Uma Heurística ILS para Resolução do Problema de Carregamento e Descarregamento de Contêineres em Navios Porta-contêineres

Amaro José de S. Neto[1], Dalessandro S. Vianna[2], Marcilene de Fátima D. Vianna[3]

[1]Universidade Candido Mendes (UCAM-Campos)
Rua Anita Pessanha, 100, Parque São Caetano, Campos dos Goytacazes-RJ. 28040-320.

[2]Departamento de Ciência e Tecnologia – RCT
Universidade Federal Fluminense, Pólo Universitário Rio das Ostras – UFF/PURO
Rua Recife, s/n, Jardim Bela Vista, Rio das Ostras-RJ. 22890-000.

[3]Departamento Fundamentos de Ciência e Sociedade – SFC
Universidade Federal Fluminense, Pólo de Campos dos Goytacazes – UFF/PUCG
Rua Recife, s/n, Jardim Bela Vista, Rio das Ostras-RJ. 22890-000.

amaro.neto@gmail.com, dalessandro@pq.cnpq.br,
marcilenedianin@vm.uff.br

***Abstract**: When docking at a port terminal it may be necessary to perform various operations of loading and unloading containers. Sometimes, when unloading, the target container which needs to be unloaded may be positioned below other containers that will not be unloaded at this time. These ones need to be removed to unload the target container. The goal is to find the best loading sequence minimizing thus the number of "rearrangements". The proposed heuristic was compared with a greedy heuristic and a local search method. The results show the adequacy of ILS heuristic to the problem addressed.*

***Resumo**: Ao atracar em um terminal portuário pode ser necessário realizar diversas operações de carregamento e descarregamento de contêineres. Algumas vezes, no descarregamento, o contêiner alvo que necessita ser desembarcado pode estar posicionado abaixo de outros contêineres que não serão descarregados neste momento. Estes precisarão ser removidos para descarregar o contêiner alvo. O objetivo é encontrar a melhor sequência de carregamento, minimizando, desta forma, o número de "remanejos". A heurística proposta foi comparada com uma heuristica gulosa e um método de busca local. Os resultados obtidos mostram adequação da heurística ILS ao problema abordado.*

Palavras-chave: Otimização; Carregamento; Contêineres; Metaheurísticas; ILS.

1. Introdução

Os navios porta-contêineres são embarcações especializadas em transporte de carga conteinerizada. Estas embarcações dispõem de espaços celulares (baias), onde os contêineres são empilhados (ver Figura 1). A movimentação da carga ocorre tanto nas baias quanto no convés do navio através de equipamentos de bordo ou de terra (GÓES,

2002). Devido à estrutura do navio e a forma que a carga deve estar disposta, o acesso aos contêineres é feito pelo topo da pilha. Pode ser necessário movimentar alguns contêineres para descarregar outros que estão numa posição inferior. A essa operação dá se o nome *remanejo*.

Terminais Concentradores são portos que têm a finalidade de atender à concentração de cargas (conteinerizada) de toda uma região para posterior distribuição para outros portos. (BERTOLANI; LEME, 2004). Sua eficiência está relacionada à ordenação e forma de lidar com os contêineres (SOBRAL *et al.*, 2009). Um navio porta-contêineres pode atracar em terminais concentradores para realizar diversas operações de carga e descarga de contêineres. Existem dois critérios fundamentais no problema de planejamento de estiva: a minimização do número de remanejos e as restrições de estabilidade.

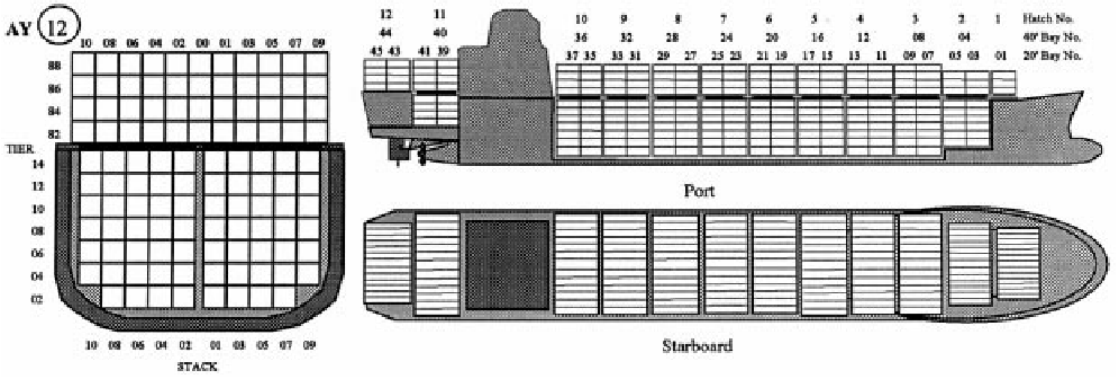

Figura 1. Estrutura de um navio. Fonte: Wilson e Roach (2000)

