

Heurística para solução do Problema de Alocação de Berços em Terminais Marítimos

Rudinei Martins de Oliveira^{1,*}, Geraldo Regis Mauri², Luiz Antonio Nogueira Lorena³

¹rudmart@gmail.com, INPE, Brasil

²mauri@cca.ufes.br, UFES, Brasil

³lorena@lac.inpe.br, INPE, Brasil

Abstract. *This work presents an 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 as discrete and continuous case and we propose a new alternative to solve it. The proposed alternative is based on applying the Clustering Search (CS) (Oliveira and Lorena, 2007) 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.*

Resumo. *Este trabalho apresenta uma 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 contínuo, nesse contexto, é proposta uma alternativa para resolvê-lo. Essa alternativa é baseada na aplicação do método Clustering Search (CS) (Oliveira e Lorena, 2007), 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-chave: *Alocação de Berços, Clustering Search, Simulated Annealing.*

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, 2010).

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 atraque, 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 atracação 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 e ao tempo em que o navio deverá atracar.

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 ou contínuo (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. Além disso, se for levado em conta a chegada dos navios, o problema pode ser tratado como estático ou dinâmico. O caso estático assume que todos os navios já estão no porto para o atendimento, 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 alternativa para resolver o PAB onde os berços são modelados como discretos e contínuos. Assim, é proposta uma aplicação do método híbrido conhecido como *Clustering Search – CS* (Oliveira e Lorena, 2007), 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. A Seção 3 apresenta de forma detalhada o CS proposto, e os resultados computacionais obtidos são apresentados na Seção 4. Por fim, as conclusões são resumidas na Seção 5.

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 base naval *Norfolk* (EUA). A partir desse modelo, Brown *et al.* (1994) elaboraram um plano para minimizar os conflitos dos carregamentos nessa mesma base. 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.* (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. Clustering Search (CS)

Segundo Chaves (2009) 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* e, caso esse volume atinja um limitante λ ($v_i \geq \lambda$), 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; Lorena (2007).

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.

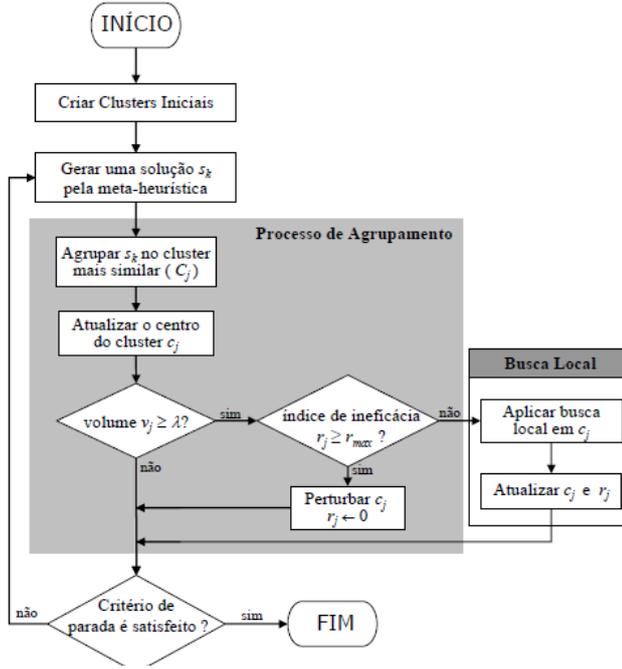


Figura 2 - Fluxograma do CS (Chaves, 2009).

