

GRASP COM PATH RELINKING PARA O PROBLEMA DE ALOCAÇÃO DE BERÇOS

André Teixeira Lopes

Universidade Federal do Espírito Santo – UFES
andreteixeiralopes@hotmail.com

Vinicius Miranda Lopes Schulz

Universidade Federal do Espírito Santo - UFES
vinicius-schulz@hotmail.com

Geraldo Regis Mauri

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

Resumo

Atualmente, o Problema de Alocação de Berços (PAB) é considerado um dos principais problemas de otimização em terminais marítimos e apresenta uma grande quantidade de restrições físicas e técnicas. Neste trabalho, é proposta uma abordagem baseada na aplicação do método *Greedy Randomized Adaptive Search Procedure* (GRASP) de forma integrada com o método *Path Relinking* (PR) para resolução do PAB. Basicamente, a abordagem proposta consiste na aplicação do GRASP, para construção de soluções, com a aplicação do PR como uma estratégia de intensificação de busca. Para validação da abordagem proposta, foi utilizado um conjunto de instâncias baseado em dados reais e considerado em diversos trabalhos recentes. Os resultados computacionais obtidos demonstram que o GRASP com o PR foi capaz de gerar soluções de boa qualidade para todas as instâncias consideradas. Os resultados ainda foram comparados com outras abordagens recentes encontradas na literatura e, em todos os casos, as soluções obtidas foram iguais ou melhores.

Palavras-chave: GRASP; Path Relinking; Alocação de berços.

Abstract

Currently, the Berth Allocation Problem (BAP) is considered one of the main optimization problems on marine terminals, presenting several physical and technical constraints. In this paper, we propose an approach based on the Greedy Randomized Adaptive Search Procedure (GRASP) with the Path Relinking (PR) method to solve the BAP. Basically, the proposed approach consists on applying GRASP for building solutions using PR as a search intensifier. A set of instances based on real data and discussed in several recent works was used to validate the proposed approach. Computational results show that GRASP with PR was able to find good solutions for all of the used instances. The results were also compared against other recent approaches found in the literature and the solutions were equal or better in all of cases.

Keywords: GRASP; Path Relinking; Berth allocation.

1. Introdução

O transporte marítimo é um dos pilares fundamentais do crescimento econômico mundial ao longo de toda a história. Os serviços de transporte marítimo são essenciais para ajudar a economia e as empresas competirem em nível mundial. 80% do comércio mundial faz-se por via marítima (Comissão das Comunidades Europeias, 2009). O crescimento da economia mundial e do comércio internacional de mercadorias incentivou, nos últimos anos, a procura de serviços de transporte marítimo e, com isso, o número de contêineres, desde o início da década, subiu cerca de 154% (UNCTAD, 2009) e, segundo a Comissão das Comunidades Europeias (CCE, 2009), o número de navios deverá aumentar cerca de 29% até 2018.

Devido a este aumento de fluxo nos portos mundiais, surgiu uma maior dificuldade em gerenciar navios, o que motivou o surgimento de um problema conhecido na literatura como Problema de Alocação de Berço (PAB), que tem por finalidade estabelecer uma sequência de atendimento dos navios para cada berço disponível no cais, visando minimizar o tempo total de operação e espera dos navios dentro dos portos.

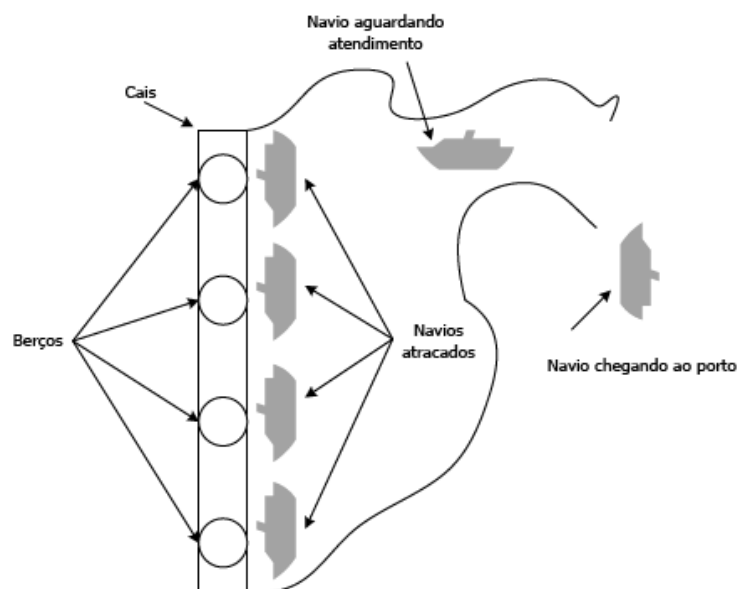


Figura 1 - Cenário típico do PAB (Mauri, 2008).