Neste artigo o foco é a embarcação porta-contêiner. É discutida a posição onde as cargas são armazenadas no navio, de modo que em cada porto, mesmo necessitando realizar operações de carregamento e descarregamento de contêineres, o número de remanejos seja o menor possível. De acordo com Avriel *et al*. (2000), este problema é NP-Completo.

O objetivo deste trabalho é desenvolver uma heurística ILS (*Iterated Local Search*) para o problema de carregamento e descarregamento de contêineres em um navio porta-contêiner, visando minimizar o número de remanejos.

Na literatura podem ser encontrados trabalhos que abordam problemas semelhantes ao tratado neste trabalho. Dentre estes, pode-se citar:

- Sobral *et al.* (2009) desenvolveram um algoritmo *Beam Search* para Resolução do problema de carregamento e descarregamento de contêineres em terminais portuários;
- Campos (2008) elaborou um estudo sobre a integração entre carregamento e roteamento de veículos;
- Morabito e Arenales (1997) mostraram diferentes abordagens para o problema de carregamento de contêineres;
- Raidl (1999) propôs um algoritmo genético para o problema de empacotamento de múltiplos contêineres.
- Martins *et al.* (2009) estudaram o problema de estocagem de contêineres; e

- Azevedo (2009) propôs um algoritmo genético para resolver o problema de carregamento e descarregamento de contêineres em terminais portuários.

O restante do trabalho está organizado da seguinte maneira: na Seção 2 é apresentado o problema a ser resolvido. Na Seção 3 é mostrada a representação da demanda de contêineres, da baia e da solução, além da função de avaliação. Na Seção 4 é apresentada a metaheurística ILS desenvolvida. Na Seção 5 são apresentados os resultados computacionais obtidos e na Seção 6 as conclusões.

2. Apresentação do problema

Os navios porta-contêineres são embarcações de grande porte possuem uma estrutura celular que facilita a manipulação da carga armazenada em contêineres. As células são agrupadas por seções (baias), onde contêineres podem ser empilhados. Possui uma rota $R = \{p_0, p_1, p_2, ..., p_n\}$ que é conhecida antes de sua partida. Geralmente são circulares, partindo de um Porto p_0, percorrendo todos os Portos p_i, descarregando as cargas nos portos de destino e carregando novas cargas destinadas a outros portos, e ao fim retorna ao Porto p_0. Cada um dos portos p que compõe R tem uma demanda de contêineres que necessitarão ser carregados na embarcação. Além disto, o navio possui uma capacidade que é medida em TEU (*Twenty-foot Equivalent Units*). Por exemplo, uma embarcação com capacidade de 1000 TEUs pode armazenar 1000 contêineres de 20 pés.

Ao atracar em um Porto p_i, o navio primeiro efetuará o descarregamento de contêineres destinados a p_i, caso haja. Nesta operação, pode ser necessário descarregar temporariamente os contêineres que estão armazenados numa pilha de uma determinada baia do navio, para conseguir descarregar um contêiner que está localizado em uma posição inferior desta pilha. Posteriormente, caso haja demanda, os contêineres serão carregados do terminal para o navio, isto implica numa sequência de carregamento $S = \{c_0, c_1, c_2, ..., c_n\}$ para cada porto que o navio irá atracar, onde deve ser buscado um melhor arranjo destas sequências de forma a minimizar o número de remanejos para os portos seguintes.

3. Representação

Nesta seção será discutida a representação da demanda de contêineres, da baia, da solução e também a função de avaliação.