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, programação (Mauri *et al.*, 2008a) e atualização para o caso contínuo do PAB (Mauri *et al.*, 2011). A função objetivo (Equação 1) presente na heurística de programação busca minimizar o tempo de serviço multiplicado por um custo associado (v_i), mais as violações das janelas de tempo dos navios e as violações nas janelas de tempo dos berços. Os ômeegas ($\omega_0, \omega_1, \omega_2$) em cada termo da expressão são os fatores de penalidades; a_i : horário de chegada do navio i ; b_i : horário de término da janela de tempo para o navio i ; t_i^k : duração do atendimento do navio i no berço k ; s^k : horário de abertura do berço k ; e^k : horário de fechamento do berço k ; $T_i^k \forall k \in M, i \in N$ 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 ; $d(k)$ é o último navio saiu do berço k ; $o(k)$ é o primeiro navio atracou no berço k ; v_i : valor (custo) do tempo de serviço do navio i ; $x_{ij}^k \in \{0, 1\}, \forall k \in M, x_{ij}^k = 1$ se o navio j é atendido pelo berço k após o navio i .

$$f(x) = \left\{ \begin{array}{l} \omega_0 \sum_{i \in N} \sum_{k \in M} v_i \left(T_i^k - a_i + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{ij}^k \right) + \\ \omega_1 \sum_{i \in N} \sum_{k \in M} \left(\max(0, a_i - T_i^k) + \left(\max \left(0, T_i^k + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{ij}^k - b_i \right) \right) \right) + \\ \omega_2 \sum_{k \in M} \left(\max(0, s^k - T_{o(k)}^k) + \max(0, T_{d(k)}^k + e^k) \right) \end{array} \right. \quad 1)$$

A partir de então, o SA, baseado no proposto por Mauri *et al.* (2008a), é executado, e a cada nível de temperatura, a solução corrente (não a melhor) é enviada ao CS. A Figura 3 apresenta um pseudo-código do SA implementado. Pode-se notar que o CS é chamado na linha 20 desse algoritmo, ou seja, a cada nível de temperatura.

XII Workshop de Computação Aplicada -WORCAP 2012

1. DADO (α , S_{Amax} , 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. $IterT \leftarrow 0$; {Número de iterações na temperatura T }
6. $T \leftarrow T_0$; {Temperatura corrente}
7. ENQUANTO ($T > T_c$) FAÇA
8. ENQUANTO ($IterT < S_{Amax}$) FAÇA
9. $IterT \leftarrow IterT + 1$;
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. $T \leftarrow \alpha * T$; $IterT \leftarrow 0$;
22. FIM-ENQUANTO;
23. $S \leftarrow S^*$;
24. RETORNAR (S);

Figura 3 - 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*. Esses movimentos são apresentados em Mauri *et al.* (2008a,b). Após a execução de cada um dos movimentos de troca, a heurística de programação é aplicada para eliminar as sobreposições e recalculando o valor da função objetivo da nova solução. 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. O CS é implementado de acordo com o apresentado em Chaves (2009) e Oliveira; Lorena (2007). Vale destacar que antes mesmo da execução do SA, as soluções “centro” de *clusters* já foram criadas. 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 4.

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 4 - Algoritmo EXECUTAR-CS.

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* (Chaves, 2009) entre a solução dada S e a solução c_j centro do

cluster C_j . 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 . A perturbação apresentada na linha 8 do EXECUTAR-CS (Figura 4) é dada por uma simples aplicação do movimento trocar navios. Por fim, a busca local (Chaves, 2009) (linha 11 do EXECUTAR-CS – Figura 4) utilizada para intensificar a busca em *clusters* promissores é executada. É interessante destacar que a busca local é aplicada por berço, evitando assim um alto tempo de processamento.