De acordo com Buhrkal *et al.* (2009), o PAB pode ser tratado como estático, onde todos os navios encontram-se no porto para o atendimento, ou dinâmico, que permite que os navios cheguem a qualquer instante. Neste trabalho, o PAB será abordado de modo dinâmico e discreto (o cais é dividido em vários berços que podem atender apenas um navio por vez, desconsiderando suas dimensões).

Para resolver o PAB, este trabalho propõe uma metaheurística híbrida que consiste em combinar os métodos *Greedy Randomized Adaptive Search Procedure* - GRASP (Feo & Resende, 1989) com *Path Relinking* - PR (Glover *et al.*, 2000). A abordagem proposta, GRASP+PR, consiste na aplicação do GRASP, para construção de soluções, com a aplicação do PR como uma estratégia de intensificação de busca.

Para validação da abordagem proposta, foi utilizado um conjunto de instâncias baseado em dados reais e considerado em diversos trabalhos recentes. Os resultados

obtidos demonstram que o GRASP+PR é capaz de gerar soluções competitivas em comparação a outros trabalhos recentes encontrados na literatura.

A Seção 2 apresenta uma breve revisão da literatura sobre o PAB. A Seção 3 apresenta o problema e sua formulação matemática. Já na Seção 4, é detalhado o método proposto para resolução do PAB. Na Seção 5 são apresentados os resultados computacionais e, por fim, na Seção 6 é feita uma breve conclusão sobre o trabalho.

2. Revisão da Literatura

O primeiro modelo de otimização para o PAB foi proposto por Thurman (1989). Brown *et al.* (1994) aprimoraram este modelo e minimizaram os conflitos entre os navios. Os autores propuseram ainda um planejamento de berços para submarinos em Brown *et al.* (1997).

Usando um método baseado na relaxação lagrangiana, Imai *et al.* (2001) abordaram o PAB em sua forma dinâmica e contínua, considerando as dimensões dos berços e navios. Imai *et al.* (2003) evoluíram sua abordagem inicial e consideraram prioridades no atendimento dos navios. Para gerar as soluções, os autores ainda propuseram um Algoritmo Genético.

Baseados na Busca Tabu, Cordeau *et al.* (2005) propuseram duas heurísticas e duas formulações matemáticas para o PAB. Cheong *et al.* (2008) apresentaram o método *Multiobjective Evolutionary Algorithm*. Já Giallombardo *et al.* (2010) utilizaram um modelo de programação quadrática, um de programação linear, uma Busca Tabu e uma técnica de programação matemática, usando instâncias reais do problema.

Mauri *et al.* (2008a) aplicaram o método *Simulated Annealing* para o caso discreto do PAB e Mauri *et al.* (2008b) propuseram uma metaheurística híbrida que utiliza o Algoritmo de Treinamento Populacional em conjunto com um modelo de Programação Linear por meio da técnica de Geração de Colunas. Por fim, Oliveira *et al.* (2010) utilizaram os métodos *Clustering Search* com *Simulated Annealing* para resolver o PAB

3. Descrição do Problema

O PAB consiste em selecionar os navios que chegam ao porto e atribuí-los a berços ao longo do cais. As principais decisões a serem tomadas neste processo envolvem a escolha de onde e quando os navios deverão atracar (Cordeau *et al.*, 2005). O PAB tem como objetivo minimizar o tempo de serviço de cada navio no porto. Esse tempo é dado pela soma do tempo de espera (tempo que o navio aguarda para ser atendido) e o tempo de atendimento:

