

## **CLUSTERING SEARCH APLICADO AO PROBLEMA DE ALOCAÇÃO DE BERÇOS**

**Rudinei Martins de Oliveira**

Instituto Nacional de Pesquisas Espaciais - INPE  
rudmart@gmail.com

**Geraldo Regis Mauri**

Universidade Federal do Espírito Santo - UFES  
mauri@cca.ufes.br

**Luiz Antonio Nogueira Lorena**

Instituto Nacional de Pesquisas Espaciais - INPE  
lorena@lac.inpe.br

### **RESUMO**

Este trabalho apresenta uma nova abordagem para o Problema de Alocação de Berços (PAB) de navios em portos. Devido à crescente demanda de navios que transportam containeres, o PAB pode ser considerado como um dos principais problemas de otimização em terminais marítimos. Neste trabalho, o PAB é tratado como dinâmico e modelado como discreto e, nesse contexto, é proposta uma nova alternativa para resolvê-lo. Essa alternativa é baseada na aplicação do método *Clustering Search* (CS), utilizando o *Simulated Annealing* (SA) como gerador de soluções. O CS é um método iterativo que divide o espaço de busca em *clusters* e é composto por uma metaheurística geradora de soluções, um processo de agrupamento e uma heurística de busca local. Os resultados obtidos são comparados a métodos recentes encontrados na literatura.

**PALAVRAS CHAVES.** Alocação de Berços. *Clustering Search*. *Simulated Annealing*. MH - Metaheurísticas.

### **ABSTRACT**

This work presents a new approach to the Berth Allocation Problem (BAP) for ships in ports. Due to the increasing demand for ships carrying containers, the PAB can be considered as a major optimization problem in marine terminals. In this paper, the PAB is considered as dynamic and modeled in discrete case and we propose a new alternative to solve it. The proposed alternative is based on applying the *Clustering Search* (CS) method using the *Simulated Annealing* (SA) for solutions generation. The CS is an iterative method which divides the search space in clusters and it is composed of a metaheuristic for solutions generation, a grouping process and a local search heuristic. The computational results are compared against recent methods found in the literature.

**KEYWORDS.** Berth Allocation. *Clustering Search*. *Simulated Annealing*. MH - Metaheuristics.

## 1. Introdução

O comércio internacional tem como pilar central o transporte marítimo. Aproximadamente 80% do comércio global de mercadorias é realizado através do mar (Buhrkal *et al.*, 2009). Além disso, em 2008 a frota de navios que transportam containeres teve um aumento em sua capacidade de 17,3 milhões de toneladas, ou 11,9%, e passaram a representar 13,6% do total mundial. No início de 2009, a frota mercante mundial atingiu 1,19 milhões, um crescimento de 6,7% em comparação a janeiro de 2008 e, desde o início da década, a quantidade de containeres aumentou em 154% (UNCTAD, 2009).

Desse modo, devido ao intenso fluxo de navios e containeres nos portos, estes são forçados a investir pesadamente para acomodar os navios, aprofundando e alargando canais e construindo novas instalações de atracque, tudo para que o tempo de atendimento do navio seja o menor possível.

Assim, a busca por uma logística de acomodar e minimizar o tempo de espera e atendimento dos navios motivou o surgimento de um problema conhecido na literatura como Problema de Alocação de Berços (PAB). O PAB consiste em alocar navios a posições de atracque de forma que seja utilizado o máximo de espaço do cais minimizando o tempo de serviço. As decisões a serem tomadas dizem respeito à posição (onde?) e ao tempo (quando?) em que o navio deverá atracar (Imai *et al.*, 2001).

O PAB possui grande quantidade de restrições físicas, técnicas, entre outras. Isso faz com que seja possível modelá-lo de diferentes maneiras. Quanto aos aspectos espaciais dos berços, o PAB pode ser modelado como discreto, contínuo ou híbrido (Imai *et al.*, 2005). No caso discreto, o cais é dividido em vários berços e somente um navio é atendido de cada vez em cada berço, independente do seu tamanho. No caso contínuo, não há nenhuma divisão do cais e, dessa forma, os navios podem atracar em qualquer posição. Já no caso híbrido, como no caso discreto, o cais é dividido em berços, mas os navios grandes podem ocupar mais de uma posição, permitindo assim que navios pequenos compartilhem seu berço. Além disso, se for levado em conta a chegada dos navios, o problema pode ser tratado como estático ou dinâmico (Imai *et al.*, 2001). O caso estático assume que todos os navios já estão no porto para o atendimento dinâmico, o caso dinâmico permite aos navios chegarem a qualquer momento.

Em ambos os casos, busca-se por uma melhor distribuição do espaço minimizando o tempo total de permanência dos navios no porto. Nesse contexto, este trabalho apresenta uma nova alternativa para resolver o PAB. É proposta uma aplicação do método híbrido conhecido como *Clustering Search* - CS (Chaves, 2009), utilizando o *Simulated Annealing* como gerador de soluções. O CS proposto é comparado a métodos recentes encontrados na literatura, permitindo assim verificar sua eficiência na resolução do PAB.

O restante do artigo está organizado como segue. A Seção 2 apresenta uma breve revisão bibliográfica sobre o PAB. Na Seção 3 é apresentada uma formulação matemática existente, e uma relaxação dessa formulação, que é utilizada como base neste trabalho, é descrita na Seção 4. A Seção 5 apresenta de forma detalhada o CS proposto, e os resultados computacionais obtidos são apresentados na Seção 6. Por fim, as conclusões são resumidas na Seção 7.