4. Experimentos computacionais

Foram utilizadas 10 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). Os experimentos para o caso discreto do PAB foram realizados em um *PC* com processador *AMD Athlon™ 64* de 2.2 GHz e 1GB de memória RAM, para o caso contínuo os experimentos foram realizados em um *PC* com processador *intelcore2duo* de 1.66GHz e 2 GB de memória RAM. Toda a implementação foi desenvolvida na linguagem C++. Os parâmetros utilizados pelo CS, em todos os experimentos, foram $T_0 = 20000$, $\alpha = 0.975$, $T_C = 0.01$, $SA_{max} = 1000$, $\lambda = 7$, $r_{max} = 3$ e o número de *clusters* foi igual 10. As penalizações utilizadas em ambos os casos foram $\omega = [1, 10, 10]$. A Tabela 1 apresenta uma comparação direta dos resultados obtidos com o algoritmo CS com trabalhos encontrados na literatura. Foram realizados 5 testes aleatórios para cada instância. Nesses testes, o CS utilizou um tempo computacional médio de 12,59 segundos por instância. 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 para todas as instâncias. A Busca Tabu (BT²) proposta por Cordeau *et al.* (2005) utilizou em média 120 segundos para resolver cada instância, segundo os próprios autores. 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.

A Tabela 2 mostra uma comparação entre o Algoritmo Memético (AM) e o CS para o caso contínuo do PAB. Nas três primeiras colunas encontram-se as instâncias utilizadas, o valor ótimo para o PAB-D (*OT DISC*) e a melhor solução conhecida (*MSC*) entre os três métodos (BT, AM e CS). Os resultados de cada método são apresentados nas colunas restantes (de 4 a 14), onde: *MS FO* é a melhor solução, *MA FO* e *MAMS* são as médias em 10 execuções do algoritmo para cada instância: da melhor solução e do tempo para achar a melhor solução (*MS FO*). Os tempos computacionais estão em segundos. O *DEV* e *DEV. MSC* são os desvios: da melhor solução (*MS FO*) em relação a média (*MA FO*) e desvio da melhor solução (*MS FO*) obtida pelo método no cabeçalho da coluna (BT, AM ou CS) e a melhor solução conhecida (*MSC*) dos três métodos em análise, onde $DEV = 100 \times (MA FO - MS FO) / MS FO$ e $DEV. MSC = 100 \times (MS FO - MS FO^*) / MS FO^*$, onde *MSFO** é a melhor solução obtida entre as três heurísticas. De forma geral, os valores de *FO* na Tabela 2 para o PAB-C são maiores que os do PAB-D, o que é correto, pois o caso discreto é uma relaxação do caso contínuo (Cordeau *et al.*, 2005). Quanto à comparação dos métodos, observa-se que em todas as instâncias o CS conseguiu os melhores resultados, além disso, comparando o CS com o AM observa-se que em média o CS foi mais rápido em encontrar as melhores soluções (*MA BST*) e obteve os menores valores para os desvios com médias totais de $DEV = 1,52\%$ e $DEV. MSC = 0.00\%$.

XII Workshop de Computação Aplicada -WORCAP 2012

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

Inst.	CPLEX ¹		BT ²		SA+RA ³		PTA/LP ⁴		GSPP ⁵		CS		MELHORAS (%)				
	FO	Tempo	FO	Tempo	FO	Tempo	FO	Tempo	FO	Tempo	FO	Tempo	CPLEX	BT	SA+RA	PTA/LP	GSPP
i01	-	3600,00	1415	-	1409	53,12	1409	74,61	1409	17,92	1409	12,47	-	0,42	0,00	0,00	0,00
i02	2606	3600,00	1263	-	1261	58,94	1261	60,75	1261	15,77	1261	12,59	51,61	0,16	0,00	0,00	0,00
i03	2565	3600,00	1139	-	1129	54,03	1129	135,45	1129	13,54	1129	12,64	55,98	0,88	0,00	0,00	0,00
i04	4353	3600,00	1303	-	1302	67,33	1302	110,17	1302	14,48	1302	12,59	70,09	0,08	0,00	0,00	0,00
i05	2672	3600,00	1208	-	1207	55,38	1207	124,70	1207	17,21	1207	12,68	54,83	0,08	0,00	0,00	0,00
i06	-	3600,00	1262	-	1261	53,88	1261	78,34	1261	13,85	1261	12,56	-	0,08	0,00	0,00	0,00
i07	2887	3600,00	1279	-	1279	60,52	1279	114,20	1279	14,60	1279	12,63	55,70	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	74,91	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
Média		3600,00		120,00		59,15		95,11		15,22		12,59					

¹Ilog (2006)

³Mauri *et al.* (2008a)

⁵Buhrkal *et al.* (2009)

²Cordeau *et al.* (2005)

⁴Mauri *et al.* (2008b)

Tabela 2: Comparação entre os resultados do AM, BT e CS.

