com custo reduzido negativo, melhorando a cobertura dos navios para o PPC⁺ e evitando a geração de um número excessivo de colunas, e conseqüentemente acelerando o processo de geração de colunas. A Figura 8 apresenta o pseudocódigo da ATP/PL.

CRIAR (conjunto inicial de colunas);
RESOLVER (PL);
POR (número de iterações ou número máximo de colunas não for gerado)
EXECUTAR (ATP);
REMOVER (colunas inválidas);
CALCULAR (custo reduzido das novas colunas);
ADICIONAR (colunas com custo reduzido negativo);
RESOLVER (PL);
FIM-POR;
CONVERTER (PL para PLI);
RESOLVER (PLI);

Figura 8 – Algoritmo ATP/PL.

Um conjunto inicial de colunas é gerado aleatoriamente, porém de forma direcionada ao problema, ou seja, contendo colunas que formem uma solução para o PPC⁺. Essas colunas são geradas através de uma única execução da *heurística de distribuição* (Figura 2) seguida pela execução da *heurística de programação* (Figura 3) para cada coluna. É interessante destacar que a solução formada por essas colunas provavelmente será inválida, pois as

colunas podem apresentar violações nas janelas de tempo, porém devido ao seu alto custo (devido às penalizações), essas colunas deixarão de pertencer à solução à medida que novas colunas forem criadas.

A partir do conjunto inicial de colunas gerado é formado um PPC^+ , que é resolvido através da PL. Feito isso, novas colunas são geradas através do ATP, considerando os valores das variáveis duais, para construir as funções de adaptação. As colunas válidas (que não apresentam violações nas janelas de tempo) e com custo reduzido negativo são adicionadas ao atual PPC^+ , e este é resolvido novamente através da PL. Esse processo é repetido por um determinado número de iterações ou enquanto um número máximo de colunas não for gerado.

Finalmente, terminada essa interação, a PL é convertida em Programação Linear Inteira – PLI e resolvida através do software CPLEX, a qual apresenta uma solução viável para o PPC^+ , e conseqüentemente para o PAB.

5. Resultados computacionais

Para avaliar o desempenho do ATP/PL, foram utilizadas 30 instâncias distintas, cada uma com 60 navios e 13 berços. Essas instâncias foram geradas aleatoriamente por Cordeau *et al.* (2005). Todos os testes foram realizados em um *PC* com processador *AMD Athlon™ 64 3500* de 2.2 GHz e 1GB de memória RAM. Toda a implementação foi desenvolvida na linguagem C++.

Os parâmetros utilizados pelo ATP/PL são apresentados na Tabela 1. Além disso, em todos os experimentos, os valores de g_{max} utilizados foram obtidos a partir da maior função g dos indivíduos gerados na população inicial. Já o valor de α , inicialmente, foi igual a 0. As penalizações utilizadas foram $w = [1, 10, 10]$.

Tabela 1 – Parâmetros do ATP/PL.

Parâmetro	Valor
Tamanho da pop. Inicial	10
<i>Step</i>	0,001
N. máximo de gerações	70
Percertual base	10
Prob. de mutação	60
Tamanho da vizinhança	6
<i>d</i>	0,01
Número máximo de colunas	70000
Número de iterações	10000

A Tabela 2 apresenta alguns detalhes da atuação do ATP/PL. Nessa tabela, pode-se notar que o número médio de colunas geradas para cada instância é relativamente baixo em relação ao número máximo de colunas previsto. Pode-se notar também que o valor da solução do último PPC^+ (formado por todas as colunas geradas) foi o mesmo em todos os casos, sendo resolvido tanto através da PL quanto da PLI, o que indica a obtenção da solução ótima para o PPC^+ formado pelas colunas geradas. O tempo de interação entre o ATP e a PL também é baixo, assim como o tempo para resolução do PPC^+ final através da PLI, o que resulta em um tempo total de processamento competitivo para o ATP/PL.

Tabela 2 – Detalhes do ATP/PL.