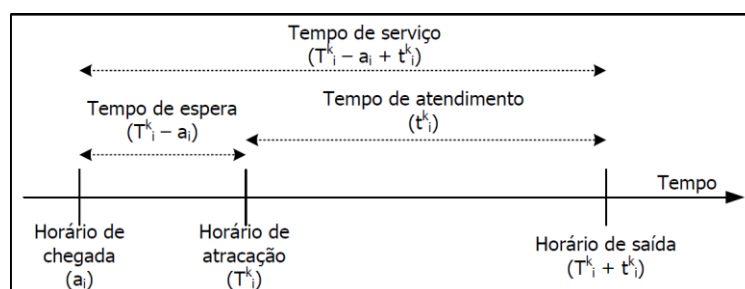


Figura 2 - Representação das variáveis de tempo (Mauri *et al.*, 2008a).

Considerando então os tempos definidos na Figura 2, pode-se definir matematicamente o PAB por meio da 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^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 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 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 ;
- $M_{ij}^k = \max\{b_i + t_i^k - a_i\}, \forall k \in M, \forall (i, j) \in N$.

Segundo Cordeau *et al.* (2005), o PAB pode ser descrito da seguinte forma: a função objetivo (1) minimiza a soma do tempo de serviço; 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”, ou seja, o atendimento para os demais navios; a restrição (6) faz o cálculo do horário de atracação dos navios, sendo considerados apenas os navios válidos para cada berço, ou seja, alguns navios não podem ser atendidos em determinados berços devido a restrições técnicas; 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; por fim, a restrição (11) garante que as variáveis de decisão sejam binárias. Mais detalhes sobre esse modelo são apresentados em Cordeau *et al.* (2005).

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:

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

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

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

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

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

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

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

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

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

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

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. Sendo assim, o tempo de serviço, juntamente com seu custo, é representado na expressão (12). A expressão (13) minimiza as violações nas janelas de tempo dos navios e a expressão (14) minimiza as violações nas janelas de tempo dos berços. Dessa forma, tem-se a seguinte formulação:

Minimizar:

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

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

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

Sujeito a:

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

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

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

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

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

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

Deve-se destacar que esse modelo (Equações 12 a 20) pode resultar em soluções inviáveis para o problema. Porém, essas inviabilidades são eliminadas durante a execução do GRASP+PR, por meio da penalização imposta.

4. Método de Resolução

Neste trabalho, o PAB foi tratado por meio do modelo relaxado proposto por Mauri *et al.* (2008a), descrito na seção anterior. Já para sua resolução, foi utilizado o *Greedy Randomized Adaptive Search Procedure* (GRASP) com o *Path Relinking* (PR) propostos por Feo e Resende (1989) e Glover (1996), respectivamente. Cada iteração do GRASP é composta por uma fase construtiva, na qual uma solução viável é construída, e por uma fase de busca local que, baseada na solução construída, aplica movimentos iterativos até que o ótimo local seja encontrado. Em seguida, o PR é aplicado como uma estratégia de intensificação, explorando as trajetórias que conectam a solução da iteração atual à melhor solução obtida até então.

Uma solução é representada como uma matriz na qual cada linha representa um berço, e as colunas a sequência de atendimento dos navios. Os elementos dessa matriz indicam o navio que deverá ser atendido. Para simplificar a implementação, cada linha da matriz apresenta um número diferente de colunas (número de navios atendidos pelo respectivo berço), e todos os navios são distribuídos nessa matriz. A Figura 3 apresenta uma possível solução para um problema hipotético com 4 berços (linhas) e 14 navios.

4	10	6	13	
2	9			
7	1	3	8	12
11	14	5		

Figura 3 - Representação de uma solução.

A cada iteração do GRASP, é gerada uma solução por meio de uma heurística de construção (Figura 5), na qual é aplicada uma busca local (Figura 6) e, a cada intervalo de iterações, é aplicado o PR. Para isso, são utilizados os seguintes parâmetros: número máximo de iterações do GRASP (*maxIter*), tamanho da busca local (*tamBL*) e intervalo de iterações para aplicação do PR (*iterPR*). A Figura 4 apresenta o algoritmo GRASP+PR proposto.