PORTO DESTINO

PORTO ORIGEM	1	2	3	4	5
1	0	2	4	2	4
2	5	0	0	4	0
3	0	2	0	2	6
4	6	0	0	0	0
5	0	6	0	4	0

Figura 3. Exemplo de uma matriz de transporte de um navio

3.1 Demanda de contêineres

A representação da demanda de contêineres será através de uma matriz de transporte T_{od}, onde o=1,2,...,N e d = 1, 2, ...,N (N é o número de portos). Considere o sendo o porto de origem, e d o porto de destino. O número na posição T_{od} corresponde a

quantidade de contêineres que deverão ser carregados no Porto *o* para posteriormente serem descarregados no Porto *d*, ou seja, as demandas correspondentes. Na Figura 3 é mostrado um exemplo de uma matriz de transporte. Por exemplo, quando o navio atracar no Porto 2, será necessário carregar $T_{21} = 5$ contêineres que têm como destino o Porto 1 e $T_{24} = 4$ contêineres com destino ao Porto 4.

3.2 Baia

Uma baia será representada computacionalmente por uma matriz de ocupação O_{lc}, onde l=1,2,...,L e c = 1, 2, ..., H (onde L e H significam quantos contêineres cabem na largura e na altura do navio respectivamente). Nela serão registradas suas posições (l,c) livres e ocupadas. Em cada célula O_{lc} ocupada, o valor inteiro corresponde ao destino em que a carga deverá ser entregue. $O_{lc} = 0$ significa que a posição (l,c) da matriz está livre. A simplicidade desta representação facilita à compreensão do armazenamento, assim como a manipulação dos itens. Veja no exemplo da Figura 4, o valor na posição (1,1) = 0, o que significa que esta posição está livre e pronta para receber alguma carga. Enquanto na posição (3,2) = 3, corresponde ao porto no qual o contêiner da posição (3,2) deverá ser descarregado, neste caso o Porto 3.

	1	2	3
1	0	5	0
2	0	3	3
3	2	3	2

Figura 4. Exemplo da representação da baia de um navio com capacidade de 9 contêineres.

Quando ocorre a atracação do navio em um porto, ocorrem operações de carregamento e descarregamento de contêineres. Devido à estas operações as informações nesta matriz de ocupação são atualizadas constantemente. A cada porto atracado, será necessário pelo menos uma destas operações, caso contrário a atracação não ocorreria.

PORTO ORIGEM \ PORTO DE DESTINO												
1	4	4	2	2	5	5	5	5	3	3	3	3
2	1	1	1	1	1	4	4	4	4			
3	4	4	2	2	5	5	5	5	5	5		
4	1	1	1	1	1	1						
5	4	4	4	4	2	2	2	2	2	2		

Figura 5. Representação de uma solução

3.3 Solução

Neste trabalho, uma solução é representada por uma lista, com o tamanho determinado pelo número de portos. Cada posição *i* da lista armazena um vetor que possui as demandas que devem ser embarcadas no Porto *i*. Cada vetor é uma sequência de carregamento, onde o primeiro item será o primeiro a ser armazenado no navio, e assim sucessivamente. A Figura 5 ilustra uma solução para um problema com 5 portos. A demanda considerada será a mesma da Figura 3. Suponha a rota de visitação do navio: 1-3-5-2-4-1. Cada posição do vetor corresponde a um contêiner, e o número contido em cada uma destas posições refere-se ao porto que o contêiner deve ser descarregado. Por exemplo, quando a embarcação atracar no Porto 5, de acordo com a respectiva

sequência, primeiramente serão carregados no navio 4 contêineres com destino ao Porto 4, em seguida serão carregados 6 contêineres destinados ao Porto 2.

Tabela 1. Execução da função de avaliação

ITERAÇÃO	ORDEM VISITAÇÃO	SITUAÇÃO NAVIO		DESCARREGAR CONTÊINERES		CARREGAR CONTÊINERES		TOTAL REMANEJOS
1	1	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 5 5 3 4 4 2	0 0 0 3 3 3 2 5 5	0
2	3	0 0 0 5 5 3 4 4 2	0 0 0 3 3 3 2 5 5	0 0 0 5 5 0 4 4 2	0 0 0 0 0 0 2 5 5	5 5 5 5 5 2 4 4 2	5 5 5 4 4 2 2 5 5	0
3	5	5 5 5 5 5 2 4 4 2	5 5 5 4 4 2 2 5 5	0 0 0 0 0 2 4 4 2	0 0 0 4 0 0 2 4 2	2 2 2 4 4 2 4 4 2	2 2 2 4 4 4 2 4 2	2
4	2	2 2 2 4 4 2 4 4 2	2 2 2 4 4 4 2 4 2	0 0 0 4 4 0 4 4 0	0 0 0 0 4 0 4 4 4	1 4 4 4 4 1 4 4 1	4 4 0 1 4 1 4 4 4	4
5	4	1 4 4 4 4 1 4 4 1	4 4 0 1 4 1 4 4 4	0 0 0 0 0 1 1 0 1	0 0 0 0 0 0 1 1 0	0 0 0 1 1 1 1 1 1	0 0 0 1 1 0 1 1 1	7
6	1	0 0 0 1 1 1 1 1 1	0 0 0 1 1 0 1 1 1	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	7