## 2. Revisão bibliográfica

Os trabalhos iniciais acerca do PAB surgiram no final dos anos 80, quando Thurman (1989) propôs um modelo de otimização para o planejamento de berços para a estação naval *Norfolk* (EUA). A partir desse modelo, Brown *et al.* (1994) elaboraram um plano para minimizar os conflitos dos carregamentos nessa mesma estação. Os mesmos autores ainda apresentaram um planejamento de berços para submarinos em Brown *et al.* (1997). São raros os trabalhos

relacionados ao PAB até meados dos anos 90. Entretanto, tais trabalhos vêm ganhando foco, principalmente nos últimos 10 anos.

Imai *et al.* (2001) abordaram o PAB em sua forma dinâmica. Os autores apresentam um método baseado na relaxação lagrangiana do problema original. Dois anos depois, Imai *et al.* (2003) aprimoraram sua abordagem considerando diferentes prioridades de atendimento entre os navios. Além disso, os autores propuseram um Algoritmo Genético como método de solução.

Cordeau *et al.* (2005) propõem duas formulações e duas heurísticas baseadas na Busca Tabu para resolver o PAB. Os autores apresentam testes realizados para o porto de *Gioia Tauro* (Itália). Cheong *et al.* (2008) apresentam uma aplicação do método *Multiobjective Evolutionary Algorithm (MOEA)* para resolver o PAB. Giallombardo *et al.* (2010) apresentam um modelo de programação quadrática e um de programação linear para representar o PAB. Além disso, os autores utilizam uma Busca Tabu e uma técnica de programação matemática para resolver instâncias baseadas em dados reais.

No Brasil, Mauri *et al.* (2008a) propõem uma abordagem baseada na aplicação do *Simulated Annealing* para resolução do caso discreto do PAB. Os autores tratam o problema como um *Problema de Roteamento de Veículos com Múltiplas Garagens e Janelas de Tempo (PRVGMJT)*. Os resultados computacionais superam os obtidos pelo CPLEX e pela Busca Tabu proposta por Cordeau *et al.* (2005). Por fim, Mauri *et al.* (2008b) tratam o PAB com um método híbrido chamado ATP/PL, que utiliza o Algoritmo de Treinamento Populacional em conjunto com um modelo de Programação Linear por meio da técnica de Geração de Colunas. Os resultados obtidos superam os apresentados em Mauri *et al.* (2008a).

### 3. Formulação matemática para o PAB

Considerando o PAB em sua forma discreta e dinâmica, assim como descrita por Cordeau *et al.* (2005), o PAB pode ser modelado como um PRVGMJT (Cordeau *et al.*, 2001; Mauri *et al.*, 2008a), onde os navios são vistos como clientes e os berços como garagens. Dessa forma, existem  $m$  veículos, um para cada garagem. Cada veículo inicia e termina seu “tour” em sua garagem. Esses navios são modelados como vértices em um multi-grafo. Cada garagem é dividida em um vértice de origem e um de destino. Nos vértices inicial e final, as janelas de tempo correspondem ao período de funcionamento dos berços.

O PRVGMJT é especificado como um multi-grafo  $G^k = (V^k, A^k)$ ,  $\forall k \in M$ , onde  $V^k = N \cup \{o(k), d(k)\}$  e  $A^k \subseteq V^k \times V^k$ . Assim, utilizou-se a seguinte notação:  $N$ : 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  $i$ ;  $s_i^k$ : horário de abertura do berço  $k$ ;  $e_i^k$ : horário de fechamento do berço  $k$ ;  $b_i$ : horário de término da janela de tempo para o navio  $i$ ;  $v_i$ : valor do tempo de serviço do navio  $i$ ;  $x_{ij}^k \in \{0,1\}$ ,  $\forall k \in M$ ,  $\forall (i,j) \in A^k$ ,  $x_{ij}^k = 1$  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 em 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  $k$ ;  $M_{ij}^k = \max\{b_i + t_i^k - a_j, 0\}$ ,  $\forall k \in M$ ,  $\forall (i,j) \in N$ . A Figura 1 ilustra os intervalos de tempo utilizado por cada navio, além das diferentes variáveis utilizadas na formulação do PAB. Em seguida, é apresentada a formulação matemática para o PAB proposta por Cordeau *et al.* (2005).

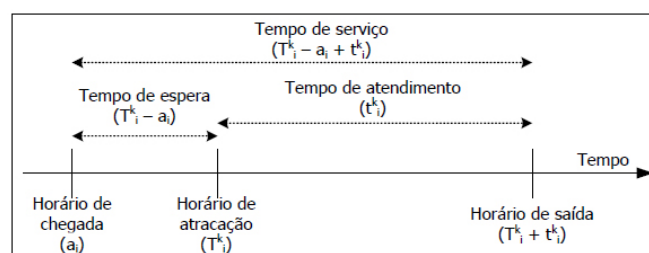


Figura 1 - Representação das variáveis de tempo (Mauri *et al.*, 2008a).

**Minimizar:**

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

**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 (1) minimiza a soma do tempo de serviço, ponderada por um custo associado  $v_i$ . A restrição (2) indica que cada navio é atendido por apenas um único berço. As restrições (3) e (4) asseguram que para cada berço  $k$  um navio será o primeiro e outro será o último a ser atendido. A restrição (5) garante a conservação do fluxo de atendimento para os navios restantes. A restrição (6) indica a consistência do horário de atracação dos navios. As restrições (7) e (8) garantem que o horário de atracação seja maior que o tempo de chegada e o horário de saída do navio seja menor que seu tempo limite de atendimento (janela de tempo). As restrições (9) e (10) garantem o tempo de disponibilidade do berço. Por fim, a restrição (11) define o domínio das variáveis de decisão.

