Recebido em: 11/09/11
Aprovado em: 11/09/11

Resolução do problema de carregamento e descarregamento 3D de contêineres em terminais portuários, via representação por regras e algoritmo genético

Anibal Tavares de Azevedo (UNICAMP – SP/Brasil) - anibal.azevedo@fca.unicamp.br
• R. Pedro Zaccaria, Pedregulho, 1300, Jardim Sta Luzia, CEP 13484-350, Limeira-SP
Cassilda Maria Ribeiro (UNESP – SP/Brasil) cassilda@feg.unesp.br
Luiz Leduino Salles Neto (UNIFESP – SP/Brasil) - luiz.leduino@unifesp.br
Misael Pimentel Espiríndio e Silva (UNESP – SP/Brasil) - misaelsjc@hotmail.com
Murilo Cardoso Silvestre (UNESP – SP/Brasil) - omurilo9@hotmail.com

Resumo

Neste artigo, são apresentadas uma nova formulação e uma nova representação para a solução do problema de carregamento de contêineres 3D em terminais portuários (PCCTP 3D), que consiste em determinar como carregar e descarregamento um conjunto de contêineres de um navio porta-contêiner (*containership*), respeitando restrições operacionais relacionadas aos contêineres e à estrutura do navio; isso, porque num navio porta-contêiner, os contêineres são colocados em pilhas verticais, localizadas em diversas seções (baias). O acesso aos contêineres é feito somente através do topo da pilha. Muitas vezes, para se descarregar um contêiner num determinado porto j, é necessário remover o contêiner cujo destino é o porto j+1, porque ele está acima do contêiner que se deseja descarregar. Esta operação pode ser chamada de remanejamento. Neste artigo, procura-se fazer o carregamento e descarregamento do navio, procurando minimizar o número de remanejamentos, levando em conta a distância do centro de massa e do centro de gravidade, para que não ocorram problemas de instabilidade do navio. Assim como o problema 2D é NP-Completo (AVRIEL *et al.*, 2000), o problema 3D também o é, e é aconselhável a utilização de métodos heurísticos. É proposto um algoritmo genético que emprega uma nova representação das soluções, denominada representação por regras, para resolver o PCCTP 3D. A representação por regras tem a grande vantagem de utilizar uma representação bastante compacta, que assegura a geração de soluções factíveis e possibilita a inserção de conhecimento do tomador de decisões. Nós também, apresentamos alguns resultados obtidos com este método.

Palavras-chave: Algoritmo Genético; Carregamento de Contêiner 3D; Otimização Combinatória.

Abstract

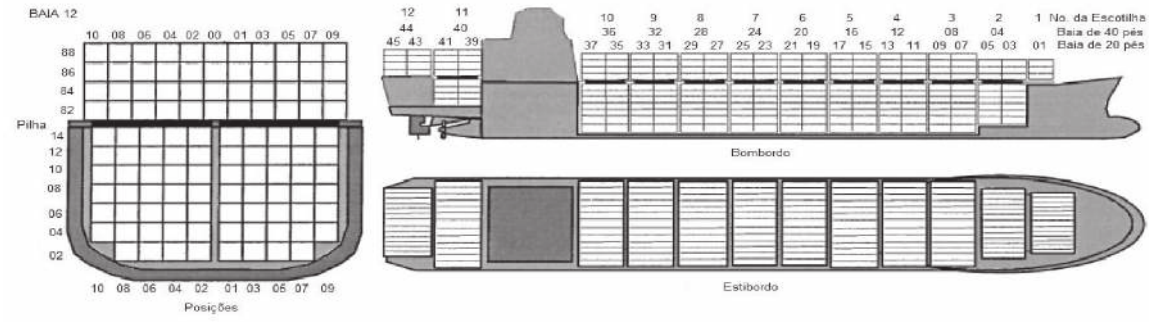
This paper presents a 3D formulation for the Container Ship Loading Planning Problem (3D CLPP) and also proposes a new and compact representation to efficiently solve it. The containers on board in a containership are placed in vertical stacks, located in several sections called bays. The only way to access the containers is through the top of the stack. In order to unload a container at a given port j, it is necessary to remove the container, where the destination is port j+1, because it is located above the container we want to download. This operation is called "shifting". A ship container carrying cargo to several ports may require a large number of shifting operations. These operations spend a lot of time and money but they can be avoided by using efficient stowage planning. The 3D-CLPP consists of determining, at each port, how to load and unload a set of containers in a container ship respecting the operational constraints related to the containers and the ship's structure to minimize the number of container movements and also ship instability. As the 2D problem NP-Complete (Avriel etal. 2000), the 3D problem is NP-Complete as well and it is advisable to use heuristic methods to solve it. Here we describe the binary formulation to 3D-CLPP and also an alternative formulation called representation by rules. The genetic algorithm was combined with the "Representation by Rules" to solve the 3D-CLPP in a manner that ensures that every solution analyzed in the optimization process is compact and feasible. We also present some results obtained with this method.

Keywords: Genetic Algorithm; 3D Containership Problem; Combinatorial Optimization

1. INTRODUÇÃO

A eficiência de um terminal portuário especializado em movimentação de contêineres, depende da ordenação e da agilidade do processo de lidar com os contêineres, especialmente durante o processo de carregamento dos navios. A estiva e o plano de carregamento associado são determinados fundamentalmente por dois critérios: estabilidade do navio e o número mínimo de remanejo, requerido nos diversos pontos de entrega (AVRIEL *et al*., 2000; WILSON e ROACH, 2000; AMBROSINO *et al*., 2006). O último critério é baseado no fato de que muitos navios possuem uma estrutura celular, conforme pode ser observado na figura 1, e os contêineres devem ser carregados de modo a formarem pilhas verticais, o que acarreta, em muitos casos, a necessidade de movimentar alguns contêineres para descarregar outros, posicionados na parte inferior da pilha.

Figura 1 – Estrutura celular de um navio.



Fonte: WILSON e ROACH (2000).

Concomitantemente, outra restrição emerge durante a escolha dos contêineres para carregamento no pátio do terminal, onde geralmente, os contêineres são empilhados, formando blocos à espera do momento de serem carregados. Se os contêineres-alvo, que devem ser carregados mais tarde, são posicionados, nas pilhas abaixo de outros, então a tarefa de carregamento requer remanejo, de modo a remover e reposicionar os contêineres-alvo. Esta situação ocorre com frequência, uma vez que a ordem de carregamento não é conhecida quando as cargas chegam ao pátio do terminal e é denominada de realocação. No entanto, mesmo quando esta informação é disponibilizada a tempo, o arranjo ideal de contêineres na área de armazenamento é praticamente impossível de ser obtido, devido à chegada aleatória de diversas outras cargas. Mais especificamente, o problema de carregamento de contêineres em terminais portuários (PCCTP) consiste em determinar como carregar um conjunto de contêineres de diferentes tipos, em um navio porta-contêiner (*containership*), respeitando restrições operacionais relacionadas aos contêineres, navio e pátio do terminal portuário.

Neste artigo, será apresentado um Algoritmo Genético e um modelo para a solução do PCCTP 3D, que considera a minimização do tempo de carregamento e descarregamento dos contêineres em terminais portuários e a minimização da instabilidade medida, como a distância entre o centro de massa e o centro geométrico. O modelo 3D foi criado para que se pudesse levar em conta a medida de instabilidade, que é uma das contribuições deste artigo. A abordagem, aqui apresentada, é inovadora, porque reduz consideravelmente o número de variáveis do problema e também, porque permite que seja utilizada a experiência do profissional portuário, através de uma representação adequada dos seus conhecimentos. Esse modelo também, permite que ele seja implementado com diferentes heurísticas. Avriel *et al*. (1998) apresentaram uma formulação binária do problema 2D (bidimensional) e mostraram que ele é NP-Completo. Na seção 2, é apresentado o problema 3D (tridimensional) a ser resolvido, bem como sua formulação matemática, utilizando variáveis zero-um. Na seção 3, são apresentadas a representação matricial utilizada na implementação do Algoritmo Genético e sua combinação com a Representação por Regras. Na seção 4, são apresentados os resultados computacionais obtidos e na seção 5, conclusões e sugestões de trabalhos futuros.