3.4 Função de Avaliação

O objetivo principal do problema proposto é minimizar o número de remanejos. Para isto, a função que irá avaliar o problema terá como entrada a lista de sequências de carregamento (solução). Ao término, será retornado o número total de remanejos efetuados, através da simulação do cumprimento de um determinado trajeto, realizando descarregamentos e carregamentos, quando necessários, de acordo com as respectivas sequências de carregamento. Um exemplo desta função é mostrado pela Tabela 1, a qual utiliza como dado de entrada a solução descrita na Figura 5 e que possui duas baias de tamanho 3x3 (LxH, onde L e H significam quantos contêineres cabem na largura e na altura do navio respectivamente). Por exemplo, na iteração 1, de acordo com a ordem de visitação, o navio atracará no Porto 1 e encontra-se vazio. Como não há contêineres a serem descarregados, é feito o carregamento de 2 contêineres com destino para o Porto 4, 2 com destino para o Porto 2, 4 com destino para o Porto 5 e 4 com destino para o Porto 3. Na iteração 2, o navio está parcialmente ocupado e atracado no Porto 3. É feito

o descarregamento de 4 contêineres que tinham como destino o Porto 3, e em seguida realiza-se o carregamento. Na iteração 3 o navio está totalmente ocupado e atracado no Porto 5. Será feito o descarregamento de contêineres com destino ao Porto 5, mas como pode ser observado, na baia 2 existem contêineres que precisam ser descarregados e estão no fundo da baia e com contêineres sobre eles. Será necessário descarregar esses contêineres numa pilha auxiliar para então descarregar os contêineres que têm como destino o Porto 5. Computa-se o número de remanejos = 2, e então inicia-se o processo de carregamento e o navio fica totalmente ocupado. Este processo é repetido ao longo de todo o trajeto de visitação.

4. Heurística ILS proposta

A Metaheurística ILS (*Iterated Local Search*) é um método de busca local que procura em um subespaço do espaço de busca de soluções, definido por soluções que são ótimas locais de determinado procedimento de otimização (LOURENÇO *et al.*, 2002). Existem 5 pontos importantes para o funcionamento do algoritmo: a geração de uma solução de partida; o procedimento de perturbação; o procedimento de busca local; o critério de parada; e o critério de aceitação. Primeiramente é realizada uma busca local na solução de partida *s* (gerada após método de construção guloso), posteriormente *s* será perturbada e realizada uma busca local gerando uma nova solução *s´´* e caso seja mais apta que *s* passará a ser a nova solução de partida e o processo de perturbação e busca local se repete até um critério de parada seja satisfeito. A solução mais apta encontrada ao longo de todas as iterações ILS é retornada como resultado.

O sucesso do uso da metaheuristica ILS em problemas de otimização é descrito em diversos trabalhos da literatura, dentre os quais pode se citar: (TEMPONI, 2007), (SOUZA FILHO, 2008), e (MESQUITA, 2010).

Nas próximas subseções serão apresentadas as etapas da heurística proposta. As subseções 4.1 e 4.2 e 4.3 apresentam as fases de construção, de perturbação e de busca local, respectivamente.

4.1 Fase de construção

Esta fase tem como objetivo prover uma solução factível e de boa qualidade para a fase de busca local. A solução será construída iterativamente, elemento por elemento. Para cada porto são selecionados todos os contêineres que devem ser carregados. Em seguida, são ordenados numa lista de forma que os contêineres destinados ao porto mais distante no trajeto de visitação ficarão como primeiros na sequência (critério guloso).

Na Tabela 2 é mostrado o algoritmo de construção de uma solução. O laço que vai da linha 1 até 9 garante que será montada uma sequência de carregamento para cada porto. O laço que vai da linha 3 até 8 garante a montagem da sequência na ordem inversa a de visitação, ou seja, os contêineres destinados aos portos mais distantes no trajeto de visitação serão colocados como primeiros na lista. O laço que vai da linha 4 até 7 garante que todas as demandas serão colocadas na sequência de carregamento. Na linha 5 é montada a sequência de carregamento de cada porto, para então na linha 10 retornar à solução completa.