## 4. Formulação relaxada para o PAB

Mauri *et al.* (2008a) propõem a relaxação das restrições (7), (8), (9) e (10), de tal forma que, as restrições (7) e (8) são transferidas para o termo (13) da função objetivo e as restrições (9) e (10) são inseridas no termo (14). Além disso, coeficientes de penalização ( $\omega = [\omega_0, \omega_1, \omega_2]$ ) são adicionados em cada termo da função. Dessa forma, tem-se a seguinte formulação:

**Minimizar:**

$$\omega_0 \sum_{i \in N} \sum_{k \in M} v_i \left( T_i^k - a_i + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{ij}^k \right) + \quad (12)$$

$$\omega_1 \sum_{i \in N} \sum_{k \in M} \left( \max(0, a_i - T_i^k) + \left( \max \left( 0, T_i^k + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{ij}^k - b_i \right) \right) \right) + \quad (13)$$

$$\omega_2 \sum_{k \in M} \left( \max(0, s^k - T_{o(k)}^k) + \max(0, T_{d(k)}^k + e^k) \right) \quad (14)$$

Sujeito a:

$$\sum_{k \in M} \sum_{j \in N \cup \{d(k)\}} x_{ij}^k = 1 \quad \forall i \in N \quad (15)$$

$$\sum_{j \in N \cup \{d(k)\}} x_{o(k)j}^k = 1 \quad \forall k \in M \quad (16)$$

$$\sum_{i \in N \cup \{o(k)\}} x_{i,d(k)}^k = 1 \quad \forall k \in M \quad (17)$$

$$\sum_{j \in N \cup \{d(k)\}} x_{i,j}^k - \sum_{j \in N \cup \{o(k)\}} x_{j,i}^k = 0 \quad \forall k \in M, \forall i \in N \quad (18)$$

$$T_i^k + t_i^k - T_j^k \leq (1 - x_{i,j}^k) M_{i,j}^k \quad \forall k \in M, \forall (i, j) \in A^k \quad (19)$$

$$x_{i,j}^k \in \{0,1\} \quad \forall k \in M, \forall (i, j) \in A^k \quad (20)$$

A partir dessa formulação, tem-se uma nova função objetivo (12), (13) e (14). Em cada termo dessa função há um fator de penalidade indicados pelos ômega. O termo (12) mantém na função objetivo o tempo de serviço junto com um custo associado. No termo (13) as violações das janelas de tempo dos navios são minimizadas. Por último, o termo (14) minimiza as violações nas janelas de tempo dos berços. Segundo Mauri *et al.* (2008a) com base nessa nova formulação do PAB e avaliando suas restrições, nota-se que apesar do problema ter sido modificado para um problema menos árduo (Problema de Roteamento de Veículos com Garagens Múltiplas SEM Janelas de Tempo) ele poderá apresentar as mesmas soluções do problema original (com janelas de tempo). Por outro lado, o modelo também poderá apresentar soluções inviáveis, mas essas inviabilidades são eliminadas por meio das penalizações inseridas no modelo.

## 5. CS proposto

Segundo Chaves (2009) o CS é um método iterativo que procura dividir o espaço de busca e localizar regiões promissoras por meio do enquadramento dessas em *clusters*. Um *cluster* pode ser definido por três atributos  $C = \{c, v, r\}$ . O centro  $c_i$  é uma solução que representa o *cluster*  $i$ , e identifica a sua localização dentro do espaço de busca.

O volume  $v_i$  é a quantidade de soluções agrupadas no *cluster*  $i$ . Um *cluster* se torna promissor quando o volume atingir um certo limitante  $\lambda$ . O índice de ineficácia  $r_i$  é uma variável de controle para identificar se a busca local está ou não melhorando o centro do *cluster*  $i$ . O valor de  $r_i$  indica o número de vezes consecutivas que a busca local foi aplicada no *cluster*  $i$  e não melhorou a solução. Esse atributo evita que a busca local seja executada em regiões ruins ou regiões que já tenham sido suficientemente exploradas por mais de  $r_{max}$  vezes.

O CS é formado basicamente por três componentes principais: uma metaheurística geradora de soluções, um processo de agrupamento e uma heurística de busca local. A cada iteração do CS, uma solução  $S$  é gerada pela metaheurística e enviada para o processo de agrupamento. Essa solução é então agrupada no *cluster* mais similar  $C_j$  e o centro desse *cluster*  $c_i$  é atualizado com informações contidas na nova solução agrupada, fazendo com que o centro se desloque no espaço de busca.

Em seguida, é analisado o volume  $v_j$  do *cluster*  $j$ , caso esse volume atinja um limitante  $\lambda$  ( $v_j \geq \lambda$ ), percebe-se que algum padrão de solução está sendo predominantemente gerado pela metaheurística. Portanto, esse *cluster* pode estar em uma região de busca promissora.

Por fim, é analisado o índice de ineficácia  $r_j$ , ou seja, caso a heurística de busca local não melhore a solução por  $r_{max}$  vezes consecutivas ( $r_j \geq r_{max}$ ), é aplicada uma perturbação aleatória no centro  $c_j$ , objetivando escapar de um possível ótimo local. Por outro lado, se  $r_j < r_{max}$ , a heurística de busca local é aplicada no centro  $c_j$  analisando a vizinhança do *cluster*. Encerrado esse

processo, retorna-se para a metaheurística que irá gerar uma nova solução. O critério de parada do CS é geralmente definido pela metaheurística escolhida. A Figura 2 apresenta o fluxograma de execução do CS. Mais detalhes sobre esse método são apresentados em Chaves (2009) e Oliveira (2004).