2. APRESENTAÇÃO DO PROBLEMA

Um navio porta-contêiner tem sua capacidade medida em TEU (*Twenty-foot Equivalent Units*) ou Unidade Equivalente de Vinte pés. Por exemplo, um navio com capacidade de 8000 TEUs pode carregar 8000 contêineres de vinte pés. Os navios têm uma estrutura celular (vide figura1), onde são alojados os contêineres. Essas células são agrupadas por seções ou baias (em inglês *bays*) e os contêineres são empilhados nessas seções, formando pilhas verticais. Então uma baia é um agrupamento de células, com capacidade de se empilhar um certo número de contêineres. A baia tem então linhas horizontais numeradas r= 1, 2, ..., R (a linha 1 é a linha que está em baixo e a linha R é a linha do topo da pilha) e colunas numeradas c= 1, 2, ..., C (coluna 1 é a primeira coluna da esquerda).

O problema PCCTP 3D, que será resolvido aqui, levará em conta dois objetivos: a redução do número de remanejamentos dos contêineres para um certo número de portos N e a instabilidade do arranjo dos contêineres, isto é, o objetivo é reduzir, ao máximo, o número de remanejamento dos contêineres, de modo que o navio não se torne instável. Para o primeiro objetivo, define-se remanejamento como sendo o descarregamento temporário de contêineres, da pilha de contêineres, com a finalidade de descarregar, num determinado terminal portuário i, um contêiner que está na parte inferior da pilha. Os remanejamentos são necessários, porque os contêineres que estão numa pilha, só podem ser acessados pelo topo. Então, um contêiner que está no meio da pilha, só pode ser descarregado num determinado porto *i*, se os contêineres que estão acima dele forem removidos. Para o segundo objetivo, define-se como medida da instabilidade do navio em cada porto a distância do centro de massa ao centro geométrico (AVRIEL *et al.*, 2000). A seguir, será apresentada a formulação do problema PCCTP 3D como sendo um problema de programação linear inteira, com variáveis binárias 0-1. Esta formulação respeita as restrições operacionais relacionadas aos contêineres e ao navio e foi baseada na formulação bidimensional, que aparece em Avriel e Penn (1993), Botter e Brinati (1992) e em Avriel *et al.* (1998).

2.1. Modelo Matemático Binário

Considere um navio de transporte de contêineres que possui D baias numeradas, $d = 1, ..., D$. Cada baia tem R linhas horizontais numeradas $r = 1, 2, ..., R$, (a linha 1 é a linha que está em baixo e a linha R é a linha do topo da pilha) e C colunas verticais numeradas $c = 1, 2, ..., C$ (coluna 1 é a primeira coluna da esquerda). As baias têm então formato tridimensional e apesar delas poderem ser de tamanhos e capacidades diferentes, as mesmas podem ser representadas, sem perda de generalidade, como sendo baias de mesma capacidade. Deste modo, é possível representá-las por um vetor de matrizes. Assim, uma baia pode alocar, no máximo, $R \times C$ contêineres. É assumido também, que todos os contêineres têm o mesmo tamanho e peso. Suponha agora que o navio chega ao porto 1 completamente vazio e, sequencialmente, ele visita os portos 2, 3, ..., N. Em cada porto $i = 1, ..., N–1$, o navio recebe o carregamento de contêineres com destino aos portos $i+1, ..., N$. No último porto, ele descarrega os contêineres e fica totalmente vazio. Seja $T = [T_{ij}]$ a matriz de transporte de dimensão $(N–1)\times(N–1)$, onde T_{ij} é o número de contêineres, com origem em i e destino em j. A matriz é triangular superior, porque $T_{ij} = 0$ para todo $i \geq j$. Seja $x_{ijv}(r,c,d)$ a variável binária que assume o valor 1, se existir um contêiner no compartimento (r, c, d), que foi ocupado no porto i, com destino final o porto j e foi movimentado no porto v; caso contrário, a variável $x_{ijv}(r,c,d)$ assume valor zero. O índice v pode assumir os valores $i + 1, i + 2, ..., j – 1, j$. Se $v = j$, significa que o contêiner foi descarregado no porto de destino. Por outro lado, se $v < j$, o contêiner pode ter sido descarregado e, em seguida, carregado no mesmo porto. Uma vez que o custo de descarregamento dos contêineres no porto de destino não tem como ser minimizado, ele não está incluído na função objetivo. Por compartimento (r, c, d), entende-se a linha r, a coluna c e a baia d no compartimento de carga do navio. É necessário salientar que a numeração das linhas é feita de baixo para cima; assim, a linha de número 5 está acima da linha de número 4; a numeração das colunas é feita da esquerda para a direita e a profundidade é iniciada da popa até a proa. Seja agora, $y_i(r,c,d)$ a variável que possui valor 1, se ao sair do porto i o compartimento (r,c,d) for ocupado por um contêiner; e possui valor 0, no caso contrário. A formulação de programação linear inteira do PCCTP 3D é dada pelas eqs. (1)-(6).

$$\text{Min } f(x) = \qquad f(x) = \alpha\phi_1(x) + \beta\phi_2(y) \qquad i = 1, ..., N–1, j = i + 1, ..., N; \tag{1}$$

$$\text{s.a.} \quad \sum_{v=i+1}^{j}\sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{d=1}^{D} x_{ijv}(r,c,d) - \sum_{k=1}^{i-1}\sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{d=1}^{D} x_{kji}(r,c,d) = T_{ij} \qquad i = 1, ..., N–1, j = i + 1, ..., N; \tag{2}$$

$$\sum_{k=1}^{i}\sum_{j=i+1}^{N}\sum_{v=i+1}^{j} x_{kjv}(r,c,d) = y_i(r,c,d) \qquad \begin{array}{l} i = 1, ..., N–1, r = 1, ..., R; \\ c = 1, ..., C; \text{ d} = 1, ..., \text{D}; \end{array} \tag{3}$$

$$y_i(r,c,d) - y_i(r+1,c,d) \geq 0 \qquad \begin{array}{l} i = 1, ..., N–1, r = 1, ..., R–1; \\ c = 1, ..., C; \text{ d} = 1, ..., \text{D}; \end{array} \tag{4}$$

$$\sum_{i=1}^{j-1}\sum_{p=j}^{N} x_{ipj}(r,c,d) + \sum_{i=1}^{j-1}\sum_{p=j+1}^{N}\sum_{v=j+1}^{P} x_{ipv}(r+1,c,d) \leq 1 \qquad \begin{array}{l} i = 2, ..., N, r = 1, ..., R–1; \\ c = 1, ..., C; \text{ d} = 1, ..., \text{D}; \end{array} \tag{5}$$

$$x_{ijv}(r,c,d) = 0 \text{ ou } 1; \; y_i(r,c,d) = 0 \text{ ou } 1 \tag{6}$$

Nesta formulação, a restrição (2) é a restrição de conservação de fluxo, onde T_{ij} é o elemento da matriz de transporte, que representa o número de contêineres que embarcam no porto *i*, com destino ao porto *j*. A restrição (3) garante que cada compartimento (*r*, *c*, *d*) tem, no máximo, um contêiner. A restrição (4) é necessária para garantir que existem contêineres embaixo do contêiner que ocupa o compartimento (*r*, *c*, *d*). A restrição (5) é responsável por definir a movimentação dos contêineres: se um contêiner que ocupa a posição (*r*, *c*, *d*) é descarregado no porto *j*, então, ou não existem contêineres acima dele ou o índice *v* do contêiner que ocupa o compartimento (*r*+1, *c*, *d*) não é maior que *j*. A função objetivo da Eq. (1) é composta por dois critérios diferentes: o primeiro, $\phi_1(x)$, depende da movimentação dos contêineres e o segundo, $\phi_2(x)$, depende da posição ocupada pelos contêineres em cada porto. Os dois critérios podem ser combinados, através de valores escalares (α e β), que são utilizados como pesos dentro de um modelo de otimização bi-objetivo. Assumindo que a movimentação de um contêiner possui um custo unitário e igual para todos os portos, o termo $\phi_1(x)$, relativo ao custo total de movimentação dos contêineres em todos os portos, é dado pela Eq. (7).