Na Figura 6 é ilustrada à construção de uma solução. Considere que o navio visitará 3 portos, sua ordem de visitação é 1-3-2-1. Observe que na montagem da ordem de carregamento do Porto 1, primeiro são selecionadas as cargas que têm como destino

o porto que será visitado por último. De acordo com a ordem de visitação, o último é o Porto 1, mas como a sequência em construção pertence ao Porto 1, não teria sentido este porto ter demanda de carregamento para ele mesmo. Então o Porto 2 tem sua demanda selecionada e colocada como os primeiros contêineres que serão carregados da sequência. Em seguida, seleciona-se a demanda do porto que será visitado antes do Porto 2, neste caso, o Porto 3, que têm sua demanda selecionada e colocada na sequência também. A montagem das sequências para os outros portos segue a mesma lógica.

Tabela 2. Algoritmo de construção de uma solução

Algoritmo constroiSolucao(*ordemvisita*, *matrizdemanda*)
Entrada
ordemvisita – ordem na qual o navio fará os carregamentos e descarregamentos
matrizdemanda – matriz onde são registradas as demandas de carregamento de cada porto
Saída
solucao – matriz onde são registradas as sequências de carregamento de cada porto
Inicio
1. **para cada porto** *i* **faça**
2. $s \leftarrow 0$
3. **para cada elemento** *j* **de** *ordemvisita* **do último para o primeiro faça**
4. **para cada elemento** *k* **de** *matrizdemanda*[*i*][*j*] **faça**
5. *solucao*[*i*][*s*] $\leftarrow j$
6. $s \leftarrow s + 1$
7. **fim**-para
8. **fim-**para
9. **fim-**para
10. **retorne** *solução*

Fim-constroiSolucao

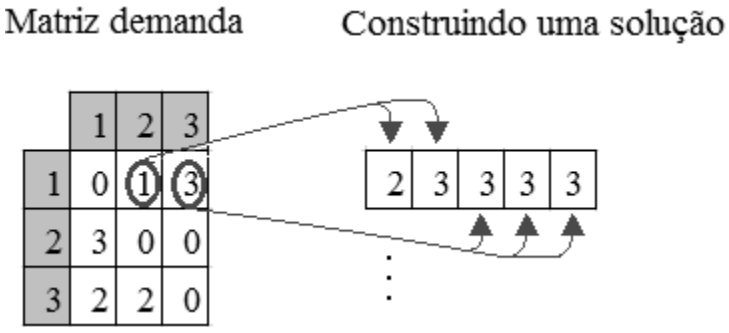

Figura 6. Montagem de uma solução

4.2 Fase de perturbação

Esta fase tem como objetivo prover mudanças em uma solução. A perturbação considerada consiste em efetuar *r* trocas em cada sequência de carregamento, a escolha dos contêineres trocados é aleatória. O nível perturbação deve ser controlado, pois fortes perturbações podem implicar em reinício aleatório, e perturbações demasiadamente fracas podem reduzir as chances de encontrar novas soluções.

4.3 Fase de busca local

O objetivo desta fase é possibilitar uma melhoria para a solução construída através da busca em uma vizinhança por uma solução de melhor qualidade.

A vizinhança é o conjunto de soluções próximas a solução inicial que pode ser obtida por um movimento, onde o algoritmo de busca local poderá procurar por uma solução melhor. Neste trabalho, o movimento consiste em trocar nas sequências de carregamento, um contêiner de uma determinada posição por um outro na mesma sequência em posição diferente. Com a troca realizada, a respectiva sequência é alterada e reavaliada, para assim, obter seu número de remanejos.

A Tabela 3 mostra o algoritmo de busca local. O laço que vai da linha 2 até 14 garante que as sequências de cada porto serão contempladas com a troca. O laço que vai da linha 3 até 13 irá percorrer toda a sequência. O laço que vai da linha 4 até 10 fará com que se busque na sequência a próxima posição para efetuar a troca. Na linha 5 e 6, é realizada a troca e a avaliação, respectivamente. O bloco delimitado pela linha 7 e 11 serve para verificar se a troca gerou uma solução melhor; caso contrário desfaz a troca e a solução volta a seu estado anterior.