Para que as soluções possam ser agrupadas em *clusters* similares, deve-se definir uma forma para medir a distância entre duas soluções. Para isso, utiliza-se neste trabalho a distância de Hamming (Hamming, 1950), que é definida como sendo o número de arcos diferentes entre as soluções. Assim, seja  $d(i,j)$  uma função de medida de distância entre as soluções  $i$  e  $j$ , onde  $d(i,j)$  é um número positivo que deverá aumentar de acordo com o aumento da distância entre as duas soluções.

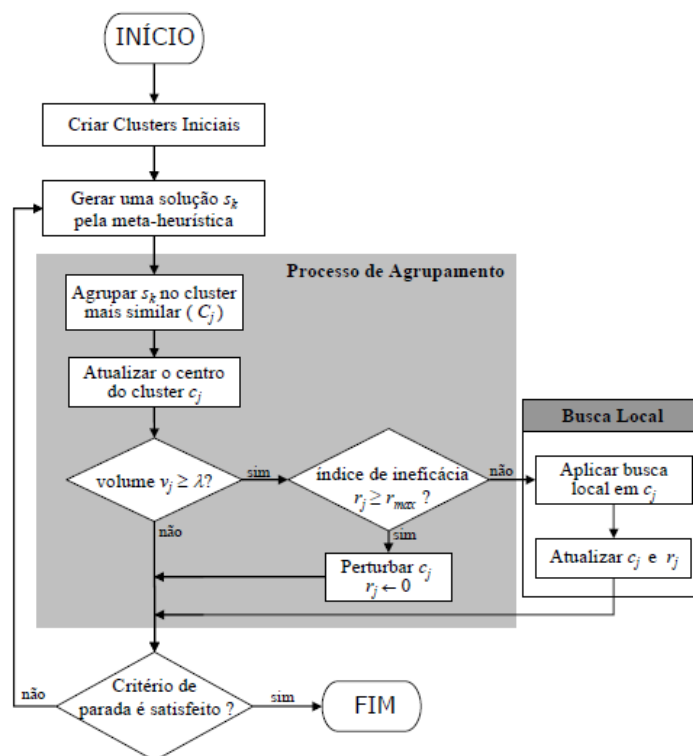


Figura 2 - Fluxograma do CS (Chaves, 2009).

Seguindo o fluxograma do CS (Figura 2), são criados então os *clusters* iniciais. Assim, para cada *cluster* é criada uma solução por meio das heurísticas de distribuição e programação apresentadas em Mauri *et al.* (2008a). Essas heurísticas são apresentadas nas Figuras 3 e 4, respectivamente.

1. CRIAR ( $m$  berços vazios);
2. CRIAR (uma lista  $L$  com todos os navios);
3. ORDENAR (a lista  $L$  pelo horário de chegada dos navios ao porto);
4. PARA (cada navio  $j$  em  $L$ ,  $j = 1, 2, \dots, n$ ) FAÇA
5.     SELECIONAR (um berço  $i$ ,  $i = 1, 2, \dots, m$ );
6.     SE (o berço  $i$  não puder atender ao navio  $j$ )
7.         VOLTAR (para o passo 5);
8.     SENÃO
9.         ATRIBUIR (o navio  $j$  ao berço  $i$ );
10.     FIM-SE;
11. FIM-PARA;

Figura 3 - Heurística de distribuição (Mauri *et al.*, 2008a).

A partir de então, o SA, baseado no proposto por Mauri *et al.* (2008a), é executado, e a cada temperatura, a solução corrente (não a melhor) é enviada ao CS. A Figura 5 apresenta um

pseudo-código do SA implementado. Pode-se notar que o CS é chamado na linha 20 desse algoritmo, ou seja, a cada temperatura.

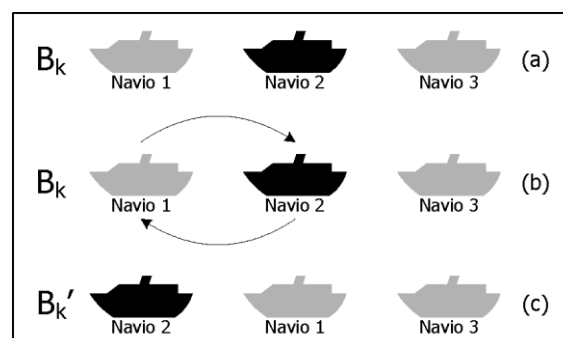
1. PARA (cada berço  $k$ ,  $k = 1, 2, \dots, m$ ) FAÇA
2.     PARA (cada navio  $i$  atribuído a  $k$ ) FAÇA
3.          $T_i^k = \begin{cases} \max(a_i, s^k), & i = 1 \\ \max(a_i, T_{i-1}^k + t_{i-1}^k), & i > 1 \end{cases}$
4.     FIM-PARA;
5.     FIM-PARA;
6.     CALCULAR (a função objetivo para a solução atual);

