

UMA APLICAÇÃO DAS METAHEURÍSTICAS ALGORITMO GENÉTICO E COLÔNIA ARTIFICIAL DE ABELHAS ATRAVÉS DA CODIFICAÇÃO POR REGRAS PARA RESOLVER O PROBLEMA DE CARREGAMENTO DE NAVIOS-CONTÊINERES

Luziana F. Carraro¹, Leonardo D. Chiwiacowsky², Arthur T. Gómez³
Universidade do Vale do Rio dos Sinos (UNISINOS - PIPCA)

Av. Unisinos, 950, São Leopoldo/RS

¹luzianacarraro@gmail.com; ²ldchiwiacowsky@unisinos.br; ³breno@unisinos.br

Alexandre Cesar Muniz de Oliveira

Universidade Federal do Maranhão (UFMA – DEINF/PPGCC)

Av. dos Portugueses, 1966, São Luís/MA

acmo@deinf.ufma.br

RESUMO

Com a expansão do transporte marítimo, adotou-se o uso de contêineres para o transporte de cargas, sendo evidenciados alguns problemas. Dentre eles, destaca-se o problema de carregamento e descarregamento de contêineres em navios, em virtude dos altos custos operacionais gerados a partir da movimentação de contêineres. O objetivo deste trabalho é a elaboração de planos de carga eficientes que gerem um número mínimo de movimentações de contêineres, nas operações de carga e descarga realizadas em terminais portuários que constituem as rotas dos navios. Para a determinação de planos de carga adequados, é proposta a aplicação de duas diferentes metaheurísticas, Algoritmo Genético e Colônia Artificial de Abelhas, utilizando a codificação da solução baseada em regras de carregamento e descarregamento. O emprego de novas regras, em complemento àquelas propostas na literatura, trouxeram bons resultados. Assim, foram obtidas soluções de boa qualidade e melhores que aquelas encontradas na literatura, abordando o mesmo problema.

PALAVRAS CHAVE. Otimização Combinatória, Algoritmo Genético, Colônia Artificial de Abelhas.

Área principal (Logística e Transporte, Metaheurísticas, Otimização Combinatória)

ABSTRACT

With the expansion of maritime transportation, the use of containers for goods transportation has increased, being evidenced some problems. Among these problems, the container ship stowage problem arose as one of the main problems due to the high operational costs related to movement of containers. The objective this work is the formulation of efficient stowage plans that generate a minimum number of container shiftings in the operations of loading and unloading performed in port calls of container ships. In order to determine a suitable stowage plan, the application of Genetic Algorithm and Artificial Bee Colony metaheuristics are proposed to solve the problem by using a rule-based encoding for the solution. The use of new rules of loading and unloading, in addition with those proposed in literature, has produced good solutions. Thereby, good quality solutions were achieved and also better than that found in the literature which discuss the same problem.

KEYWORDS. Combinatorial Optimization, Genetic Algorithm, Artificial Bee Colony.

Main area (Logistics and Transport, Metaheuristics, Combinatorial Optimization)

1. Introdução

O transporte marítimo mercante há muitos séculos vem sendo utilizado como meio de transporte de mercadorias e de passageiros. Desde sua origem revelou-se importante pela realização de transações comerciais de grande porte, fazendo ligações entre a produção e o consumo. Porém, em meados dos anos 1960, o transporte marítimo revolucionou com a adoção de contêineres para organização e transporte de cargas, tornando-se mais fácil, mais seguro, rápido e ainda proporcionando garantia ao transporte.

Com o passar do tempo, houve a expansão do transporte marítimo e o comércio global tornou-se cada vez mais competitivo com o uso de contêineres. A vantagem na utilização de navios-contêineres está relacionada a sua capacidade de carga e à rapidez nas operações de carga e descarga, realizadas em terminais portuários adequados, em virtude da movimentação de carga unitizada.

Com o aumento crescente do emprego de navios-contêineres no transporte marítimo, diversos problemas começaram a surgir nos portos, que afetaram os custos operacionais e a qualidade produtiva. Dentre estes problemas, destaca-se a necessidade de um planejamento avançado para as atividades de carga e descarga desta classe de navios. O objetivo deste planejamento é a minimização de custos, principalmente aqueles relacionados à movimentação de contêineres, uma vez que a realização de remoções desnecessárias gera um custo adicional de movimentação e de tempo de operação.