Tabela 3. Algoritmo de busca local

```
Algoritmo buscalocal(solucao)
Entrada
   solucao - matriz onde são registradas as sequências de carregamento de cada porto
Início
1.  remanejo ← ∞
2.  para cada porto i faça
3.     para cada elemento j de solucao[i] faça
4.        para cada elemento k, onde k > j de solucao[i] faça
5.           trocar elemento solucao[i][j] por solucao[i][k]
6.           remanejoObtido ← avalia(solucao)
7.           se remanejoObtido < remanejo então
8.              remanejos←remanejoObtido
9.           senão
10.             trocar elemento solucao[i][j] por solucao[i][k]
11.          fim-se
12.       fim-para
13.    fim-para
14. fim-para
Fim- buscalocal
```

5. Resultados

Por questões de balanceamento do navio, o atendimento a uma determinada demanda de carregamento será de acordo com o menor valor entre a disponibilidade da baia selecionada e o bloco máximo de contêineres que pode ser atendido por uma baia. A baia selecionada é a que detém o menor número de contêineres armazenados. Para calcular o bloco máximo é necessário que seja determinado o número de camadas da baia que pode ser preenchido sem comprometer o balanceamento da embarcação. O bloco é obtido a partir do número de camadas x largura da baia. Neste trabalho foi considerado que o número de camadas é a metade da altura. Por exemplo, um navio com 2 baias que suportam 7 contêineres na largura *L* e 18 na altura *H* (*L* e *H* são dados de entrada do algoritmo), teria o número de camadas igual a 9 e o bloco máximo seria de 63.

Sobre a carga que será carregada e descarregada, não é considerado seu peso e são supostas cargas homogêneas, ou seja, com o mesmo tamanho.

Foram geradas 18 instâncias, considerando navios com capacidade de 600 a 2400 TEUs (*Twenty-foot Equivalent Units*), com visitação para 10, 15, e 20 portos, totalizando 54. Estas instâncias têm as seguintes informações: número de portos, número de baias do navio, a dimensão (*L*x*H*) das baias, a ordem de visitação do navio e a demanda que o navio deve atender em cada porto. Foi considerado que as demandas dos portos não excedem a capacidade do navio.

O ILS tem como entrada uma solução inicial obtida através de um método construtivo guloso. Para avaliar o desempenho dos resultados do algoritmo proposto, foi usado como parâmetro de comparação o número de remanejos obtidos com o método construtivo puramente guloso (construtivo) e o construtivo puramente guloso seguido da busca local. Na perturbação foi considerado *r* movimentos = 2. e o número de iterações do ILS foi de 20 iterações. Estes parâmetros foram determinados empiricamente.

Os algoritmos foram desenvolvidos na linguagem de programação C e executado em um computador Core 2 Duo 2.0, com 2 GB de memória RAM no sistema operacional Windows XP. As Tabelas 4, 5 e 6 mostram os resultados dos algoritmos após uma execução de cada algoritmo. Conforme pode ser observado, o algoritmo ILS se mostrou superior ao método puramente guloso em todos os casos testados. Em relação ao construtivo seguido da busca local, o ILS somente não obteve melhora no teste número 17 da tabela 6. A média percentual de melhora do ILS em relação à construção encontrada foi de 29%. Houve casos em que a diferença percentual de melhora entre o ILS e o construtivo seguido da busca local foi superior a 15%, como pode ser visto no teste de número 10 da Tabela 4.

Tabela 4. Resultados obtidos para instâncias com 10 portos

Nº	BAIA			TEUs	CONSTRUTIVO		BUSCA LOCAL			ILS		
	Q	L	H		CUSTO	TEMPO	CUSTO	%	TEMPO	CUSTO	%	TEMPO
1	12	5	10	600	558	0	470	16%	1,6	408	27%	40,23
2	12	7	10	840	979	0	674	31%	3,73	624	36%	86,57
3	12	10	10	1200	945	0	721	24%	6,89	751	21%	155,03
4	12	5	13	780	1713	0	1444	16%	8,87	1338	22%	283,59
5	12	7	13	1092	835	0	544	35%	3,28	528	37%	71,57
6	12	10	13	1560	1569	0	1299	17%	13,23	1196	24%	311,45
7	12	5	15	900	777	0	638	18%	3,57	573	26%	90,71
8	12	7	15	1260	1024	0	708	31%	7,04	679	34%	158,06
9	12	10	15	1800	1489	0	992	33%	15,51	882	41%	325,73
10	16	5	10	800	809	0	652	19%	3,12	540	33%	74,62
11	16	7	10	1120	1169	0	805	31%	6,68	650	44%	147,21
12	16	10	10	1600	993	0	695	30%	10,62	679	32%	249,9
13	16	5	13	1040	1352	0	1165	14%	6,17	1053	22%	147,57
14	16	7	13	1456	1303	0	985	24%	9,95	937	28%	233,71
15	16	10	13	2080	1665	0	1456	13%	25,48	1272	24%	563,73
16	16	5	15	1200	1130	0	888	21%	7,82	865	23%	168,09
17	16	7	15	1680	1302	0	924	29%	13,5	870	33%	295,53
18	16	10	15	2400	3899	0	3042	22%	40,96	2593	33%	926,34