Instância	N. de colunas geradas	Solução do PPC ⁺ via PL	Solução do PPC ⁺ via PLI	Tempo de processamento (seg.)		
				Interação	PLI	Total
i01	26664	1409,00	1409	72,14	2,47	74,61
i02	12752	1261,00	1261	58,92	1,83	60,75
i03	70000	1129,00	1129	94,62	40,83	135,45
i04	54612	1302,00	1302	103,16	7,02	110,17
i05	70019	1207,00	1207	72,20	52,50	124,70
i06	25990	1261,00	1261	74,22	4,12	78,34
i07	70023	1279,00	1279	86,73	27,47	114,20
i08	70005	1299,00	1299	48,77	8,30	57,06
i09	37846	1444,00	1444	91,86	4,61	96,47
i10	70005	1213,00	1213	61,81	37,59	99,41
i11	43507	1369,00	1369	95,34	4,00	99,34
i12	18508	1325,00	1325	77,39	3,30	80,69
i13	70017	1360,00	1360	62,55	27,39	89,94
i14	26221	1233,00	1233	69,05	4,91	73,95
i15	70002	1295,00	1295	71,28	2,91	74,19
i16	30063	1365,00	1365	169,81	0,55	170,36
i17	70033	1283,00	1283	32,89	13,67	46,58
i18	36108	1345,00	1345	81,78	2,23	84,02
i19	16135	1367,00	1367	122,00	1,19	123,19
i20	20528	1328,00	1328	74,25	8,05	82,30
i21	48386	1341,00	1341	103,52	4,56	108,08
i22	54140	1326,00	1326	104,17	1,20	105,38
i23	70010	1266,00	1266	41,59	2,12	43,72
i24	70008	1260,00	1260	75,81	3,09	78,91
i25	41210	1376,00	1376	95,09	1,48	96,58
i26	70011	1318,00	1318	70,00	31,11	101,11
i27	37022	1261,00	1261	77,38	5,48	82,86
i28	70004	1360,00	1360	51,52	1,39	52,91
i29	70001	1280,00	1280	196,36	7,00	203,36
i30	7837	1344,00	1344	69,62	1,39	71,02
Média	48256	1306,87	1306,87	83,53	10,46	93,99

As soluções obtidas pelo ATP/PL foram comparadas com as melhores soluções conhecidas para as instâncias utilizadas. Essas melhores soluções foram obtidas através de uma heurística baseada na Busca Tabu, apresentada em Cordeau *et al.* (2005). Além disso, o CPLEX 10.0.1 (Ilog, 2006), também foi utilizado, de forma isolada, para resolver o modelo descrito na Seção 3. Foi utilizado um limite máximo de processamento de 1 hora para cada instância. A Tabela 3 apresenta essas comparações.

Tabela 3 – Comparações com outros métodos.

Instância	BT	CPLEX		ATP/PL	Melhorias (%)	
	Z	Z	Gap	Z*	A	B
i01	1415	-	-	1409	0,43	-
i02	1263	2606	3,82	1261	0,16	106,66
i03	1139	2565	4,00	1129	0,89	127,19
i04	1303	4353	8,62	1302	0,08	234,33
i05	1208	2672	4,89	1207	0,08	121,38
i06	1262	-	-	1261	0,08	-
i07	1279	2887	4,73	1279	0,00	125,72
i08	1299	5177	11,69	1299	0,00	298,54
i09	1444	-	-	1444	0,00	-
i10	1212	-	-	1213	-0,08	-
i11	1378	-	-	1369	0,66	-
i12	1325	3206	5,48	1325	0,00	141,96
i13	1360	-	-	1360	0,00	-
i14	1233	-	-	1233	0,00	-
i15	1295	4672	9,77	1295	0,00	260,77
i16	1375	4320	8,97	1365	0,73	216,48
i17	1283	-	-	1283	0,00	-
i18	1346	3681	6,94	1345	0,07	173,68
i19	1370	2400	3,04	1367	0,22	75,57
i20	1328	-	-	1328	0,00	-
i21	1346	-	-	1341	0,37	-
i22	1332	3489	7,31	1326	0,45	163,12
i23	1266	-	-	1266	0,00	-
i24	1261	4867	10,13	1260	0,08	286,27
i25	1379	1993	2,67	1376	0,22	44,84
i26	1330	2520	3,62	1318	0,91	91,20
i27	1261	3209	5,70	1261	0,00	154,48
i28	1365	-	-	1360	0,37	-
i29	1282	4809	9,43	1280	0,16	275,70
i30	1351	-	-	1344	0,52	-
Média	1309,67	3495,65	6,52	1306,87	0,21	170,46

A função objetivo (1) do modelo original do problema (ver Seção 3) pode ser reescrita como apresentado a seguir (eq. 22). Assim, pode-se notar que esta apresenta um termo constante (somatórios de $v_i a_i$) que é subtraído. Como o CPLEX não “aceita” a inclusão de valores constantes na função objetivo dos modelos, estes foram resolvidos sem considerar o termo constante, e os *gaps* apresentados na Tabela 3 foram obtidos dessa forma. Porém, quando o termo constante é subtraído do valor da solução obtida pelo CPLEX, tem-se o valor da solução para o problema.