**Figura 4** - Heurística de programação (Mauri *et al.*, 2008a).

1. DADO ( $\alpha$ ,  $S_{\max}$ ,  $T_0$  e  $T_c$ ) FAÇA
2.     GERAR (uma solução  $S$  por meio da heurística de distribuição);
3.     AVALIAR (a solução  $S$  por meio da heurística de programação);
4.      $S^* \leftarrow S$ ;                     {Melhor solução obtida até então}
5.      $\text{IterT} \leftarrow 0$ ;                 {Número de iterações na temperatura  $T$ }
6.      $T \leftarrow T_0$ ;                 {Temperatura corrente}
7.     ENQUANTO ( $T > T_c$ ) FAÇA
8.         ENQUANTO ( $\text{IterT} < S_{\max}$ ) FAÇA
9.              $\text{IterT} \leftarrow \text{IterT} + 1$ ;
10.            GERAR (um vizinho qualquer  $S'$  por meio de um dos mov. de troca);
11.            APLICAR (a heurística de programação em todos os berços de  $S'$ );
12.             $\Delta \leftarrow f(S') - f(S)$ ;
13.            SE ( $\Delta < 0$ )  $S \leftarrow S'$ ;
14.            SE ( $f(S') < f(S^*)$ )  $S^* \leftarrow S'$ ; FIM-SE;
15.            SENÃO
16.                 TOMAR ( $x \in [0, 1]$ );
17.                 SE ( $x < e^{-\Delta/T}$ )  $S \leftarrow S'$ ; FIM-SE;
18.            FIM-SE;
19.            FIM-ENQUANTO;
20.     EXECUTAR-CS (solução corrente  $S$ );
21.      $T \leftarrow \alpha * T$ ;  $\text{IterT} \leftarrow 0$ ;
22.     FIM-ENQUANTO;
23.      $S \leftarrow S^*$ ;
24.     RETORNAR ( $S$ );

**Figura 5** - Algoritmo *Simulated Annealing* utilizado no CS (Adaptado de Mauri *et al.*, 2008a).

Como estrutura de vizinhança no SA (linha 10) foram utilizados três diferentes movimentos de troca: *Reordenar navios*, *Realocar navio* e *Trocar navios* (Figuras 6, 7 e 8, respectivamente). Assim como na geração da solução inicial, esses movimentos garantem que cada navio seja atribuído apenas a berços que possam atendê-los.

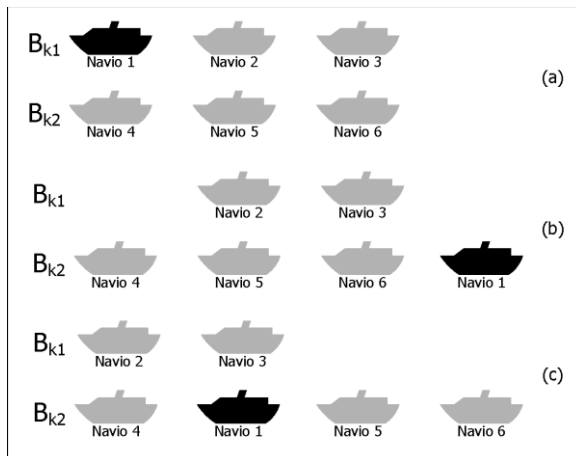
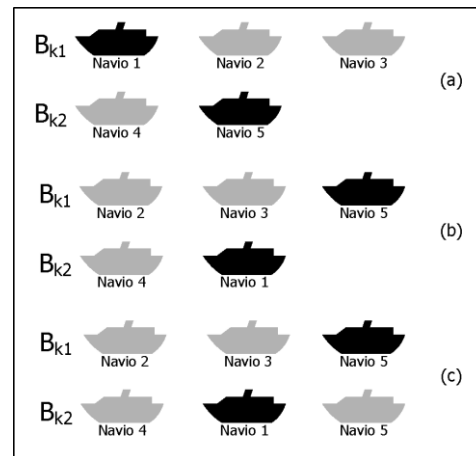




**Figura 6** - Movimento reordenar navios.

Após a execução de cada um dos movimentos de troca, a heurística de programação é aplicada para eliminar as sobreposições e recalculer o valor da função objetivo da nova solução. Esses movimentos são apresentados em Mauri *et al.* (2008a,b).

Cada solução vizinha no SA (linha 10) é gerada por apenas um desses movimentos, sendo a sua escolha feita de forma aleatória, porém uniformemente distribuída, possibilitando assim uma boa diversidade entre as soluções intermediárias geradas, e consequentemente uma boa exploração do espaço de soluções.

**Figura 7** - Movimento realocar navios.**Figura 8** - Movimento trocar navios.

O CS é implementado de acordo com o apresentado em Chaves (2009). Vale destacar que antes mesmo da execução do SA, as soluções “centro” de *clusters* já foram criadas (como descrito anteriormente). Ao final da execução do SA, ou seja, do método CS-SA como um todo, a melhor solução encontrada é tomada como solução final para o problema. O algoritmo EXECUTAR-CS é apresentado na Figura 9.

1. DADA (uma solução  $S$ );
2. ENCONTRAR (o *cluster*  $C_j$  mais similar a  $S$ );
3.  $v_j \leftarrow v_j + 1$ ;
4. ATUALIZAR (o centro do *cluster*  $C_j$ );
5. SE  $v_j \geq \lambda$  ENTÃO
6.      $v_j \leftarrow 1$ ;
7.     SE  $r_j \geq r_{max}$  ENTÃO
8.         APLICAR (perturbação em  $c_j$ )
9.          $r_j \leftarrow 0$ ;
10.     SENÃO
11.         APLICAR (busca local – encontrar  $c_j'$ );
12.         SE  $f(c_j') < f(c_j)$  ENTÃO
13.              $c_j \leftarrow c_j'$ ;
14.              $r_j \leftarrow 0$ ;
15.         SENÃO
16.              $r_j \leftarrow r_j + 1$ ;
17.         FIM-SE;
18.     FIM-SE;
19. FIM-SE;

**Figura 9** - Algoritmo EXECUTAR-CS.

Como mencionado anteriormente, a determinação do *cluster* mais similar (linha 2) é dada pela menor distância de Hamming (Hamming, 1950). A atualização do centro do *cluster* (linha 4)



é dada pela execução do *Path-Relinking* entre a solução dada  $S$  e a solução  $c_j$  centro do cluster  $C_j$  (Figura 10).