O problema de planejamento das operações de carga e descarga de navios-contêineres, ou simplesmente problema de carregamento de contêineres, é classificado como um problema de otimização combinatória de complexidade computacional NP-Completo (Botter e Brinati, 1992; Avriel *et al.*, 2000). A elaboração de planos de estiva visa definir a ordem de descarregamento e carregamento dos contêineres em um navio, de modo a minimizar a necessidade de remoções temporárias, em virtude do bloqueio de contêineres que devem ser descarregados, e posterior realocação nas pilhas ainda presentes a bordo. Em virtude da grande capacidade de carga dos navios-contêineres atuais, torna-se inviável a enumeração de todas as soluções possíveis, sendo necessário o emprego de técnicas heurísticas de solução, (Sciomachen e Tanfani, 2007; Wilson e Roach, 1999).

Há um conjunto de trabalhos na literatura que propõem o emprego de diferentes técnicas para solução do problema de carregamento de navios-contêineres, assumindo diferentes restrições e propondo modelagens variadas (Avriel *et al.*, 1998; Wilson e Roach, 1999; Dubrovsky *et al.*, 2002; Azevedo *et al.*, 2010). Neste trabalho, é proposta a solução do problema de minimização de movimentação de contêineres, quando da operação de carga e descarga de navios-contêineres, através do emprego de dois métodos distintos, as metaheurísticas Algoritmo Genético e Enxame de Abelhas.

Observa-se que, com relação às técnicas adotadas, foram encontrados na literatura somente trabalhos empregando o Algoritmo Genético para solução do problema em estudo, sendo então o método do Enxame de Abelhas uma das contribuições deste trabalho. Ademais, destaca-se o emprego de regras de carregamento e descarregamento de contêineres para representação das soluções como outra contribuição do trabalho. A adoção de regras para codificação de soluções para o problema de carregamento de navios-contêineres foram inicialmente propostas por Azevedo *et al.* (2010). Em virtude do conjunto reduzido de regras adotadas, os resultados apresentados em (Azevedo *et al.*, 2010) mostraram-se promissores porém ainda distantes do ideal, isto é, da solução ótima. Em função disso, no presente trabalho são propostas novas regras de carregamento e descarregamento, em adição àquelas apresentadas em (Azevedo *et al.*, 2010). A adoção deste novo conjunto de regras possibilitou a obtenção de soluções de melhor qualidade.

Este trabalho está estruturado em cinco seções. Além desta primeira seção, na seção 2 é apresentada o modelo matemático para o problema de carregamento de navios-contêineres e sua codificação através do emprego de regras; na seção 3 são detalhadas as metaheurísticas empregadas na solução, seguidas pela definição dos operadores e parâmetros empregados nos algoritmos propostos. Na seção 4 são apresentados os resultados dos experimentos realizados

para um conjunto de problemas teste e, por fim, na seção 5 são apresentadas as considerações finais deste trabalho, sendo destacados os principais resultados e contribuições.

2. Modelagem do Problema

Para modelagem do problema, assume-se que o navio-contêiner possui apenas uma baía para armazenamento, uma vez que não serão consideradas restrições relativas à estabilidade do navio. Esta baía possui linhas horizontais identificadas por $r = 1, \dots, R$, onde o valor 1 indica a parte inferior do navio e R indica o topo. As linhas verticais são identificadas por $c = 1, \dots, C$, onde o valor 1 indica esquerda e C indica direita. Desta forma, a quantidade máxima que a baía poderá alocar é de $R \times C$ contêineres. Para a realização da operação de descarregamento, os contêineres, que estão dispostos em pilhas, são desembarcados exclusivamente pelo topo.

O navio inicia a sua rota partindo do Porto 1 e estando vazio. Em seguida, visita os demais portos da rota, isto é os portos 2, 3, ..., N . Em cada porto $i = 1, \dots, N - 1$, o navio carrega e descarrega contêineres com destino $j = i + 1, \dots, N$. Quando chegar ao último porto do trajeto, o navio deve ficar vazio. O modelo define também a matriz de transporte, $T = [T_{ij}]$ de tamanho $(N - 1) \times (N - 1)$, onde T_{ij} representa a informação relativa à quantidade de contêineres com origem no porto i e destino no porto j (Avriel *et al.*, 1998).

Assim, define-se uma variável binária $x_{ijv}(r, c)$, que pode assumir valor igual a 1 na posição (r, c) se for carregado um contêiner no Porto i com destino em j e movimentado no porto v , e 0 em caso o contrário. De maneira similar, define-se a variável $y_i(r, c) = 1$, se, partindo do porto i , a posição (r, c) estiver ocupada por um contêiner, e 0 se não estiver.