GRASP+PATH RELINKING (*maxIter*, *tamBL*, *iterPR*)

```

1.  $f^* \leftarrow \infty$ 
2.  $j \leftarrow 0$ 
3. PARA  $i \leftarrow 1$  ATÉ  $maxIter$  FAÇA
4.    $s \leftarrow$  HEURÍSTICA_DE_CONSTRUÇÃO( )
5.    $s \leftarrow$  BUSCA_LOCAL ( $s$ ,  $tamBL$ )
6.   SE  $f(s) < f^*$  ENTÃO
7.      $f^* \leftarrow f(s)$ 
8.      $s^* \leftarrow s$ 
9.   FIM-SE
10.  SE ( $j = iterPR$ ) ENTÃO
11.     $s^* \leftarrow$  PATH_RELINKING( $s$ ,  $s^*$ )
12.     $f^* \leftarrow f(s^*)$ 
13.     $j \leftarrow 0$ 
14.  FIM-SE
15.   $j \leftarrow j + 1$ 
16. FIM-PARA
17. RETORNA ( $s^*$ )

```

Figura 4 - Algoritmo GRASP+PR.

A heurística de construção (Figura 5) cria uma lista N de navios por ordem de chegada e uma lista B com todos os berços. A partir destas listas, os navios da lista N são atribuídos a um berço selecionado de forma aleatória, mas que seja capaz de atender ao navio em questão.

<u>HEURÍSTICA DE CONSTRUÇÃO ()</u>	
1.	CRIA (lista B com todos os berços vazios)
2.	CRIA (lista N de navios)
3.	ORDENA (lista N de navios por ordem de chegada)
4.	<u>PARA</u> (cada navio em N) <u>FAÇA</u>
5.	SELECIONA (um berço aleatório em B)
6.	ATRIBUI (o navio atual ao berço selecionado)
7.	<u>FIM-PARA</u>

Figura 5 - Heurística de Construção.

Após a criação de uma solução s , proveniente da fase de construção, é aplicada uma busca local (Figura 6) e, em cada uma de suas iterações, é aplicado um movimento de troca em s gerando uma solução vizinha s' . Os movimentos utilizados foram os mesmos propostos por Mauri *et al.* (2008b): *Re-ordenar Navios*, *Re-alocar Navio* e *Trocar Navios*. Assim, após $tamBL$ iterações, a melhor solução é retornada.

<u>BUSCA LOCAL ($s, tamBL$)</u>	
1.	<u>PARA</u> $i \leftarrow 1$ <u>ATÉ</u> $tamBL$ <u>FAÇA</u>
2.	GERAR (solução s' vizinha de s)
3.	<u>SE</u> $f(s') < f(s)$ <u>ENTÃO</u>
4.	$s \leftarrow s'$
5.	<u>FIM-SE</u>
6.	<u>FIM-PARA</u>
7.	<u>RETORNA</u> (s)

Figura 6 - Busca Local.

O movimento *Re-ordenar Navios* faz uma simples troca na sequência de atendimento de um berço, selecionado aleatoriamente. Ou seja, para um berço qualquer, dois navios são selecionados e suas posições na sequência de atendimento são trocadas.

Já o movimento de *Re-alocar Navio* consiste em selecionar dois berços quaisquer, pertencentes à solução s , selecionar um navio qualquer em um dos berços, extraí-lo de seu berço atual e atribuí-lo ao outro berço; caso o berço onde o navio será atribuído não possa atendê-lo, outro berço deverá ser selecionado. Após a atribuição, a lista de atendimento do novo berço deverá ser ordenada pelo horário de chegada dos navios.

Por fim, o movimento *Trocar Navios* consiste em selecionar dois berços quaisquer, pertencentes à solução s , selecionar um navio qualquer em cada um dos berços e trocá-los. Assim como no movimento de *Re-alocar navio*, caso os navios não possam ser atendidos pelos berços onde serão alocados, deverão ser selecionados outros navios e/ou berços. Após a troca, a lista de atendimento dos dois berços deverá ser ordenada pelo horário de chegada dos navios.

O PR funciona como uma estratégia para explorar as trajetórias que conectam a solução corrente com a melhor solução encontrada. Assim, realizadas *iterPR* iterações (ver Figura 4), a solução corrente s é progressivamente transformada na melhor solução s^* , ou seja, para cada navio, se os seus berços, em ambas as soluções forem diferentes,

então o navio da solução corrente s será remanejado para o mesmo berço (e posição de atendimento) em que é atendido na melhor solução s^* . O algoritmo referente ao *Path Relinking* implementado é apresentado na Figura 7.