A idéia desse algoritmo é simples, e consiste em executar os movimentos necessários para “transformar” a solução  $S'$  (cópia de  $S$ ) na solução  $c_j$ . A partir destes movimentos, a melhor solução encontrada é tomada como novo centro do cluster  $C_j$ .

1.	<u>DADO</u> ( $S'$ e $c_j$ )
2.	<u>PARA</u> (cada navio $i$ , $i = 1, \dots, n$ ) <u>FAÇA</u>
3.	<u>SE</u> (berço que atende o navio $i$ em $S' \neq$ berço que atende o navio $i$ em $c_j$ );
4.	<u>REMOVER</u> (o navio $i$ de seu respectivo berço em $S'$ );
5.	<u>INSERIR</u> (o navio $i$ no berço em $S'$ correspondente ao berço em $c_j$ );
6.	<u>SE</u> ( $f(S') < f(c_j)$ ) <u>ENTÃO</u>
7.	$c_j \leftarrow S'$ ;
8.	<u>SENÃO</u>
9.	<u>SE</u> ( $f(S') = f(c_j)$ ) <u>ENTÃO</u>
10.	<u>PARE</u> ;
11.	<u>FIM-SE</u> ;
12.	<u>FIM-SE</u> ;
13.	<u>FIM-PARA</u> ;

**Figura 10** - *Path-Relinking* utilizado na atualização dos centros de *clusters*.

A perturbação apresentada na linha 8 do EXECUTAR-CS (Figura 9) é dada por uma simples aplicação do movimento trocar navios (Figura 8). Por fim, a busca local (linha 11 do EXECUTAR-CS – Figura 9) utilizada para intensificar a busca em *clusters* promissores é apresentada na Figura 11. É interessante destacar que a busca local é aplicada por berço, evitando assim um alto tempo de processamento.

1.	<u>PARA</u> (cada berço pertencente a $c_j$ ) <u>FAÇA</u>
2.	<u>ENQUANTO</u> (melhorar a solução) <u>FAÇA</u>
3.	<u>PARA</u> (todos os navios de $c_j$ ) <u>FAÇA</u>
4.	<u>INSERIR</u> (o navio $p$ em todas as posições do berço);
5.	<u>ARMAZENAR</u> (os navios e as posições que resultam na melhor solução);
6.	<u>FIM-PARA</u> ;
7.	<u>INSERIR</u> (os navios nas melhores posições armazenadas);
8.	<u>FIM-ENQUANTO</u> ;
9.	<u>FIM-PARA</u> ;

**Figura 11** - Busca local.

## 6. Experimentos computacionais

Foram utilizadas 30 instâncias distintas, cada uma com 60 navios e 13 berços. Essas instâncias são baseadas em dados do porto de *Gioia Tauro* (Itália), e foram geradas aleatoriamente por Cordeau *et al.* (2005). Todos os experimentos foram realizados em um PC com processador *AMD Athlon™* 64 de 2.2 GHz e 1GB de memória RAM (mesma máquina utilizada por Mauri *et al.* 2008a,b). Toda a implementação foi desenvolvida na linguagem C++.

Os parâmetros utilizados pelo CS, em todos os experimentos, foram  $T_0 = 20000$ ,  $\alpha = 0.975$ ,  $T_C = 0.01$ ,  $S_{Amax} = 1000$ ,  $\lambda = 7$ ,  $r_{max} = 3$  e o número de *clusters* foi igual 10. As penalizações utilizadas em ambos os casos foram  $\omega = [1, 10, 10]$ .

Foram realizados 5 testes aleatórios para cada instância. Nesses testes, o CS utilizou um tempo computacional médio de 12,79 segundos por instância. Além disso, o desvio entre a média das soluções obtidas em relação a melhor solução obtida para cada instância, foi de 0,04%.