Sem perda de generalidade, assume-se que o custo de movimentação por contêiner é o mesmo para qualquer contêiner e em qualquer porto. Assim, o objetivo é minimizar o número de movimentações de contêineres, com base no seguinte modelo de programação linear binária:

A Função Custo é definida conforme a equação (1),

$$f(x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \sum_{v=i+1}^{j-1} \sum_{r=1}^R \sum_{c=1}^C x_{ijv}(r, c), \quad (1)$$

sujeita às restrições definidas pelas equações (2) - (6):

A equação (2) determina o valor de T_{ij} , ou seja, a quantidade de contêineres carregados no Porto de origem i e descarregados no Porto de destino j .

$$\sum_{v=i+1}^j \sum_{r=1}^R \sum_{c=1}^C x_{ijv}(r, c) - \sum_{k=1}^{i-1} \sum_{r=1}^R \sum_{c=1}^C x_{kji}(r, c) = T_{ij}; \quad i = 1, \dots, N - 1; j = i + 1, \dots, N; \quad (2)$$

A equação (3) indica se a posição (r, c) está ocupada ou desocupada em cada segmento de rota, garantindo que não existem rotas com mais de um contêiner nesta posição.

$$\sum_{k=1}^i \sum_{j=i+1}^N \sum_{v=i+1}^j x_{kji}(r, c) = y_i(r, c); \quad i = 1, \dots, N - 1; r = i, \dots, R; c = i, \dots, C; \quad (3)$$

A restrição (4) garante o carregamento dos contêineres em pilhas, uma vez que existindo contêiner na posição $(r + 1, c)$, então também deverá existir um contêiner em (r, c) .

$$y_i(r, c) - y_i(r + 1, c) \geq 0; \quad i = 1, \dots, N - 1; r = i, \dots, R - 1; c = 1, \dots, C; \quad (4)$$

Na equação (5), é definido o movimento de remanejamento. Se um contêiner que ocupa a posição (r, c) é descarregado no porto j , então, ou não existem contêineres acima dele, ou o índice v do contêiner que ocupa o compartimento $(r + 1, c)$ não é maior que j .

$$\sum_{i=1}^{j-1} \sum_{p=j}^N x_{ipj}(r, c) + \sum_{i=1}^{j-1} \sum_{p=j+1}^N \sum_{v=j+1}^p x_{ipv}(r + 1, c) \leq 1; \quad j = 2, \dots, N; r = 1, \dots, R - 1; c = 1, \dots, C; \quad (5)$$

Por fim, a restrição (6) garante a integralidade das variáveis.

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

2.1 Representação por Regras

Para a implementação dos algoritmos deste trabalho, adotou-se a representação das soluções através do emprego de regras, conforme proposto em (Azevedo *et al.*, 2010). Logo, para definir a forma como os contêineres serão colocados a bordo do navio, são definidas regras de carregamento. Em contrapartida, regras de descarregamento de contêineres são empregadas para definir a forma como os contêineres serão retirados do navio. Para tornar viável a representação da localização dos contêineres no navio, será utilizada uma representação matricial que mostra a forma de ocupação do navio, facilitando a visualização da posição e da quantidade de contêineres presentes a bordo. As atividades que modificam esta matriz de ocupação são as operações de carregamento e descarregamento de contêineres, que são realizadas em cada porto. A maneira como são realizadas estas operações, fica a cargo das regras que serão adotadas.

A abordagem através de regras é definida a partir da combinação de uma regra de descarregamento e outra de carregamento para o navio, ou seja, uma combinação k usada em um porto j significa a união destas duas operações de carga e descarga. Em adição ao trabalho de Azevedo e colaboradores (2010), que criou quatro combinações, com base em duas regras de carregamento (C1 e C2) e duas regras de descarregamento (D1 e D2), nesta pesquisa destaca-se como contribuição a criação de outras quatro regras de carregamento (C3, C4, C5 e C6), e uma de descarregamento (D3) que, combinadas com as regras já existentes, totalizam 18 combinações de regras. Na Tabela 1 são apresentadas as definições das combinações empregadas neste trabalho:

Tabela 1 - Definição das Combinações das Regras de Movimentação de Contêineres

Combinação k usada no porto j	Regra de carregamento	Regra de descarregamento
1	C1	D1
2	C1	D2
3	C2	D1
4	C2	D2
5	C3	D1
6	C3	D2
7	C4	D1
8	C4	D2
9	C5	D1
10	C5	D2
11	C6	D1
12	C6	D2
13	C1	D3
14	C2	D3
15	C3	D3
16	C4	D3
17	C5	D3
18	C6	D3

Para representar as etapas de aplicação das regras, é utilizada uma matriz de transporte T (Figura 1), mostrando a quantidade de contêineres que devem ser carregados no porto de origem i (O_i) e com destino em j (D_j). Para a aplicação das regras nos exemplos que seguem, no Porto 1 são empregados somente procedimentos de carregamento, pois o navio inicia a viagem no primeiro porto vazio. Analogamente, no Porto 5 apenas os procedimentos de descarregamento serão aplicados, pois ao final do percurso o navio não efetuará mais nenhum carregamento, devendo finalizar a viagem vazio. Para os demais portos, são aplicadas as combinações de regras de movimentação de contêineres de forma completa, isto é, uma regra de carregamento combinada com uma de descarregamento.

O1	2	5	0	0
O2	0	2	3	1
O3	0	0	2	2
O4	0	0	0	1

D2 D3 D4 D5

Figura 1 - Matriz de Transporte T

Conforme a Figura 1, e assumindo que se esteja no Porto 1, pode-se interpretar que foram carregados dois contêineres com destino no Porto 2 e cinco contêineres com destino no Porto 3 e assim sucessivamente. Ao final, são descarregados, no Porto 5, um contêiner que teve origem no segundo porto, dois contêineres com origem no Porto 3 e um contêiner com origem no Porto 4.

A seguir são descritas as regras empregadas neste trabalho, com a apresentação de um exemplo de sua aplicação.

Regra C1: Nesta regra, a matriz de ocupação é preenchida linha a linha, da esquerda para a direita, destinando a parte inferior da pilha para cargas com destinos mais distantes. Considerando a matriz de transporte da Figura 1, e que se esteja no Porto 1, a matriz de ocupação resultante é representada na Figura 2.

0	0	0	0
0	0	0	0
3	2	2	0
3	3	3	3

Figura 2 - Matriz de Ocupação após Regra C1 (Fonte: (Azevedo *et al.*, 2010))

Regra C2: Em um porto p , esta regra preenche a matriz de ocupação, a cada coluna, da esquerda para a direita, até a linha θ_p . Para determinação da linha θ_p na matriz, o cálculo é realizado após o descarregamento dos contêineres com destino em p , conforme a equação (7) (Azevedo *et al.*, 2010):

$$\theta_p = \left\lceil \frac{\sum_{i=1}^p \sum_{j=p+1}^N T_{ij}}{C} \right\rceil, \quad (7)$$

onde p é o porto atual, T_{ij} é o número de contêineres a serem embarcados no porto i e desembarcados no porto j , e C é o número de colunas da matriz de ocupação no porto p . Na sequência do exemplo anterior, supondo agora que o navio se encontre no Porto 2, aplica-se a Regra C2, carregando-se cada coluna até a altura máxima $\theta_2 = 3$. Logo, a matriz de ocupação resultante, após a aplicação da Regra C2, é apresentada na Figura 3.

0	0	0	0
4	3	3	0
4	3	3	3
5	4	3	3

Figura 3 - Matriz de Ocupação da Regra C2 (Fonte: (Azevedo *et al.*, 2010))

Regra C3: Nesta regra, a matriz de ocupação inicial é completada linha a linha, da direita para a esquerda, destinando a parte inferior da pilha para cargas com destinos mais distantes. Considerando que o navio encontra-se no Porto 2, a matriz resultante é representada na Figura 4.

0	0	0	0
0	3	3	3
3	3	3	3
4	4	4	5

Figura 4 - Matriz de Ocupação da Regra C3

Regra C4: Da mesma forma que as demais regras, a Regra C4 destina a parte inferior do navio para cargas com destinos mais distantes. Os contêineres são carregados coluna a coluna, da direita para a esquerda, com limite até a linha θ_p (equação (7)). A Matriz de ocupação segue conforme Figura 5, admitindo que o navio esteja no Porto 2.