Tabela 5. Resultados obtidos para instâncias com 15 portos

Nº	BAIA			TEUs	CONSTRUÇÃO		BUSCA LOCAL			ILS		
	Q	L	H		CUSTO	TEMPO	CUSTO	%	TEMPO	CUSTO	%	TEMPO
1	12	5	10	600	1191	0	941	21%	4,25	843	29%	98,12
2	12	7	10	840	1263	0	994	21%	7,14	890	30%	156,04
3	12	10	10	1200	1502	0	1153	23%	13,98	1091	27%	324,75
4	12	5	13	780	1191	0	941	21%	4,23	865	27%	95,45
5	12	7	13	1092	1716	0	1322	23%	13,96	1234	28%	310,56
6	12	10	13	1560	2506	0	1804	28%	30,01	1689	33%	653,65
7	12	5	15	900	1924	0	1472	23%	16,15	1398	27%	337,43
8	12	7	15	1260	3089	0	2379	23%	22,34	2140	31%	515,04
9	12	10	15	1800	2497	0	2041	18%	32,81	1869	25%	773,37
10	16	5	10	800	1174	0	859	27%	7,25	806	31%	155,01
11	16	7	10	1120	1648	0	1120	32%	12,84	1049	36%	298,7
12	16	10	10	1600	2659	0	1647	38%	26,43	1547	42%	604,15
13	16	5	13	1040	2078	0	1615	22%	13,57	1552	25%	351,59
14	16	7	13	1456	2291	0	1769	23%	26,92	1693	26%	630,35
15	16	10	13	2080	2706	0	2035	25%	51,14	1898	30%	1156,75
16	16	5	15	1200	1682	0	1256	25%	14,93	1225	27%	339,35
17	16	7	15	1680	2383	0	1893	21%	31,59	1740	27%	680,21
18	16	10	15	2400	3895	0	3141	19%	68,98	2800	28%	1498,1

Tabela 6. Resultados obtidos para instâncias com 20 portos

Nº	BAIA			TEUs	CONSTRUÇÃO		BUSCA LOCAL			ILS		
	Q	L	H		CUSTO	TEMPO	CUSTO	%	TEMPO	CUSTO	%	TEMPO
1	12	5	10	600	1389	0	983	29%	6,98	904	35%	154,96
2	12	7	10	840	1863	0	1346	28%	14,65	1292	31%	329,15
3	12	10	10	1200	3044	0	2263	26%	28,98	2153	29%	681,28
4	12	5	13	780	1984	0	1444	27%	12,54	1319	34%	295,09
5	12	7	13	1092	2490	0	1903	24%	24,71	1714	31%	549,85
6	12	10	13	1560	3293	0	2335	29%	42,95	2276	31%	975,61
7	12	5	15	900	1974	0	1618	18%	17,32	1554	21%	392,81
8	12	7	15	1260	3628	0	2860	21%	38,78	2662	27%	838,45
9	12	10	15	1800	3140	0	2380	24%	59,56	2347	25%	1339,32
10	16	5	10	800	1098	0	802	27%	11,26	794	28%	251,96
11	16	7	10	1120	2250	0	1668	26%	23,07	1519	32%	538,18
12	16	10	10	1600	2333	0	1596	32%	39,67	1556	33%	884,37
13	16	5	13	1040	2171	0	1600	26%	22,81	1530	30%	505,17
14	16	7	13	1456	3471	0	2758	21%	45,59	2646	24%	1074,76
15	16	10	13	2080	5940	0	4551	23%	98,82	4277	28%	2296,87
16	16	5	15	1200	3187	0	2457	23%	29,31	2247	29%	686,4
17	16	7	15	1680	2487	0	1904	23%	47,46	1894	24%	1050,53
18	16	10	15	2400	5740	0	4364	24%	128,78	4101	29%	2868,2