$$\phi_1(x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{v=i+1}^{j-1} \sum_{r=1}^{R} \sum_{c=1}^{C} \sum_{d=1}^{D} x_{ijv}(r,c,d) \tag{7}$$

O termo $\phi_2(x)$ é relativo à estabilidade do navio. Para estimar a estabilidade do navio, foi empregado, como critério, a soma dos quadrados dos desvios entre as coordenadas do centro de massa ($xcm_{d,i}$, $zcm_{d,i}$) de cada baia e o centro geométrico da mesma. A estabilidade de cada baia é dada pela Eq.(8).

$$\phi_2(y) = \sum_{i=1}^{N} \sum_{d=1}^{D} (\Delta xcm_{d,i})^2 + \sum_{d=1}^{D} (\Delta zcm_{d,i})^2 \tag{8}$$

onde:

$$\Delta zcm_{d,i} = zcm_{d,i} - R/2,\ \Delta xcm_{d,i} = xcm_{d,i} - C/2 \quad \text{e}$$

$$zcm_{d,i} = \left(\sum_{r=1}^{R} \sum_{c=1}^{C} (y_i(r,c,d) \times (r - 0.5)) \right) \Big/ \left(\sum_{r=1}^{R} \sum_{c=1}^{C} y_i(r,c,d) \right)$$

$$xcm_{d,i} = \left(\sum_{c=1}^{C} \sum_{r=1}^{R} (y_i(r,c,d) \times (c - 0.5)) \right) \Big/ \left(\sum_{r=1}^{R} \sum_{c=1}^{C} y_i(r,c,d) \right)$$

Observe na equação 8, que para se calcular a soma dos desvios entre o centro de massa ($xcm_{d,i}$,$zcm_{d,i}$) e o centro geométrico ($R/2$,$C/2$) de cada baia, é necessário calcular as coordenadas do centro de massa $xcm_{d,i}$ e $zcm_{d,i}$, relativas respectivamente ao eixo das colunas *c* e linhas *r* de cada baia *d*.

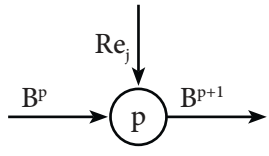
Na formulação apresentada pelas equações (1)-(6), para se representar uma solução, cuja instância tenha $D = 5$, $R = 6$, $C = 50$, e $N = 30$ portos, serão necessárias $D*R*C*N^3$ variáveis $x_{ijv}(r,c,d)$, ou seja, 40.500.000 variáveis *x*, e $D*R*C*N$ variáveis $y_i(r,c,d)$, ou seja, 45.000 variáveis *y*, somando um total de 40.545.000 variáveis para representar uma única solução. Deste modo, o problema assume um tamanho muito grande, tornando proibitivo o uso dessa formulação para problemas reais. Para resolver este problema, utilizamos uma representação que denominamos Representação por Regras, que diminui

consideravelmente o número de variáveis. Para se ter uma ideia de quanto é esta redução, a instância que no exemplo acima necessita de 40.545.000 variáveis, na formulação apresentada, necessitará de 29 variáveis ao se utilizar à Representação por Regras, ou seja, o número de variáveis é igual ao número de portos, menos 1. Além de ser compacta, a Representação por Regras tem a vantagem de assegurar que todas as soluções geradas pelo método utilizado sejam factíveis.

3. REPRESENTAÇÃO POR REGRAS

Em vez de usar o modelo matemático binário apresentado na seção 2, foi utilizado um conjunto de regras que definem, a cada porto, como o navio vai ser carregado ou descarregado. Isso pode ser melhor compreendido se o problema for representado por um grafo como mostrado na figura 2. Na figura 2, o nó representa o porto *p*, onde o navio está ancorado, esperando para ser descarregado e, em seguida, carregado. O estado do navio muda quando ele chega e parte do porto *p*, devido à maneira como ele é descarregado e depois carregado para seguir viagem.

Figura 2 – Representação, através de grafo, do estado do navio antes e depois do porto p e da decisão de como ele deve ser carregado e descarregado.



Fonte: Elaboração dos autores.

Esta mudança é representada pelo arco B^{p+1}. Então, o estado B^p vai se transformar no estado B^{p+1}, depois que for tomada a decisão Re_j de como o navio vai ser descarregado e recarregado para seguir para o porto *p*+1. Assim, a decisão Rej é composta de duas outras decisões, a saber:

- decisão Rd, que determina como o navio vai ser descarregado no porto *p*. Isto é, quais contêineres serão descarregados no porto *p*, porque seu destino é o porto *p* e quais contêineres serão descarregados em *p*, porque eles estão bloqueando contêineres cujo destino é *p*, ou porque ao descarregá-los em *p*, é possível obter uma melhor distribuição dos contêineres no navio e assim minimizar, movimentos desnecessários nos portos futuros.
- decisão Rc, que determina como carregar o navio, no porto *p*, com os contêineres cujo destino são os portos *p*+1, ..., N.

Observe que as decisões Rc e Rd têm grande influência no número de contêineres que serão movimentados em cada porto. Na vida real, o carregamento e descarregamento dos contêineres em cada porto, são feitos de acordo com a experiência e/ou regras práticas. Em outras palavras: o processo de carregamento e descarregamento segue regras existentes, em decorrência da prática do pessoal portuário. Essas regras podem ser transformadas em programas computacionais que as simulam, mostrando como ficaria o navio ao aplicá-las e deste modo, é possível escolher a melhor maneira de se resolver o PCCTP 3D.

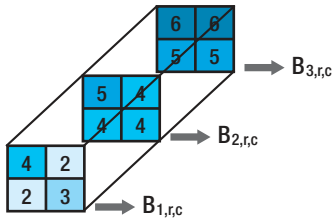
A representação do PCCTP 3D, através de regras, tem três vantagens em relação ao modelo matemático binário, a saber:

- ela permite a codificação compacta da solução, através de um vetor com (N-1) elementos, onde cada elemento guarda qual regra vai ser utilizada em cada porto.
- o número de variáveis do problema a ser resolvido é igual ao número de regras Re_j, para j=1, 2, ..., N-1, isto porque no porto N, só vai ocorrer descarregamento.
- a experiência profissional do pessoal portuário pode ser incorporada no processo de otimização do navio, através da simulação das regras que eles costumam utilizar.

As soluções obtidas através deste processo, são sempre factíveis, porque ao se obedecer ao esquema proposto pela figura 2, a factibilidade da restrição (2) está assegurada, uma vez que o arranjo dos contêineres, quando o navio sai de cada porto *p*, leva em conta o arranjo dos contêineres quando o navio chega no porto *p*. A incorporação das regras no modelo de simulação computacional assegura também, a factibilidade das restrições (3)-(6).