PATH RELINKING (s, s^*)	
1.	<u>PARA</u> (cada navio) <u>FAÇA</u>
2.	OBTENHA (em s a posição do navio)
3.	OBTENHA (em s^* a posição do navio)
4.	<u>SE</u> (a posição do navio for diferente em s e s^*) <u>ENTÃO</u>
5.	TROQUE (a posição do navio em s para a mesma em s^*)
6.	<u>SE</u> $f(s) < f(s^*)$ <u>ENTÃO</u>
7.	$s^* \leftarrow s$
8.	<u>FIM-SE</u>
9.	<u>FIM-SE</u>
10.	<u>FIM-PARA</u>
11.	<u>RETORNA</u> (s^*)

Figura 7 - Path Relinking.

5. Resultados Computacionais

Para avaliar o desempenho do método proposto, foram utilizadas 30 instâncias distintas, cada uma com 60 navios e 13 berços. Essas instâncias foram geradas aleatoriamente por Cordeau *et al.* (2005). Os parâmetros utilizados pelo GRASP+PR foram: $maxIter = 140$, $tamBL = 3700$ e $iterPR = 5$, e os testes foram realizados em um PC com processador *AMD Athlon™* 64 de 2.2 GHz e 1GB de memória RAM.

Os trabalhos de Mauri *et al.* (2008a,b) e Oliveira *et al.* (2010) também utilizaram um PC com processador *AMD Athlon™* 64 de 2.2 GHz e 1GB de memória RAM, e Cordeau *et al.* (2005) utilizaram uma estação de trabalho SUN com processador de 900 MHz.

A função objetivo utilizada para avaliar as soluções foi semelhante à proposta por Mauri *et al.* (2008b), na qual soluções inviáveis são tratadas por meio de penalizações. Neste trabalho, os valores utilizados para as penalizações foram similares aos utilizados por Mauri *et al.* (2008b), ou seja, $\omega = [1,10,10]$.