$$\sum_{i \in N} \sum_{k \in M} \left[v_i T_i^k + \left(v_i t_i^k \sum_{j \in N \cup \{d(k)\}} x_{ij}^k \right) \right] - \sum_{i \in N} \sum_{k \in M} v_i a_i \quad (22)$$

Considerando a instância *i02*, por exemplo, o valor do termo constante é 33215. Já a solução factível obtida pelo CPLEX é 35821, e o limitante inferior é 34451,58. Calculando o *gap* então, tem-se:

$$gap = \left(\frac{35821 - 34451,85}{35821} \right) \times 100 = 3,82\%$$

Entretanto, o valor da solução do problema dever considerar o termo constante, logo, seria $35821 - 33215 = 2606$ (*Z*), e conseqüentemente, o limitante inferior seria $34451,58 - 33215 = 1236,58$, que obviamente é inferior às soluções do ATP/PL e BT, e indica que a solução obtida pelo ATP/PL (1261), por exemplo, está próxima da solução ótima para a instância *i02* (que será entre 1237 e 1261). Como pode ser observado na Tabela 3, esse fato se repete para as demais instâncias.

Ainda na Tabela 3, a coluna “A” apresenta a melhoria obtida pelo ATP/PL em relação à Busca Tabu (BT). Já a coluna “B” apresenta a melhoria do ATP/PL em relação ao CPLEX. As soluções obtidas pelo ATP/PL foram 0,21% melhores em relação às obtidas pela Busca Tabu proposta por Cordeau *et al.* (2005). A Busca Tabu obteve apenas uma solução melhor do que o ATP/PL, para a instância *i10*. Já em relação ao CPLEX, pode-se notar que este não foi capaz de obter soluções para várias instâncias (em 1 hora), e nos casos em que foram encontradas soluções, os resultados foram expressivamente piores do que os apresentados pelo ATP/PL, em média 170,46% piores.

Em relação ao tempo para obtenção das soluções, o CPLEX utilizou 1 hora para cada instância (3600 seg.). A Busca Tabu utilizou aproximadamente 120 segundos para cada instância, como descrito em Cordeau *et al.* (2005). Já o ATP/PL, utilizou um tempo médio de 93,99 segundos para cada instância, o que mostra a competitividade do método em relação à Busca Tabu e a superioridade em relação ao CPLEX.

6. Conclusões

Foi utilizado um modelo matemático baseado em um problema de roteamento de veículos para representar o PAB. Para resolvê-lo, foi proposta a aplicação de um método denominado ATP/PL, que foi proposto recentemente, e é baseado na técnica de geração de colunas.

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 através da definição da *aplicação-fg* utilizando informações dos valores duais. Dessa forma, esse método mostrou ser adequado para outros problemas para os quais a geração de colunas seja indicada. Os resultados computacionais do ATP/PL foram excelentes e obtidos em tempos razoáveis de processamento, comparados à Busca Tabu e ao CPLEX.

O ATP/PL foi capaz de obter, em todos os casos, e com pouco tempo de processamento, soluções válidas para o problema. Além disso, o método proposto se mostrou extremamente eficiente, pois como podem ser observadas na Tabela 3, as soluções foram superiores às obtidas pelo CPLEX, que por sua vez apresentou *gaps* relativamente baixos, indicando uma grande proximidade com as soluções ótimas para o problema. Os operadores genéticos e a

heurística de treinamento utilizada pelo ATP na fase de geração de colunas foram adequados e eficientes para exploração do espaço de soluções.

Os resultados obtidos mostram que o ATP/PL foi capaz de gerar soluções de boa qualidade para todas as instâncias em tempos computacionais expressivamente baixos, e ainda foram comparados com uma abordagem recente encontrada na literatura, e a qualidade das soluções encontradas foi ligeiramente superior. Além disso, os resultados obtidos ainda foram comparados com o CPLEX, e nesse caso, nota-se claramente a superioridade do método proposto tanto na qualidade das soluções quanto no tempo de processamento.

Enfim, os resultados mostram claramente o potencial da abordagem apresentada, onde soluções de alta qualidade são obtidas, para problemas relativamente grandes e em tempos de processamento expressivamente baixos. A continuação deste trabalho será voltada para o tratamento do problema em sua forma contínua, e para a aplicação do método proposto a problemas reais encontrados em portos brasileiros.