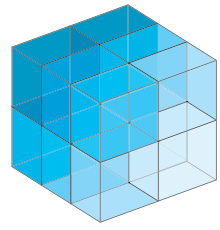
Na figura 1, viu-se que os navios possuem uma estrutura celular, de modo que os locais onde os contêineres serão alocados são pré-determinados, fazendo com que os contêineres sejam empilhados verticalmente. Este empilhamento sugere uma representação, por meio de um vetor de matrizes, dos contêineres no navio. Então, a variável de estado B^p é representada por uma matriz de ocupação B, tridimensional, que fornece a quantidade de espaços disponíveis e a localização dos contêineres no navio, em cada porto. Para tanto, cada elemento B_{drc} da matriz tridimensional B representa o estado de uma célula (*d*,*r*,*c*), isto é, se $B_{drc} = 0$, significa que a célula que ocupa a baia *d*, a linha *r* e a coluna *c*, está vazia e se $B_{drc} = j$ significa que a célula contém um contêiner, cujo destino é o porto *j*. Assim, no exemplo da figura 3, o elemento (1,2,1) pertencente a baia 1 ($B_{1,r,c}$), linha 2 e coluna 1, é igual a 4 significando que neste local existe um contêiner que será descarregado no porto 4. De modo análogo, o elemento (3,1,1) = 5 significa que a célula da terceira baia ($B_{3,r,c}$), primeira linha e primeira coluna contém um contêiner, cujo destino é o porto 5. Observe que a linha 2 representa o topo da pilha de carregamento e a linha 1 representa a parte inferior da pilha. Então, os elementos da matriz $B_{d,r,c}$ representam a ocupação do navio e são mostrados na figuras 3 e 4. Na figura 3, tem-se o detalhamento da matriz por baia, onde cada baia seria uma sequência de matrizes bidimensionais e na figura 4, tem-se o detalhamento tridimensional dos contêineres.

Figura 3 – Matriz B de ocupação para navios com capacidade de 16 contêineres e cargas para 6 portos.



Fonte: Elaboração dos autores.

Figura 4 – Representação tridimensional da Matriz de Ocupação B mostrada na figura 3.



Fonte: Elaboração dos autores.

O procedimento de resolução, aqui utilizado, trata o PCCTP 3D como um problema onde a matriz de ocupação B representa o arranjo dos contêineres no navio (B_p) antes de chegar no porto p. Esse arranjo é então modificado em cada porto, através da variável de decisão Re_j, que define a maneira (regra) como o navio será descarregado (regra de descarregamento R_c) e depois, carregado (regra de carregamento Rd).

A resolução do PCCTP 3D, usando a representação por regras, consiste em determinar qual regra Re_j vai ser utilizada em cada porto, para se fazer o carregamento e descarregamento dos contêineres, de modo a minimizar a quantidade de movimentos dos contêineres nos portos. Considerou-se como um movimento, cada operação de carregamento ou descarregamento de um contêiner num porto *p*. Aqui, foram estabelecidas algumas regras para se fazer o carregamento e o descarregamento em cada porto. A combinação de uma regra de carregamento Rc com uma regra de descarregamento Rd vai resultar na variável de decisão Re_j. As regras foram elaboradas, lembrando que, muitas vezes, para se fazer o descarregamento, no porto *p*, de um contêiner cujo destino é o próprio porto *p*, é necessário fazer operações de remanejamento (descarregamento seguido de carregamento) dos contêineres que já estão no navio e cujo destino são os portos de *p*+1 até N. Esta situação ocorre quando existem contêineres que ocupam, na pilha, uma posição acima da posição do contêiner que deve ser descarregado no porto *p*. Veja, por exemplo, que na matriz B da figura 4, para se descarregar os contêineres do porto 2, será necessário descarregar, pela ordem, os contêineres contidos nas células (1,2,1), (1,2,2) e (1,1,1). Observe que o destino do contêiner que ocupa a célula (1,2,1) é o porto 4, mas ele precisa ser remanejado no porto 2, para se descarregar o contêiner que está na célula (1,1,1). Assim, com intuito de reduzir o número de remanejamentos, ao se fazer o carregamento de contêineres num dado porto *j*, deve-se levar em conta os contêineres que já estão no navio, porque foram embarcados nos portos anteriores (portos de 1 até *j*–1), com destino aos portos *j*+1 até *N*, então se procurou estabelecer regras para o carregamento e descarregamento de contêineres, em cada porto, que leve em conta esta relação. Para tanto, foram criadas doze regras, sendo seis para o carregamento (Rc1, Rc2, Rc3, Rc4, Rc5, Rc6) e duas para o descarregamento (Rd1, Rd2). A combinação das regras Rc e Rd fornecem a regra Re_k para cada porto *p*. A tabela 1, mostrada a seguir, fornece as 12 regras Re_k a serem utilizadas nos portos p, oriundas da combinação das regras Rc e Rd. A aplicação das regras Re_k em cada porto *p*, vai atualizar a Matriz de Ocupação B no porto *p*. Vale lembrar que, inicialmente, no porto 1, a matriz B está com todos seus elementos iguais a zero.

Para ilustrar a utilização das regras, será empregada a matriz de transporte T, que fornece a quantidade de contêineres que devem ser embarcados em cada porto i, com destino a cada porto *j*, tal como dado na figura 5. A capacidade e as dimensões adotadas para o navio são as mesmas apresentadas na figura 3.

Tabela 1 – Regras Re_k , a serem utilizadas em cada porto *p*.

Regra de carregamento	Rc1	Rc1	Rc2	Rc2	Rc3	Rc3	Rc4	Rc4	Rc5	Rc5	Rc6	Rc6
Regra de descarregamento	Rd1	Rd2	Rd1	Rd2	Rd1	Rd2	Rd1	Rd2	Rd1	Rd2	Rd1	Rd2
Regra Re_k usada no porto *p*	Re_1	Re_2	Re_3	Re_4	Re_5	Re_6	Re_7	Re_8	Re_9	Re_{10}	Re_{11}	Re_{12}

Fonte: elaboração dos autores.

Na explicação do uso das regras foi suposto que o navio está no porto 1 para as regras de carregamento e porto 2 para as regras de descarregamento.

Figura 5 – Matriz de transporte T.

01	2	5	0	0
02	0	2	3	1
03	0	0	2	2
04	0	0	0	1

Fonte: elaboração dos autores.

3.1. Regras de Carregamento

Regra Rc1: Esta regra preenche a matriz de ocupação B (no porto *p*) por baia, começando pela primeira linha da primeira baia e indo até a última linha desta mesma baia, indo da esquerda para a direita, colocando na parte inferior da pilha de cada baia as cargas cujo destino é mais distante. A aplicação desta regra, considerando a matriz T da figura 5 e que o navio se encontra no porto 1, resultará na matriz B da figura 6(a).

Regra Rc2: Esta regra preenche a matriz de ocupação B (no porto *p*) por linha, começando pela primeira linha da primeira baia e indo até a última baia, da esquerda para a direita, colocando na parte inferior da pilha de cada linha as cargas cujo destino é mais distante. A Aplicação desta regra, considerando a matriz T da figura 6 e que o navio se encontra no porto 2, resultará na matriz B da figura 6(b).

Regra Rc3: Esta regra é o espelho da regra Rc1. No porto *p*, a matriz de ocupação B (vide figura 6(c)) será preenchida por baia, da direita para a esquerda, colocando na parte inferior da pilha de cada baia, as cargas cujo destino é mais distante

Regra Rc4: Esta regra é o espelho da regra Rc2. No porto *p*, a matriz de ocupação B (vide figura 6(d)) será preenchida por linha, da direita para a esquerda, colocando na parte inferior da pilha de cada baia, as cargas cujo destino é mais distante.