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

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

Inst.	CPLEX ¹	TS ²	SA+RA ³		PTA/LP ⁴		CS ⁵		GRASP+PR		MELHORAS (%)					
	FO	FO	FO	Tempo (s)	FO	Tempo (s)	FO	Tempo (s)	FO	Tempo (s)	CPLEX	TS	SA+RA	PTA/LP	CS	
i01	-	1415	1409	53,12	1409	74,61	1409	12,47	1409	15,45	-	0,42	0	0	0	
i02	2606	1263	1261	58,94	1261	60,75	1261	12,59	1261	15,81	51,61	0,16	0	0	0	
i03	2565	1139	1129	54,03	1129	135,45	1129	12,64	1129	15,37	55,98	0,88	0	0	0	
i04	4353	1303	1302	67,33	1302	110,17	1302	12,59	1302	15,38	70,09	0,08	0	0	0	
i05	2672	1208	1207	55,38	1207	124,70	1207	12,68	1207	15,45	54,83	0,08	0	0	0	
i06	-	1262	1261	53,88	1261	78,34	1261	12,56	1261	15,64	-	0,08	0	0	0	
i07	2887	1279	1279	60,52	1279	114,20	1279	12,63	1279	15,67	55,70	0	0	0	0	
i08	5177	1299	1299	61,45	1299	57,06	1299	12,57	1299	15,75	74,91	0	0	0	0	
i09	-	1444	1444	57,91	1444	96,47	1444	12,58	1444	15,30	-	0	0	0	0	
i10	-	1213	1213	68,95	1213	99,41	1213	12,61	1213	15,47	-	0	0	0	0	
i11	-	1378	1368	76,77	1369	99,34	1368	12,58	1368	15,24	-	0,73	0	0,07	0	
i12	3206	1325	1325	62,84	1325	80,69	1325	12,56	1325	15,55	58,67	0	0	0	0	
i13	-	1360	1360	68,19	1360	89,94	1360	12,61	1360	15,45	-	0	0	0	0	
i14	-	1233	1233	75,06	1233	73,95	1233	12,67	1233	15,77	-	0	0	0	0	
i15	4672	1295	1295	54,55	1295	74,19	1295	13,80	1295	15,42	72,28	0	0	0	0	
i16	4320	1375	1364	63,91	1365	170,36	1364	14,46	1364	15,60	68,43	0,80	0	0,07	0	
i17	-	1283	1283	56,28	1283	46,58	1283	13,73	1283	15,51	-	0	0	0	0	
i18	3681	1346	1345	53,98	1345	84,02	1345	12,72	1345	15,73	63,46	0,07	0	0	0	
i19	2400	1370	1370	52,83	1367	123,19	1367	13,39	1367	15,45	43,04	0,22	0,22	0	0	
i20	-	1328	1328	53,38	1328	82,30	1328	12,82	1328	14,89	-	0	0	0	0	
i21	-	1346	1341	53,52	1341	108,08	1341	12,68	1341	15,22	-	0,37	0	0	0	
i22	3489	1332	1326	57,97	1326	105,38	1326	12,62	1326	15,45	61,99	0,45	0	0	0	
i23	-	1266	1266	53,75	1266	43,72	1266	12,62	1266	15,64	-	0	0	0	0	
i24	4867	1261	1260	54,09	1260	78,91	1260	12,64	1260	15,50	74,11	0,08	0	0	0	
i25	1993	1379	1377	53,56	1376	96,58	1376	12,62	1376	15,64	30,96	0,22	0,07	0	0	
i26	2520	1330	1318	57,34	1318	101,11	1318	12,62	1318	15,59	47,70	0,90	0	0	0	
i27	3209	1261	1261	69,98	1261	82,86	1261	12,64	1261	15,22	60,70	0	0	0	0	
i28	-	1365	1360	58,47	1360	52,91	1359	12,71	1359	15,28	-	0,44	0,07	0,07	0	
i29	4809	1282	1280	69,09	1280	203,36	1280	12,62	1280	15,67	73,38	0,16	0	0	0	
i30	-	1351	1344	70,67	1344	71,02	1344	12,58	1344	15,44	-	0,52	0	0	0	
Média	3495,65	1309,70	1306,93	60,26	1306,87	93,99	1306,77	12,79	1306,77	15,48	59,873	0,222	0,012	0,007	0,000	
Melhoras no tempo médio (%)											99,57	87,10	74,31	83,53	-21,03	
¹ CPLEX 10.0.1			² Cordeau <i>et al.</i> (2005)				³ Mauri <i>et al.</i> (2008a)			⁴ Mauri <i>et al.</i> (2008b)		⁵ Oliveira <i>et al.</i> (2010)				

Foram realizados 10 testes aleatórios para cada instância. Nesses testes, o GRASP+PR utilizou um tempo computacional médio de 15,48 segundos por instância. Além disso, o desvio entre a média das soluções obtidas em relação a melhor solução conhecida para cada instância foi de apenas 0,07%.

A Tabela 1 apresenta uma comparação entre as melhores soluções obtidas pelo GRASP+PR com as melhores soluções descritas em quatro trabalhos recentes encontrados na literatura, além do CPLEX 10.0.1. Como pode ser observado na Tabela 1, as soluções obtidas pelo GRASP+PR foram melhores do que as obtidas pela Busca Tabu proposta por Cordeau *et al.* (2005) e pelo CPLEX. Já em relação ao SA+RA (Mauri *et al.*, 2008a), ao PTA/LP (Mauri *et al.*, 2008b) e ao CS (Oliveira *et al.*, 2010), o GRASP+PR apresentou, em todos os casos, soluções similares.

Em relação ao tempo computacional, o CPLEX utilizou 1 hora para cada instância, enquanto que a Busca Tabu utilizou aproximadamente 120 segundos, conforme descrito em Cordeau *et al.* (2005). Já o SA+RA e o ATP/PL utilizaram tempos médios de 60,26 e 93,99 segundos, respectivamente. Assim, percebe-se a competitividade do GRASP+PR proposto, que utilizou um tempo médio de 15,48 segundos para cada instância.