Agradecimentos

Os autores agradecem aos revisores anônimos pelos comentários e sugestões, e à Fundação de Amparo à Pesquisa do Estado de São Paulo – FAPESP (processo 04/11053-9) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (processos 300692/2009-9, 470813/2010-5, 473205/2008-4 e 558084/2009-5) pelo apoio financeiro parcial dado ao desenvolvimento deste trabalho.

Referências Bibliográficas

- (1) Cordeau, J.F.; Laporte, G. & Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, **52**, 928-936.
- (2) Cordeau, J.F.; Laporte, G.; Legato, P. & Moccia, L. (2005). Models and tabu search heuristics for the berth allocation problem. *Transportation Science*, **39**, 526-538.
- (3) Desrochers, M. & Soumis, F. (1989). A column generation approach to the urban transit crew scheduling problem. *Transportation Science*, **23**, 1-13.
- (4) Dragovic, B.; Park, N.K. & Radmilovic, Z. (2005). Simulation modelling of ship-berth link with priority service. *Maritime Economics & Logistics*, **7**, 316-335.
- (5) Garey, M.R. & Johnson, D.S. (1979). *Computers and intractability: a guide to the theory of np-completeness*. Freeman, San Francisco, USA.
- (6) Hansen, P.; Oğuz, C. & Mladenović, N. (2008). Variable neighborhood search for minimum cost berth allocation. *European Journal of Operational Research*, **191**, 636-649.
- (7) Ilog (2006). ILOG CPLEX 10.0: user's manual. France, 478 p.
- (8) Imai, A.; Nagaiwa, K. & Chan, W.T. (1997). Efficient planning of berth allocation for container terminals in Asia. *Journal of Advanced Transportation*, **31**, 75-94.
- (9) Imai, A.; Nishimura, E. & Papadimitriou, S. (2001). The dynamic berth allocation problem for a container port. *Transportation Research Part B*, **35**, 401-417.

- (10) Imai, A.; Nishimura, E. & Papadimitriou, S. (2003). Berth allocation with service priority. *Transportation Research Part B*, **37**, 437-457.
- (11) Imai, A.; Nishimura, E. & Papadimitriou, S. (2008). Berthing ships at a multi-user container terminal with a limited quay capacity. *Transportation Research Part E*, **44**, 136-151.
- (12) Lee, D.; Wang, H.Q. & Miao, L. (2008). Quay crane scheduling with non-interference constraints in port container terminals. *Transportation Research Part E*, **44**, 124-135.
- (13) Legato, P.; Monaco, F. & Tigani, N. (2001). Berth planning at Gioia Tauro's maritime terminal by logistic distribution models. *AIRO Annual Conference*, Cagliari, Italy.
- (14) Lorena, L.A.N. & Furtado, J.C. (2001). Constructive genetic algorithm for clustering problems. *Evolutionary Computation*, **9**, 309-327.
- (15) Mauri, G.R. & Lorena, L.A.N. (2004). Método iterativo para resolução do problema de escalonamento de tripulações. *XXXVI SBPO – Simpósio Brasileiro de Pesquisa Operacional*, São João del Rei.
- (16) Mauri, G.R. & Lorena, L.A.N. (2006). Simulated annealing aplicado a um modelo geral do problema de roteirização e programação de veículos. *XXXVIII SBPO – Simpósio Brasileiro de Pesquisa Operacional*. Goiânia.
- (17) Mauri, G.R. & Lorena, L.A.N. (2007). A new hybrid heuristic for driver scheduling. *International Journal of Hybrid Intelligent Systems*, **4**, 39-47.
- (18) Oliveira, A.C.M. (2002). Treinamento populacional em heurísticas – aplicações em otimização. Proposta de tese de Doutorado. Instituto Nacional de Pesquisas Espaciais (INPE).
- (19) Oliveira, A.C.M. & Lorena, L.A.N. (2002). 2-opt population training for minimization of open stack problem. **In:** *Advances in Artificial Intelligence* [edited by G. Bittencourt and G.L. Ramalho], *Springer Lecture Notes in Artificial Intelligence Series*, **2507**, 313-323.
- (20) Pinedo, M. (1995). *Scheduling: theory, algorithms and systems*. Prentice-Hall, Englewood Cliffs, NJ.
- (21) Steenken, D.; Voss, S. & Stahlbock, R. (2004). Container terminal operation and operations research – a classification and literature review. *OR Spectrum*, **26**, 3-49.
- (22) Vis, I.F.A. & Koster, R.D. (2003). Transshipment of containers at a container terminal: an overview. *European Journal of Operational Research*, **147**, 1-16.
- (23) Wilhelm, W.E. (2001). A technical review of column generation in integer programming. *Optimization and Engineering*, **2**, 159-200.