Instância	OT DISC	MSC	BT		AM					CS				
			MS FO	DEV. MSC (%)	MS FO	MA FO	MA MS	DEV (%)	DEV. MSC (%)	MS FO	MA FO	MA BST	DEV (%)	DEV. MSC (%)
i01	1409	1583	1706	7,77	1613	1666,50	85,41	3,32	1,90	1583	1599,10	67,76	1,02	0,00
i02	1261	1315	1355	3,04	1326	1347,80	75,85	1,64	0,84	1315	1321,80	68,92	0,52	0,00
i03	1129	1207	1286	6,55	1234	1261,90	79,99	2,26	2,24	1207	1226,50	86,28	1,62	0,00
i04	1302	1380	1440	4,35	1392	1421,00	84,54	2,08	0,87	1380	1388,80	65,16	0,64	0,00
i05	1207	1262	1352	7,13	1285	1302,00	90,02	1,32	1,82	1262	1274,90	86,51	1,02	0,00
i06	1261	1413	1565	10,76	1461	1492,10	85,67	2,13	3,40	1413	1468,80	71,96	3,95	0,00
i07	1279	1324	1389	4,91	1333	1352,00	92,50	1,43	0,68	1324	1333,00	71,96	0,68	0,00
i08	1299	1381	1519	9,99	1425	1466,40	91,94	2,91	3,19	1381	1418,00	64,62	2,68	0,00
i09	1444	1633	1713	4,90	1651	1698,20	89,57	2,86	1,10	1633	1665,80	80,43	2,01	0,00
i10	1213	1333	1411	5,85	1371	1393,00	72,64	1,60	2,85	1333	1347,70	80,21	1,10	0,00
MA	1280,40	1383,10	1473,60	6,53	1409,10	1440,09	84,81	2,16	1,89	1383,10	1404,44	74,38	1,52	0,00

5. Conclusões

Este trabalho teve por finalidade estudar o problema de alocação de berços discreto e contínuo. 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. 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 300692/2009-9 e 300747/2010) e a CAPES.

Referências

- Brown, G. G., Lawphongpanich, S. e Thurman, K. P. (1994), Optimizing ship berthing, *Naval Research Logistics*, 41, 1-15.
- Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J., e Lusby, R. (2011). Models for the discrete berth allocation problem: a computational comparison. *Transportation Research Part E: Logistics and Transportation Review*, v. 47, p. 461–473.
- 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., Legato, P., and 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*, v. 44, p. 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: 478 p, 2006.
- Imai, A., Nishimura, E., e Papadimitriou, S. (2003). Berth allocation with service priority. *Transportation Research Part B: Methodological*, 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.
- Mauri, G.R., Andrade, L.N., Lorena, L.A.N. A memetic algorithm for a continuous case of the berth allocation problem. *ECTA 2011 – International Conference on Evolutionary Computation Theory and Applications*, Paris – França, 24 a 26 de outubro de 2011.
- Oliveira, A.C.M., Lorena, L.A.N.: Hybrid evolutionary algorithms and clustering search. *Studies in Computational Intelligence* 75, 77–99 (2007).
- Thurman, K. P., *Optimal ship berthing plans*. *Dissertação* (Masters of Science in Operations Research) - Naval Postgraduate School, Monterey, California - EUA, 1989.
- UNCTAD (2010). *Review of maritime transport*. United nations conference on trade and development.