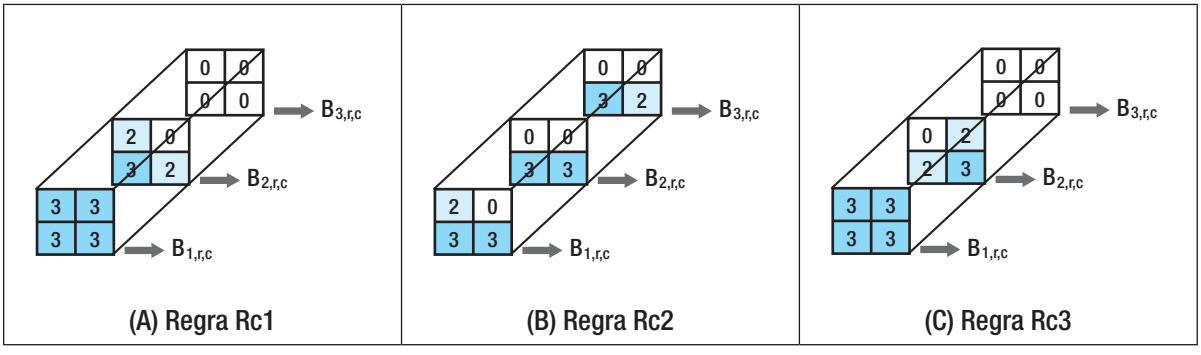
Regra Rc5: Nesta regra, o preenchimento da matriz de ocupação B em um porto *p*, é feito por baia até uma linha θ*p*, começando pela coluna da esquerda e colocando-se em primeiro lugar os contêineres cujos destinos são os portos mais distantes. A linha θ*p* é calculada, pegando-se a função teto, resultante da soma do total de contêineres que estavam no navio e foram embarcados nos portos anteriores; menos a quantidade de contêineres que serão desembarcados em *p*, mais a quantidade de contêineres a serem embarcados em *p*, dividido pelo número de baias da matriz B. O número dessa linha pode ser calculado sobre a matriz de transporte T, através da Equação (9), onde *p* é o porto atual do navio, T_{ij} é o número total de contêineres a serem embarcados no porto *i*, com destino ao porto *j* e *D* é o número total de baias da matriz B de ocupação do navio.

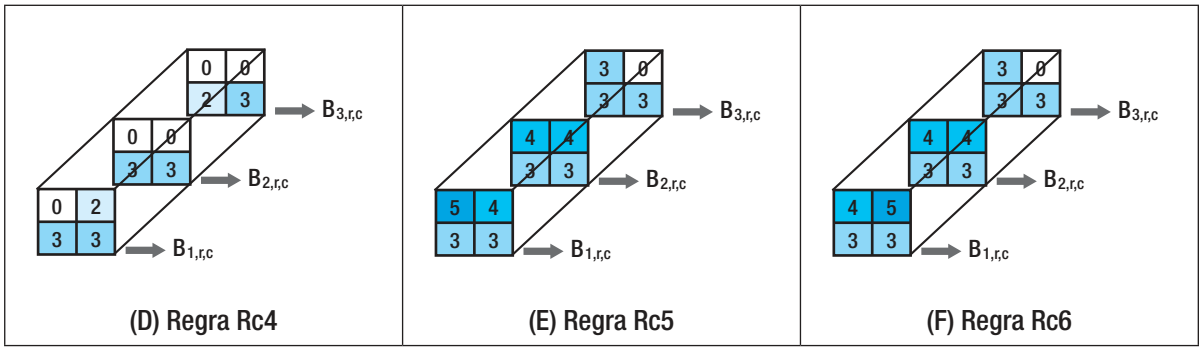
$$\theta p = \left\lceil \frac{\sum_{i=1}^{p} \sum_{j=p+1}^{N} T_{ij}}{D} \right\rceil \tag{9}$$

Suponha, agora, que quando o navio chega ao porto 2, ele esteja com a matriz B igual a da figura 6(b) e que os contêineres cujo destino é o porto 2, são descarregados. Aplicando-se a regra Rc5 para carregar o navio antes dele ir para o porto 3, a matriz B resultante é a da figura 6(E), pois no porto 3, são carregados 2 contêineres cujo destino é o porto 3, 3 contêineres cujo destino é o porto 4 e um contêiner cujo destino é o porto 5.

Regra Rc6: Esta regra, também, é o espelho da regra Rc5. Ela faz o preenchimento da matriz de ocupação B, em um porto *p*, preenchendo cada coluna até a linha θ*p* (Equação (9)), começando pela coluna da direita e colocando-se em primeiro lugar os contêineres, cujos destinos são os portos mais distantes, conforme figura 6(F).

Figura 6 – Matriz de Ocupação no porto 2, após a aplicação de cada regra Rc.





(D) Regra Rc4 (E) Regra Rc5 (F) Regra Rc6

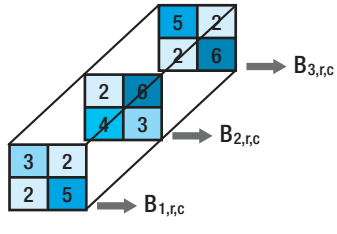
Fonte: Elaboração dos autores.

3.2. Regras de Descarregamento

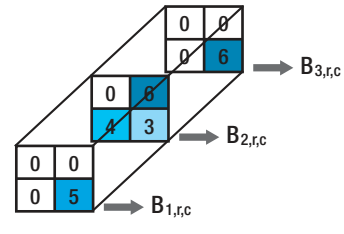
Regra Rd1: Nesta regra, quando o navio chega a um porto *p*, são removidos todos os contêineres cujo destino é *p* e todos os contêineres que estão acima dos contêineres do porto *p* e cujos destinos são os portos *p*+*j*, para *j* = 1, ..., *N* tal que (*p*+*j*) < = *N*. Suponha, por exemplo, que ao se chegar ao porto 2, a matriz de Ocupação B seja a representada na figura 7(a). De acordo com esta regra, a matriz B ficaria como mostrada na figura 7(b). Além disso, após a aplicação da regra, a matriz de transporte T tem que ser atualizada, porque foram retirados um contêiner com destino ao porto 3 e um contêiner com destino ao porto 5, que devem ser recarregados e, portanto, eles serão adicionados na matriz de transporte relativa ao porto 2. As figuras 7(c) e 7(d) mostram a matriz de transporte, antes e depois da aplicação desta regra.

Regra Rd2: Nesta regra, quando o navio chega ao porto *p*, todos os contêineres são removidos para permitir que todas as pilhas sejam reordenadas por alguma regra de carregamento, a ser aplicada posteriormente. Assim como foi feito na regra Rd1, os contêineres removidos cujo destino é para portos além do porto 2, deverão ser contabilizados na matriz de transporte T.

Figura 7 – Matriz de Ocupação no porto 2, antes e depois da aplicação da regras Rd1.



(a) Antes da aplicação da regra Rd1



(b) Depois da aplicação da regra Rd1

	D2	D3	D4	D5	D6
02	0	2	3	1	0

(c) Matriz transporte antes de aplicação de Rd1

	D2	D3	D4	D5	D6
02	0	3	4	2	0

(d) Modificação matriz transporte após Rd1

Fonte: Elaboração dos autores.