**Tabela 1** - Comparação dos resultados obtidos (CS) com outros métodos encontrados na literatura.

Inst.	CPLEX <sup>1</sup>		TS <sup>2</sup>		SA+RA <sup>3</sup>		PTA/LP <sup>4</sup>		GSPP <sup>5</sup>		CS		MELHORAS (%)				
	FO	Tempo	FO	Tempo	FO	Tempo	FO	Tempo	FO	Tempo	FO	Tempo	CPLEX	TS	SA+RA	PTA/LP	GSPP
i01	-	3600,00	1415	-	1409	53,12	1409	74,61	1409	17,92	1409	12,47	-	<b>0,42</b>	0,00	0,00	0,00
i02	2606	3600,00	1263	-	1261	58,94	1261	60,75	1261	15,77	1261	12,59	<b>51,61</b>	<b>0,16</b>	0,00	0,00	0,00
i03	2565	3600,00	1139	-	1129	54,03	1129	135,45	1129	13,54	1129	12,64	<b>55,98</b>	<b>0,88</b>	0,00	0,00	0,00
i04	4353	3600,00	1303	-	1302	67,33	1302	110,17	1302	14,48	1302	12,59	<b>70,09</b>	<b>0,08</b>	0,00	0,00	0,00
i05	2672	3600,00	1208	-	1207	55,38	1207	124,70	1207	17,21	1207	12,68	<b>54,83</b>	<b>0,08</b>	0,00	0,00	0,00
i06	-	3600,00	1262	-	1261	53,88	1261	78,34	1261	13,85	1261	12,56	-	<b>0,08</b>	0,00	0,00	0,00
i07	2887	3600,00	1279	-	1279	60,52	1279	114,20	1279	14,60	1279	12,63	<b>55,70</b>	0,00	0,00	0,00	0,00
i08	5177	3600,00	1299	-	1299	61,45	1299	57,06	1299	14,21	1299	12,57	<b>74,91</b>	0,00	0,00	0,00	0,00
i09	-	3600,00	1444	-	1444	57,91	1444	96,47	1444	16,51	1444	12,58	-	0,00	0,00	0,00	0,00
i10	-	3600,00	1213	-	1213	68,95	1213	99,41	1213	14,16	1213	12,61	-	0,00	0,00	0,00	0,00
i11	-	3600,00	1378	-	1368	76,77	1369	99,34	1368	14,13	1368	12,58	-	<b>0,73</b>	0,00	<b>0,07</b>	0,00
i12	3206	3600,00	1325	-	1325	62,84	1325	80,69	1325	15,60	1325	12,56	<b>58,67</b>	0,00	0,00	0,00	0,00
i13	-	3600,00	1360	-	1360	68,19	1360	89,94	1360	13,87	1360	12,61	-	0,00	0,00	0,00	0,00
i14	-	3600,00	1233	-	1233	75,06	1233	73,95	1233	15,60	1233	12,67	-	0,00	0,00	0,00	0,00
i15	4672	3600,00	1295	-	1295	54,55	1295	74,19	1295	13,52	1295	13,80	<b>72,28</b>	0,00	0,00	0,00	0,00
i16	4320	3600,00	1375	-	1364	63,91	1365	170,36	1364	13,68	1364	14,46	<b>68,43</b>	<b>0,80</b>	0,00	<b>0,07</b>	0,00
i17	-	3600,00	1283	-	1283	56,28	1283	46,58	1283	13,37	1283	13,73	-	0,00	0,00	0,00	0,00
i18	3681	3600,00	1346	-	1345	53,98	1345	84,02	1345	13,51	1345	12,72	<b>63,46</b>	<b>0,07</b>	0,00	0,00	0,00
i19	2400	3600,00	1370	-	1370	52,83	1367	123,19	1367	14,59	1367	13,39	<b>43,04</b>	<b>0,22</b>	<b>0,22</b>	0,00	0,00
i20	-	3600,00	1328	-	1328	53,38	1328	82,30	1328	16,64	1328	12,82	-	0,00	0,00	0,00	0,00
i21	-	3600,00	1346	-	1341	53,52	1341	108,08	1341	13,37	1341	12,68	-	<b>0,37</b>	0,00	0,00	0,00
i22	3489	3600,00	1332	-	1326	57,97	1326	105,38	1326	15,24	1326	12,62	<b>61,99</b>	<b>0,45</b>	0,00	0,00	0,00
i23	-	3600,00	1266	-	1266	53,75	1266	43,72	1266	13,65	1266	12,62	-	0,00	0,00	0,00	0,00
i24	4867	3600,00	1261	-	1260	54,09	1260	78,91	1260	15,58	1260	12,64	<b>74,11</b>	<b>0,08</b>	0,00	0,00	0,00
i25	1993	3600,00	1379	-	1377	53,56	1376	96,58	1376	15,80	1376	12,62	<b>30,96</b>	<b>0,22</b>	<b>0,07</b>	0,00	0,00
i26	2520	3600,00	1330	-	1318	57,34	1318	101,11	1318	15,38	1318	12,62	<b>47,70</b>	<b>0,90</b>	0,00	0,00	0,00
i27	3209	3600,00	1261	-	1261	69,98	1261	82,86	1261	15,52	1261	12,64	<b>60,70</b>	0,00	0,00	0,00	0,00
i28	-	3600,00	1365	-	1360	58,47	1360	52,91	1359	16,22	1359	12,71	-	<b>0,44</b>	<b>0,07</b>	<b>0,07</b>	0,00
i29	4809	3600,00	1282	-	1280	69,09	1280	203,36	1280	15,30	1280	12,62	<b>73,38</b>	<b>0,16</b>	0,00	0,00	0,00
i30	-	3600,00	1351	-	1344	70,67	1344	71,02	1344	16,52	1344	12,58	-	<b>0,52</b>	0,00	0,00	0,00
<b>Média</b>	<b>3600,00</b>		<b>120,00</b>		<b>60,26</b>		<b>93,99</b>		<b>14,98</b>		<b>12,79</b>						

<sup>1</sup>Ilog (2006)<sup>3</sup>Mauri *et al.* (2008a)<sup>5</sup>Buhrkal *et al.* (2009)<sup>2</sup>Cordeau *et al.* (2005)<sup>4</sup>Mauri *et al.* (2008b)

A Tabela 1 apresenta uma comparação direta dos resultados obtidos com os apresentados nos trabalhos mais recentes encontrados na literatura. Nessa tabela, são apresentadas as melhores soluções obtidas pelo CS e o tempo médio (considerando os 5 testes para cada instância). As colunas *FO* (função objetivo) indicam os valores das soluções obtidas. Já as últimas 5 colunas indicam as melhoras nas soluções obtidas pelo CS proposto em relação aos demais métodos apresentados nessa mesma tabela. Os tempos computacionais apresentados estão em segundos. De forma geral, o CS apresentou os melhores resultados (ótimos – iguais aos apresentados por Buhrkal *et al.*, 2009) para todas as instâncias.