Em comparação com o método proposto por Oliveira *et al.* (2010), o GRASP+PR apresentou as mesmas soluções, porém com um acréscimo médio de 2,7 segundos. Entretanto, cabe ressaltar que o método proposto apresenta menor complexidade tanto na compreensão quanto na implementação, o que o torna uma alternativa simples e eficiente para resolver o PAB e outros problemas correlatos.

6. Conclusões

Este trabalho apresentou uma abordagem baseada na aplicação do GRASP com o *Path Relinking* para resolução do Problema de Alocação de Berços (PAB).

O método proposto (GRASP+PR) é baseado em ideias simples, de fácil compreensão e implementação, o que o torna uma alternativa, de grande potencial prático, para resolução de casos reais do PAB.

O algoritmo desenvolvido se mostra robusto, apresentando um desvio de 0,07%, utilizando, no pior caso, menos de 16 segundos para resolução das instâncias consideradas. De uma forma geral, os resultados obtidos demonstram que a abordagem proposta foi capaz de gerar soluções viáveis em todos os casos.

Agradecimentos: Os autores agradecem aos revisores anônimos pelos valiosos comentários e sugestões, e à Fundação de Amparo à Pesquisa do Espírito Santo - FAPES (processo 45391998/09) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (processo 300747/2010-1) pelo apoio financeiro.

Referências Bibliográficas

BROWN, G.G., LAWPHONGPANICH, S. & THURMAN, K.P. (1994). Optimizing ship berthing. *Naval Research Logistics*, 41, 1-15.

BROWN, G.G., CORMICAN, K.J., LAWPHONGPANICH, S. & 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. & LUSBY, R. (2009). Models for the discrete berth allocation problem: a computational comparison. Technical Report 14/2009 - Technical University of Denmark.

CHEONG, C.Y., TAN, K.C., LIU, D.K. & LIN, C.J. (2008). Multi-objective and prioritized berth allocation in container ports. *Annals of Operations Research*, 180(1), 63-103.

CCE (2009). Comissão das Comunidades Europeias. Objetivos estratégicos e recomendações para a política comunitária de transporte marítimo no horizonte de 2018. Comunicação da Comissão ao Parlamento Europeu, ao Conselho, ao Comitê Econômico e Social Europeu e ao Comitê das regiões. Disponível em: <http://eur-ex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2009:0008:FIN:PT:PDF>.

CORDEAU, J.F., LAPORTE, G., LEGATO, P. & MOCCIA, P. (2005). Models and tabu search heuristics for the berth allocation problem. *Transportation Science*, 39, 526-538.

FEO, T.A. & RESENDE, M.G.C. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8, 67-71.

GIALLOMBARDO, G., MOCCIA, L., SALANI, M. & VACCA, I. (2010). Modeling and solving the tactical berth allocation problem. *Transportation Research Part B*, 44, 232-245.

GLOVER, F., LAGUNA, M. & MARTÍ, R. (2000). Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 39, 653-684.

IMAI, A., NISHIMURA, E. & PAPADIMITRIOU, S. (2001). The dynamic berth allocation problem for a container port. *Transportation Research Part B*, 35, 401-417.

IMAI, A., NISHIMURA, E. & PAPADIMITRIOU, S. (2003). Berth allocation with service priority. *Transportation Research Part B*, 37, 437-457.

MAURI, G.R. (2008). Novas abordagens para representação e obtenção de limitantes e soluções para alguns problemas de otimização combinatória. Tese (Doutorado em Computação Aplicada), Instituto Nacional de Pesquisas Espaciais (INPE).

MAURI, G.R., OLIVEIRA, A.C.M. & 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, 113-127.

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

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

OLIVEIRA, R.M., MAURI, G.R. & LORENA, L.A.N. (2010). Clustering search aplicado ao problema de alocação de berços. In: Anais do XLII SBPO – Simpósio Brasileiro de Pesquisa Operacional. Disponível em: http://www.sobrapo.org.br/sbpo2010/xliisbpo_pdf/72341.pdf.

THURMAN, K.P. (1989). Optimal ship berthing plans. Dissertation (Masters of Science in Operations Research), Naval Postgraduate School - USA.

UNCTAD (2009). United Nations conference on trade and development. Review of maritime transport. Disponível em: http://www.cbss.org/component/option,com_attachments/id,550/task,download/.