As vantagens do emprego de regras são: a facilidade de incorporar conhecimento prévio do decisor, sob a forma de regras; elas sempre produzem matrizes de Ocupação factíveis, facilitando e garantindo a obtenção de soluções factíveis por métodos heurísticos; e a codificação da solução que determina como será realizado o carregamento e o descarregamento de um navio para N portos, é um vetor de tamanho N–1. Por fim, a representação por regras fornece uma representação mais compacta da solução do problema, se comparada com outras abordagens da literatura, como por exemplo, a utilizada em Avriel *et al.* (2000), reduzindo drasticamente o número de variáveis utilizadas para representar uma solução, isto é, na representação por regras, o número de variáveis é igual ao número de portos menos um. A solução do PCCTP 3D foi feita através de uma modificação do procedimento apresentado em Azevedo *et al.* (2010), para retornar não só o número de movimentos, mas também, a medida de estabilidade. Portanto, o mesmo Algoritmo Genético utilizado na solução do PCCTP 2D, descrito em Azevedo *et al.* (2010), é empregado na solução do PCCTP 3D, bastando realizar uma única modificação no processo de simulação, que agora passa a considerar uma representação mais realista do navio (3D) e fornece, combinados em uma única função, o número de movimentos e a medida de estabilidade (vide Equações (1), (7) e (8)). Ou seja, o processo de avaliação de uma solução s combina avaliação das regras com a simulação do estado do navio, ao longo dos portos, após a aplicação de cada regra, tal como dado na figura 8.

Figura 8 – Algoritmo para avaliação de uma solução s por meio de regras.

```
Avaliação de Solução
Início
  p ← 1, nmov ← 0, instab ← 0.0
  inicializar(B,T)
  enquanto (p < N) faça
  Início
    [rc, rd] = extrairRegras(s(p))
    Se (p > 1)
      instab ← instab + calcDXDZ(B)
      [aux, B, T] = descarregar(rd, B, p)
      nmov← nmov + aux
    fim
    Se (p < N-1)
      [aux, B, T] = carregar(rc, B, T, p)
      instab ← instab + calcDXDZ(B)
      nmov ← nmov + aux
    fim
    p ← p + 1
    retornar alfa*nmov + beta*instab
  fim
Fim
```

Fonte: Elaboração dos autores.

Os símbolos e funções empregados no algoritmo da figura 9 são descritos a seguir:

p: variável contadora, que indica o porto atual da simulação.

N: número total de portos.

S: Vetor tal que o elemento s(i) contém a regra k, a ser aplicada no porto i e modificar a matriz B adequadamente.

Nmov: número de movimentos realizados para carregar ou descarregar o navio ao longo dos N portos.

B: matriz de ocupação que indica o estado do navio em cada porto i.

Rc: variável que contém o nome da regra de carregamento a ser aplicada.

Rd: variável que contém o nome da regra de descarregamento a ser aplicada.

inicializar: função que preenche a matriz B, com valores iguais a zero.

extrairRegras: função que define a correspondência entre a regra k, contida em s(i), com as regras de descarregamento e carregamento a serem armazenadas nas variáveis rd e rc, respectivamente.

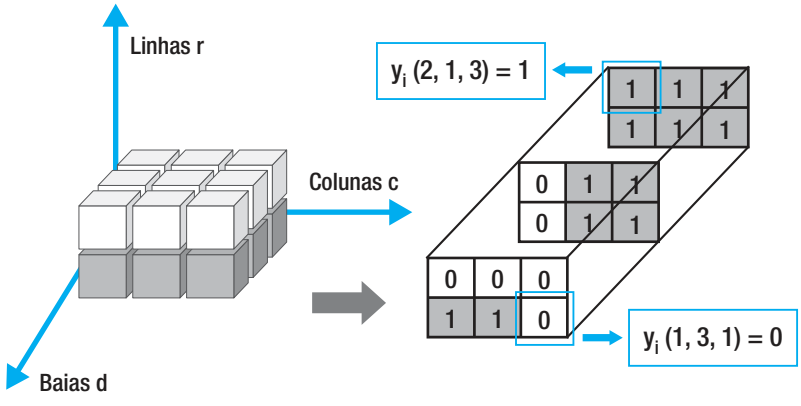
descarregar: função que aplica a regra de descarregamento contida em rd na matriz B no porto i, e retorna o número de movimentos realizados, e Be T atualizadas.

carregar: função que aplica a regra de carregamento contida em rc, na matriz B, no porto i e retorna o número de movimentos realizados e B e T atualizadas.

calcDXDZ: função que realiza o cálculo da distância entre o centro de massa e o centro geométrico por baia d, para o navio, antes de ser descarregado e depois de ser carregado em cada porto p.

As funções carregar e descarregar, empregadas no algoritmo da figura 8, utilizam as regras descritas nas seções 3.1 e 3.2, respectivamente. Para melhor ilustrar o funcionamento do procedimento calcDXDZ, suponha que o navio tenha o arranjo de contêineres, tal como ilustrado na figura 9.

Figura 9 – Ocupação do navio porta-contêiner e sua matriz de estado B correspondente.



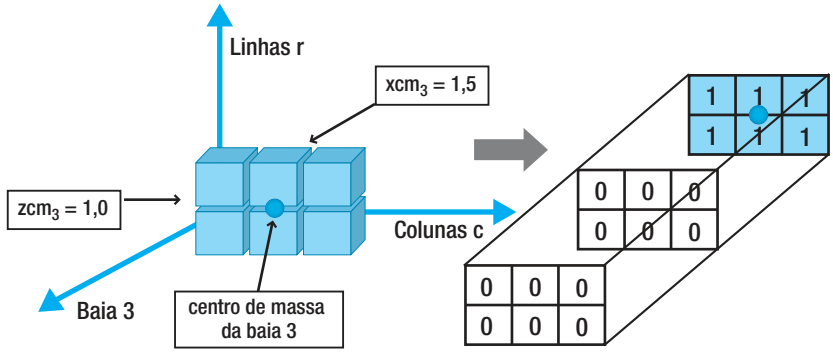
Fonte: elaboração dos autores.

Definindo xcm_3 como a coordenada do centro de massa relativa às linhas da baia e zcm_3 a coordenada do centro de massa relativa às colunas da baia 3. Os valores xcm_3 e zcm_3 podem ser calculados, usando a Eq. (8) e a informação contida na figura 9, como dado na figura 10.

$$xcm_3 = (2 \times 0{,}5 + 2 \times 1{,}5 + 2 \times 2{,}5)/(2 + 2 + 2) = 9/6 = 1{,}5$$

$$zcm_3 = (3 \times 0{,}5 + 3 \times 1{,}5)/(6) = 6/6 = 1{,}0$$

Figura 10 – Cálculo das coordenadas do centro de massa da baia 3.



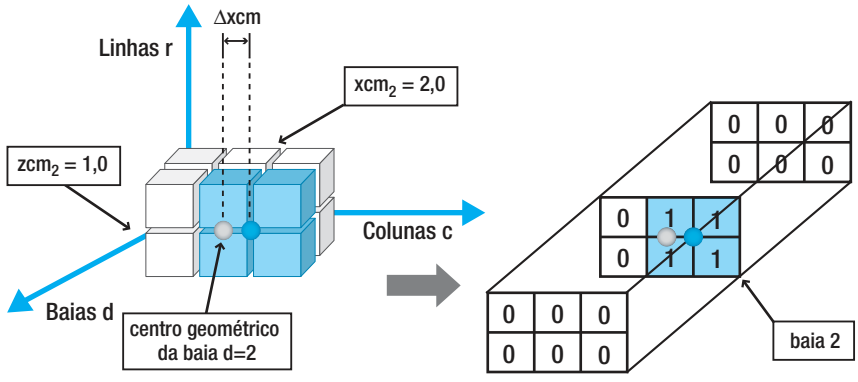
Fonte: Elaboração dos autores.

Cálculos semelhantes podem ser realizados para a baia 2, para a obtenção de xcm_2 e zcm_2 empregando-se a Eq. (8) e a informação contida na figura 11.

$$xcm_2 = (0 \times 0{,}5 + 2 \times 1{,}5 + 2 \times 2{,}5)/(0 + 2 + 2) = 8/4 = 2{,}0$$

$$zcm_2 = (2 \times 0{,}5 + 2 \times 1{,}5)/(4) = 4/4 = 1{,}0$$

Figura 11 – Cálculo das coordenadas do centro de massa da baia 2.