0	0	0	0
0	3	3	4
3	3	3	4
3	3	4	5

Figura 5 - Matriz de Ocupação da Regra C4

Regra C5: Nesta regra os contêineres são carregados da esquerda para a direita, coluna a coluna, sendo limitado até a coluna β_p da matriz. A expressão definida na equação (8) é empregada para a determinação do valor de β_p .

$$\beta_p = \left\lfloor \frac{\sum_{i=1}^p \sum_{j=p+1}^N T_{ij}}{R} \right\rfloor. \quad (8)$$

Através da equação (8), calcula-se o valor de β_p , empregado como limitante das colunas na matriz de ocupação. A Figura 6 apresenta a matriz de ocupação resultante após aplicação desta regra e assumindo-se que o navio esteja no Porto 2. Nesta caso, coluna limite é determinada pelo valor de $\beta_2 = 3$.

4	3	0	0
4	3	3	0
4	3	3	0
5	3	3	0

Figura 6 - Matriz de Ocupação da Regra C5

Regra C6: Na última regra de carregamento, os contêineres são carregados da direita para a esquerda, coluna a coluna, sendo limitado até a coluna β_p da matriz, determinada através da equação (8). Com isso, a matriz de ocupação resultante da aplicação desta regra, assumindo-se o navio no Porto 2, é mostrada na Figura 7.

0	0	3	4
0	3	3	4
0	3	3	4
0	3	3	5

Figura 7 - Matriz de Ocupação da Regra C6

Regra D1: Ao chegar ao porto j (destino), são descarregados todos os contêineres com destino em j , conseqüentemente são descarregados também todos os contêineres bloqueantes, ou seja, aqueles que estão localizados acima dos contêineres que devem ser descarregados. A mesma regra D1 realoca os contêineres bloqueantes, baseando-se na regra de carregamento C1. Considerando que o navio se encontra no Porto 2, a Figura 8 apresenta a matriz de ocupação antes e após a aplicação da Regra D1.

0	0	0	0
5	5	2	2
3	2	3	5
2	3	3	4

0	0	0	0
0	0	0	0
5	3	3	5
5	3	3	4

Figura 8 – Antes e Depois aplicação da regra D1 (Fonte: (Azevedo *et al.*, 2010))

Regra D2: Ao chegar ao Porto j , todos os contêineres são removidos permitindo que todas as pilhas sejam reordenadas. A reordenação dos contêineres com destino nos próximos portos é baseada na Regra C1, mas aplicada na própria regra D2. Assumindo-se que o navio esteja no Porto 2, a Figura 9 apresenta a matriz de ocupação antes e após a aplicação da Regra D2.

0	0	0	0
5	5	2	2
3	2	3	5
2	3	3	4

0	0	0	0
0	0	0	0
3	3	3	3
5	5	5	4

Figura 9 - Antes e Depois aplicação da regra D2 (Fonte: (Azevedo *et al.*, 2010))

Regra D3: Ao chegar ao porto j , são descarregados os contêineres com destino em j , e por consequência, os contêineres que estiverem bloqueando a retirada dos mesmos. Após realizar todo o processo de descarga do navio, os contêineres bloqueantes, que foram retirados (remoções forçadas), são realocados no navio. A realocação é realizada da esquerda para a direita, ocupando cada coluna onde existirem espaços vagos. Assumindo-se que o navio esteja no Porto 2, a Figura 10 apresenta a matriz de ocupação antes e após a aplicação da Regra D3.

0	0	0	0
5	5	2	2
3	2	3	5
2	3	3	4

0	0	0	0
3	0	0	0
5	0	3	5
5	3	3	4

Figura 10 - Antes e Depois aplicação da regra D3

Para a definição das regras, adotou-se o emprego de diferentes estratégias de como ocupar e desocupar o espaço de carga do navio, não sendo empregada qualquer penalização sobre possíveis contêineres bloqueantes ou qualquer favorecimento à ocupação de posições específicas.

Uma vez definidas as seis regras de carregamento e as três regras de descarregamento, foram criadas 18 diferentes combinações, que são efetivamente empregadas na codificação das soluções. A figura 11 apresenta o exemplo de uma solução representada com base nas combinações de regras, indicando que no porto 1 adota-se a combinação 2, no porto 2 a combinação 4 e no porto 3 a combinação 1.