6. Conclusões

Neste trabalho foi desenvolvida uma heurística ILS para a resolução do problema de carregamento e descarregamento de contêineres em navios, no qual objetiva-se realizar um número mínimo de remanejos. Experimentos computacionais foram realizados sobre 54 problemas testes, nos quais o método proposto foi comparado com um método construtivo e com um método de descida (método construtivo seguido de busca local). Os resultados obtidos destacam uma melhora média de 29% na qualidade das soluções quando comparado com o método construtivo e de 5% em relação ao método de descida. Isso demonstra a adequação da heurística ILS proposta ao problema abordado.

Agradecimentos

Este trabalho foi financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e pela Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ).

Referências

AVRIEL, M.; PENN, M.; SHPIRER, N. (2000) "Container ship stowage problem: complexity and connection to the coloring of circle graphs". Discrete Applied Mathematics, v. 103, p. 271-279.

AZEVEDO, A. T.; SOBRAL, C. M.; DEUS, N. M. R. (2009) "Resolução do problema de carregamento e descarregamento de contêineres em terminais portuários via Algoritmo genético", XVI SIMPEP.

BERTOLANI, A. D.; LEME, F. L. (2004) "Carregamento de contêineres em navios. Universidade Presbiteriana Mackenzie".

CAMPOS, D.S. (2008) "Integração dos problemas de carregamento e roteamento de veículos com janela de tempo e frota heterogênea". 119p. Tese (Doutorado em Engenharia de Produção) - Universidade de São Paulo, São Paulo - SP.

GÓES, H. A. (2002) "Planejamento portuário". Rio de janeiro - Escola de Engenharia, Universidade Federal do Rio De Janeiro.

JÚNIOR, L. O. J.; ARROYO, J. E. C; SOUZA, V. A. A. (2009) "Heurísticas GRASP e ILS para o problema no-wait flowshop scheduling multiobjetivo", XLII SBPO.

LOURENÇO, H. R.; MARTIN, O. E STUETZLE, T. (2002) "Iterated local search", Handbook of Metaheuristics, p. 321–353, Norwell, MA. Kluwer Academic Publishers.

MARTINS, P. T.; LOBO, V. J. A. S.; VAIRINHOS, V. (2009) "Container Stowage Problem Solution for Short Sea Shipping", 14º congresso da APDIO.

MESQUITA, A.C.P. (2010) "A metaheurística busca dispersa em problemas de roteirização de veículos com coleta e entrega simultâneas*: aplicação na força aérea brasileira"*. 103p. Dissertação (Mestrado em Engenharia) – Escola politécnica da Universidade de São Paulo, São Paulo - SP.

MORABITO, R.; ARENALES, M. (1997) "Abordagens para o problema do carregamento de contêineres". Pesquisa Operacional, 17 (1), 29-56.

MOTA, L. C. S. ; OCHI, L. S. (2009) “Metaheurísticas com memória adaptativa para o problema de recobrimento de rotas”, IX Congresso Brasileiro de Redes Neurais.

NOGUEIRA, R. T.; JR, G. G. P.; PÓVOA, C. L. R. (2006) “Uma heurística GRASP para o problema do pequeno investidor”, XIII SIMPEP.

RAIDL, G. R. (1999) “A weight-coded genetic algorithm for the multiple container packing problem”, 14th ACM Symposium on Applied Computing.

SOBRAL, C. M.; AZEVEDO, A. T.; LIMA, F. M. B. (2009) “Resolução do problema de carregamento e descarregamento de contêineres em terminais portuários via Beam Search”, XVI SIMPEP.

SOUZA FILHO, G.F. (2008) “Metaheurísticas sequencias e paralelas para a configuração de um serviço de distribuição de vídeo digital ao vivo”. 86p. Dissertação (Mestrado em Informática) – Universidade Federal da Paraíba, João Pessoa - PB..

TEMPONI, E.C.C. (2007) “Uma proposta de resolução do problema de corte bidimensional via abordagem metaheurística”. 100p. Dissertação (Mestrado em modelagem matemática computacional) – Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte- MG.

WILSON, I. D., ROACH, P. A. (2000) “Container Stowage Planning: A methodology for generating computerised solutions”. Journal Of The Operational Research Society.