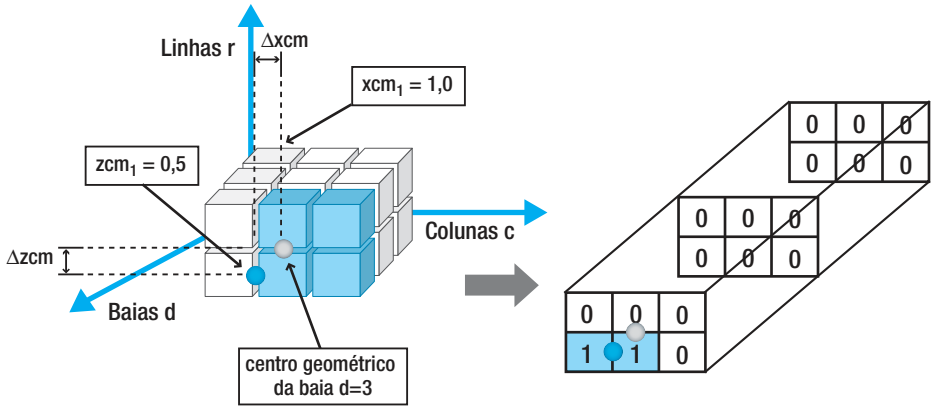
Fonte: Elaboração dos autores.

Finalmente, para a baia 1, as coordenadas xcm_1 e zcm_1 podem ser calculadas, usando-se a Eq. (8) e a figura 12, como dado a seguir:

$$xcm_1 = (1 \times 0{,}5 + 1 \times 1{,}5 + 0 \times 2{,}5)/(1 + 1 + 0) = 2/2 = 1{,}0$$

$$zcm_1 = (2 \times 0{,}5 + 0 \times 1{,}5)/(2 + 0) = 1/2 = 0{,}5$$

Figura 12 – Cálculo das coordenadas do centro de massa da baia 1.



Fonte: Elaboração dos autores.

De posse dos cálculos anteriores, é possível obter a medida de estabilidade para o navio, empregando a Eq. (8) e a configuração de contêineres apresentada na figura 9 e como mostrado a seguir:

$$\phi_2(x) = ((1{,}5 - 1{,}5)^2 + (2{,}0 - 1{,}5)^2 + (1{,}0 - 1{,}5)^2) + ((1{,}0 - 1{,}0)^2 + (1{,}0 - 1{,}0)^2 + (0{,}5 - 1{,}0)^2) = 0{,}75$$

4. RESULTADOS OBTIDOS

Para testar o algoritmo, foram geradas automática e aleatoriamente 15 instâncias e elas estão disponíveis em https://sites.google.com/site/projetonavio/home. As instâncias estão classificadas de acordo com o número de portos e o tipo da matriz de transporte. Para cada instância é gerada uma matriz de transporte T, tal que a capacidade do navio não será excedida em nenhum porto, isto é, o valor de θp, dado pela equação (9), deve ser menor ou igual a D×R×C para todo porto *p*, isto porque a matriz de transporte é factível se:

$$\sum_{i=1}^{p} \sum_{j=p+1}^{N} T_{ij} \leq D{\times}R{\times}C \text{ para todo p = 1, ..., N} \qquad (10)$$

De acordo com Avriel *et al.* (1998), podem ser gerados três tipos de matriz de transporte: 1-Mista, 2-Longa distância e 3-Curta distância. Uma matriz do tipo 3 refere-se ao transporte de contêineres que vão percorrer poucos portos, antes de serem desembarcados. Já uma matriz do tipo 2, refere-se a contêineres que vão percorrer muitos portos antes de serem desembarcados. Uma matriz do tipo mista combina os dois tipos anteriores. As instâncias foram classificadas de acordo com a quantidade de portos a serem percorridos, o tipo da matriz de transporte e a capacidade do navio. Neste trabalho, foi suposto um navio com as seguintes dimensões: (D×R×C) = 5 x 6 x 50, resultando na capacidade máxima de 1500 contêineres.

A tabela 2 mostra os resultados obtidos com a melhor solução, após 1 rodada de Algoritmo Genético, com população de 10 indivíduos, 1000 gerações, taxa de crossover igual a 80% e taxa de mutação igual a 15% para as 15 instâncias. Nesta tabela, a coluna I contém o número da instância; M: o tipo da matriz de transporte; N: o número de portos; NMin: o número mínimo de movimentos a serem realizados com os contêineres; F.O.1: os valores da função objetivo em termos do número total de movimentos realizados com os contêineres; F.O.2: os valores de estabilidade, de acordo com o arranjo de contêineres e com a equação (8); e T: tempo computacional em segundos. Note-se que os valores de F.O.1, F.O.2 e T são apresentados para dois pares de α e β: (α=1, β=0) e (α=0, β=1). Note-se, ainda, que os resultados apresentados para (α=1, β=0) correspondem à melhor solução obtida com o objetivo de minimizar o número de movimentos para cada instância e que para sete instâncias, o número de movimentos (FO1) encontrado é até 2% maior que o número mínimo (Nmin); para 5 instâncias, o número mínimo é até 10% maior e apenas 3 instâncias até 14%. Já para (α=0, β=1) como o objetivo é minimizar a medida de instabilidade, apresenta maior número de movimentos (FO1), mas soluções com medida de instabilidade significativa menor (FO2). Estes dados confirmam que a minimização do número de movimentos e da instabilidade são objetivos conflitantes. O número mínimo de movimentos é obtido, multiplicando-se por dois o valor do somatório de T_{ij} calculado de acordo com o termo do lado esquerdo da equação (7). Observe-se que esse valor é o limitante inferior para o número total de movimentos a serem realizados ao longo do percurso do navio. Os resultados do *GA* foram obtidos com um programa desenvolvido em Matlab 7.0 e executado num computador Intel Core 2Duo 2.20 GHz (E4500), 2GB RAM, Windows XP Versão 2002 (SP 3).

Tabela 2 – Resultados do Algoritmo Genético para dois pares de valores de α e β.

I	N	M	NMin	(α=1, β=0)			(α=0, β=1)		
				F01	F02	T	F01	F02	T
1	10	1	6994	7072	13646.75	146.63	10106	3413.14	152.09
2	10	2	4172	4214	11531.00	139.57	6374	354.93	145.87
3	10	3	17060	17116	21880.21	149.63	18018	13369.83	153.14
4	15	1	9974	10584	11130.51	234.00	12766	3129.49	241.33
5	15	2	4824	5030	16167.29	210.34	9078	2178.01	232.31
6	15	3	24908	25046	36803.35	221.73	25844	23697.36	258.25
7	20	1	10262	10802	12997.90	300.11	14658	6480.20	360.50
8	20	2	4982	5500	18272.35	277.82	9784	1720.70	317.05
9	20	3	32602	32638	58015.05	279.01	33700	50114.92	330.84
10	25	1	11014	11848	18788.70	353.65	17302	6427.01	558.95
11	25	2	5002	5466	28097.92	322.62	10564	7049.75	440.53
12	25	3	43722	44082	68385.30	387.19	44964	46835.36	455.14
13	30	1	11082	12580	14476.66	476.60	16556	5984.56	513.86
14	30	2	4720	5312	10344.12	381.72	7098	911.36	402.42
15	30	3	53592	54398	69052.80	461.90	56290	65026.85	472.50