A Busca Tabu ( $TS^2$ ) proposta por Cordeau *et al.* (2005) utilizou em média 120 segundos para resolver cada instância, segundo os próprios autores, e a máquina utilizada não foi especificada. Já a máquina utilizada pelo *Simulated Annealing* com *Re-annealing* ( $SA+RA^3$ ) e pela geração de colunas ( $PTA/LP^4$ ) foi a mesma usada pelo CS proposto. Por fim, a máquina utilizada pelo modelo GSPP<sup>5</sup> foi um PC com processador *Intel Xeon* de 2.66 GHz, ou seja, uma máquina bem superior às demais.

O CS proposto encontrou as soluções ótimas (provadas por Buhrkal *et al.*, 2009) em todos os casos e, além disso, as melhores soluções foram obtidas em um tempo computacional inferior aos demais métodos.

## 7. Conclusões

Este trabalho teve por finalidade estudar o Problema de Alocação de Berços (PAB). Dentro desse contexto, buscou-se contribuir no aprimoramento de uma logística na distribuição do espaço do cais minimizando o tempo total de serviço dos navios, evitando assim prejuízos para o porto com embarcações rejeitadas.

Para resolver o PAB, foi proposta uma aplicação do método híbrido *Clustering Search* (CS) utilizando o *Simulated Annealing* como gerador de soluções. O CS mostrou ser adequado e eficiente na localização de regiões promissoras por meio do enquadramento dessas em *clusters*. Dessa forma, percebe-se que o CS atuou como uma alternativa para acelerar a obtenção de boas soluções. Esse fato fica evidente quando os resultados obtidos são comparados diretamente com o  $SA+RA$  proposto por Mauri *et al.* (2008a).

De uma forma geral, os resultados obtidos demonstram que o CS foi capaz de gerar soluções de boa qualidade para todas as instâncias em tempos computacionais expressivamente baixos. Esses resultados ainda foram comparados com outras abordagens recentes encontradas na literatura, e em todos os casos, as soluções obtidas foram melhores.

**Agradecimentos:** Os autores agradecem à FAPES (processo 45391998/09), ao CNPq (processo 471837/2008-3) e a CAPES pelo apoio financeiro.

## Referências

Bierwirth, C. e Meisel, F. (2010), A survey of berth allocation and quay crane scheduling problems in container terminals, *European Journal of Operational Research*, 202(3), 615-627.

Brown, G. G., Lawphongpanich, S. e Thurman, K. P. (1994), Optimizing ship berthing, *Naval Research Logistics*, 41, 1-15.

Brown, G. G., Cormican, K. J., Lawphongpanich, S. e Widdis, D. B. (1997), Optimizing submarine berthing with a persistence incentive, *Naval Research Logistics*, 44, 301-318.

Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J. e Lusby, R., Models for the discrete berth allocation problem: a computational comparison. *Technical Report 14/2009 - Technical University of Denmark*, 2009.

**Chaves, A. A.**, Metaheurísticas híbridas com busca por agrupamentos para problemas de otimização combinatória. *Tese* (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2009.

**Cheong, C. Y., Tan, K. C., Liu, D. K. e Lin, C. J.** (2008), Multi-objective and prioritized berth allocation in container ports. *Annals of Operations Research*, [in press].

**Cordeau, J. F., Laporte, G. e Mercier, A.** (2001), A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52 (8), 928-936.

**Cordeau, J. F., Laporte, G., Legato, P. e Moccia, L.** (2005), Models and tabu search heuristics for the berth allocation problem, *Transportation Science*, 39, 526-538.

**Giallombardo, G., Moccia, L., Salani, M. e Vacca, I.** (2010), Modeling and solving the tactical berth allocation problem. *Transportation Research Part B*, 44 (2), 232-245.

**Hamming, R. W.** (1950), Error detecting and error correcting codes. *Bell System Technical Journal*, 26(2), 147-160.

**Ilog.** *ILOG CPLEX 10.0: user's manual*. France: [s.n.], 478 p, 2006.

**Imai, A., Nishimura, E. e Papadimitriou, S.** (2001), The dynamic berth allocation problem for a container port, *Transportation Research Part B*, 35, 401- 417.

**Imai, A., Nishimura, E. e Papadimitriou, S.** (2003), Berth allocation with service priority, *Transportation Research Part B*, 37, 437-457.

**Imai, A., Sun, X., Nishimura, E. e Papadimitriou, S.** (2005), Berth allocation in a container port: using a continuous location space approach. *Transportation Research Part B*, v. 39, n. 3, p. 199-221.

**Mauri, G. R., Oliveira, A. C. M. e Lorena, L. A. N.** (2008a), Heurística baseada no simulated annealing aplicada ao problema de alocação de berços, *GEPROS - Gestão da Produção, Operações e Sistemas*, 1(1), 113-127.

**Mauri, G. R., Oliveira, A. C. M. e Lorena, L. A. N.** (2008b), A hybrid column generation approach for the berth allocation problem, *Lecture Notes in Computer Science*, 4972, 110-122.

**Oliveira, A. C. M.** Algoritmos evolutivos híbridos com detecção de regiões promissoras em espaços de busca contínuo e discreto. *Tese* (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2004.

**Thurman, K. P.**, Optimal ship berthing plans. *Dissertação* (Masters of Science in Operations Research) - Naval Postgraduate School, Monterey, California - EUA, 1989.

**UNCTAD.** United nations conference on trade and development. *Review of maritime transport*, 2009.