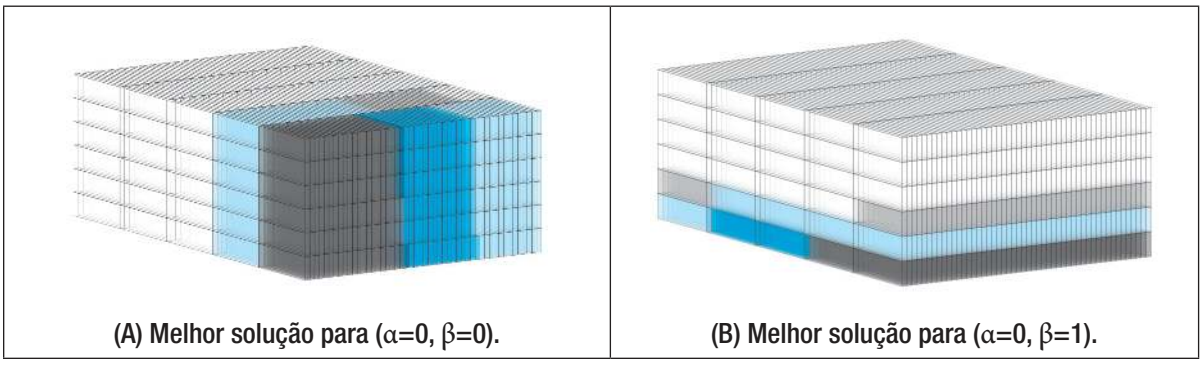
Fonte: elaboração dos autores.

A primeira observação importante acerca dos resultados da tabela 2, é sobre o tempo computacional gasto pelo algoritmo genético. É importante observar que para a formulação dada pelas equações (1)-(8), as instâncias com 30 portos, 5 baias, 6 linhas e 50 colunas são problemas com 40,545,000 variáveis inteiras. Para estas instâncias, o algoritmo genético consegue produzir boas soluções em menos de 8 minutos. Pode-se observar também que, de uma instância para outra, o aumento de 5 no número de portos a serem percorridos produz, em média, um aumento de mais ou menos 1 minuto e 30 segundos no tempo computacional. Por exemplo, em instâncias com 10 portos, levam-se 2 minutos e 30 segundos para se obter uma solução e em instâncias com 15 portos, leva-se, em média, 4 minutos. Espera-se ainda, reduzir o tempo computacional, através da implementação do algoritmo genético em linguagem C.

Os resultados indicam que para as instâncias com (α=1, β=0), onde se privilegia a minimização do número de movimentos e cuja matriz de transporte é do tipo curta distância (tipo 3), as regras são bastante adequadas e produzem resultados muito próximos do limite inferior do número de movimentos e, em alguns casos, chegam a quase atingir este limite. Este é o caso das instâncias com 10, 15 e 20 portos, que obtiveram soluções com distância da 0.33%, 0.55% e 0.11%, da solução ótima. Já para as instâncias em que a matriz de transporte é do tipo média distância (tipo 1) e longa distância (tipo 2), o algoritmo genético apresentou soluções, cujo número de movimentos é ligeiramente maior que o limitante inferior. Estes resultados indicam a necessidade de se incorporar ao sistema um número maior de regras, que levem em consideração a arrumação de contêineres que permanecerão um longo período de tempo dentro do navio. Quando a função objetivo privilegiou a medida de instabilidade (α=0, β=1), as soluções apresentadas pelo GA tiveram um aumento razoável no número de movimentos, mas com uma significativa melhora na medida de instabilidade. Em algumas instâncias, essa medida chegou a ser até 10 vezes menor, em comparação com a solução encontrada na minimização do número de movimentos (instância 14).

As figuras 13 fornecem uma visualização da organização dos contêineres, no porto 2, para a melhor solução encontrada para a instância I = 1, considerando (α=1, β=0) e (α=0, β=1). Ela ilustra bem a natureza multiobjetivo do problema.

Figura 13 – Ocupação do navio porta-contêiner no porto 2 para as duas melhores soluções obtidas com valores distintos de pesos.



(A) Melhor solução para (α=0, β=0).

(B) Melhor solução para (α=0, β=1).

Fonte: Elaboração dos autores.

5. CONCLUSÕES E TRABALHOS FUTUROS

Este artigo apresentou, pela primeira vez, uma nova representação de soluções para o problema de carregamento de contêineres 3D, em terminais portuários (PCCTP 3D). Esta nova representação permite reduzir a quantidade de informações necessárias para se representar uma solução tal que soluções próximas do limitante inferior do número de movimentos, podem ser obtidas em pequeno tempo computacional.

O artigo tem três inovações, a saber: a primeira delas é que ele apresenta uma formulação tridimensional para o problema de carregamento de contêineres em terminais portuários (PCCTP 3D). Esta representação permite considerar as medidas de instabilidade, acerca do arranjo dos contêineres no navio, juntamente com o número de movimentos para carregar e descarregar um navio. A segunda inovação é o uso da Representação por Regras, cuja aplicação teve êxito no problema 2D, e agora, o êxito se repete no problema 3D, reduzindo consideravelmente o número de variáveis do problema, diminuindo, assim, a quantidade de informações necessárias para se representar uma solução. Na formulação dada pelas Eqs. (1)-(8), as informações devem ser armazenadas num vetor de tamanho $(D\times R\times C)\times(N+N^3)$. Na representação, aqui utilizada, o tamanho do vetor é $N-1$. Tendo em vista que $D\times R\times C$ expressa o número de contêineres que podem ser armazenados em um navio e que N representa o número de portos, a representação da solução, aqui utilizada, permite uma enorme redução do tempo computacional para instâncias com grande número de portos. Para tanto, a representação emprega regras que definem como serão carregados e descarregados o navio e essas regras garantem que as soluções obtidas são sempre factíveis. A terceira inovação é que a utilização de regras de carregamento e descarregamento permite a incorporação, no sistema computacional, do conhecimento do profissional responsável por esse serviço no porto. Esta incorporação é feita através da criação e implantação de novas regras. Esta maneira de representar o PCCTP 3D, em conjunto com a aplicação de regras, é uma alternativa promissora na resolução deste problema, até porque assim como se podem acrescentar novas regras, também se podem utilizar outras heurísticas, como por exemplo, *Simulated Annealing*, VNS, ILS, Busca Tabu, dentre outras. Futuramente, pretende-se testar uma abordagem multiobjetivo, na obtenção de soluções e testes com outras medidas de instabilidade do navio.

6. AGRADECIMENTOS

Agradecimentos ao CNPq e à FAPESP pelo apoio financeiro através dos processos 2009/15107-0, 2010/51274-5, 2010-16517-4, 2010/16622-2 e 2011/01667-3.

7. REFERÊNCIAS BIBLIOGRÁFICAS

AMBROSINO, D.; SCIOMACHEN A.; TANFani, E. A decomposition heuristics for the container ship stowage problem, **J. Heuristics**, v.12, p. 211–233, 2006.

Avriel, M.; PENN, M. Container ship stowage problem, **Computers and Industrial Engineering**, v. 25, p. 271-274, 1993.

AVRIEL, M.; PENN, M.; SHPIRER, N.; WITTENBOON, S. Stowage planning for container ships to reduce the number of shifts, **Annals of Operations Research**, v. 76, p. 55-71, 1998.

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

AZEVEDO, A. T. ; RIBEIRO, C. M. ; DEUS, N. M. R. (2010) Resolução do Problema de Carregamento e Descarregamento de Contêineres em Terminais Portuários via Algoritmo Genético. **Revista INGEPRO**, v. 2, p. 38-51, 2010.

BOTTER, R. C. ; BRINATI, M. A. Stowage Container Planing: a model for getting optimal solution. *In*: **International Conference of Computer Applications in the Automation of Shipyard Operation and Ship Design**, 1992, Rio de Janeiro, 1991. p. 217-229.

WILSON, I.; ROACH, P. Container stowage planning: a methodology for generating computerised solutions, **Journal of the Operational Research Society**, v. 51, p. 1248-1255, 2000.