2	4	1
---	---	---

Figura 11 - Solução codificada por regras (Fonte: (Azevedo *et al.*, 2010))

O emprego de regras de carregamento e descarregamento facilita a geração de matrizes de ocupação factíveis, e o seu uso na codificação de soluções permite a representação de forma compacta. A sua adoção permite a redução do espaço de soluções, uma vez que reduz o número de representações das soluções, propiciando a obtenção de soluções promissoras. Apesar da adoção de uma forma compacta de representação da solução, informações relativas aos contêineres a serem carregados e descarregados em cada porto podem ser obtidas a partir da Matriz de Transporte, assim como a disposição dos contêineres no navio em cada porto da rota pode ser obtida pelo acompanhamento da Matriz de Ocupação.

3. Métodos de Solução

Como visto anteriormente, o problema de carregamento de contêineres é um problema de otimização combinatória pertence à classe NP-Completo e de difícil resolução, sendo indicado o emprego de metaheurísticas para a sua solução. Desta forma, neste trabalho serão adotados dois métodos para serem aplicados na solução do problema de carregamento de navios-contêineres. Os métodos aplicados são classificados como Metaheurísticas Populacional e Inspiradas na Natureza (Yang, 2010). Assim, a seguir será apresentado um detalhamento do funcionamento das duas técnicas adotadas: Algoritmo Genético e Algoritmo Enxame de Abelhas.

3.1 Algoritmo Genético (AG)

O Algoritmo Genético é inspirado na teoria da evolução natural proposta por Charles Darwin, tendo sido desenvolvido por John Holland em meados da década de 70. Desde então, diversas variações deste algoritmo foram propostas para solução de problemas de otimização de nível complexo (Michalewicz, 1996; Talbi, 2009). O algoritmo é baseado na evolução das espécies, onde indivíduos se reproduzem herdando características de seus ascendentes e adaptando-se ao ambiente natural. Os indivíduos que apresentarem menor nível de adaptação ao meio ambiente morrerão e a sobrevivência se dará aos mais aptos. Os indivíduos são representados por cromossomo que constituem as possíveis soluções do problema.

Inicialmente, para explorar o espaço de possíveis soluções, é criada uma população de cromossomos, ou soluções. Em seguida, cada indivíduo (cromossomo) da população é avaliado de acordo com a função objetivo. Nesta avaliação, é determinado o valor de aptidão destes indivíduos. Se depois de feita a avaliação (aptidão), o algoritmo não atingir o critério de parada, é realizada a seleção, através de um operador adequado, onde é escolhido um par de cromossomos pais, sendo estes combinados através de uma operação de cruzamento, originando novos cromossomos. Dependendo da taxa de cruzamento adotada, a combinação dos cromossomos pais pode não ser realizada. Em seguida, realiza-se a mutação destes cromossomos, também com base em uma taxa de mutação, em geral pequena. A mutação realiza a alteração dos valores de um ou mais genes, dependendo da estratégia adotada. Após este passo, é realizada a substituição dos indivíduos, quando então é gerada uma nova população através da adoção do elitismo, isto é, o melhor cromossomo da população atual é replicado na próxima população. Com isso, ocorre o encerramento do ciclo, que será repetido até que o critério de parada seja satisfeito ou seja obtido o valor ótimo da função custo, quando conhecido (Talbi, 2009).

Nesta pesquisa, adotou-se o operador de seleção torneio (Mitchell, 1996), sendo realizado entre dois indivíduos da população, escolhidos aleatoriamente, e sendo adotada a probabilidade de 75% para aceitação do melhor indivíduo.

O operador de cruzamento é aplicado após a etapa de seleção, após a escolha de dois cromossomos pais. Neste trabalho, foi adotado o Cruzamento Uniforme (Michalewicz, 1996). O cruzamento uniforme que consiste no sorteio de um cromossomo pai para a determinação de cada

bit do indivíduo filho. Cada bit do indivíduo filho é preenchido com a característica localizada na posição correspondente do indivíduo-pai selecionado. Considerou-se para este cruzamento que os dois indivíduos pais tem a mesma probabilidade de serem escolhidos.

O operador de mutação aplicado foi a Mutação Uniforme, composta pelo sorteio de uma posição do cromossomo e seguida pelo sorteio de uma regra diferente da encontrada na posição sorteada (Mitchell, 1996).

3.2 Algoritmo Enxame de Abelhas (ABC)

O algoritmo empregado neste trabalho é o *Artificial Bee Colony* – ABC que é baseado na coleta de alimento e apropriado para problemas de otimização. O algoritmo *Artificial Bee Colony* (ABC) foi proposto pela primeira vez por Karaboga (2005).

No algoritmo ABC, as abelhas de uma colônia são divididas em abelhas operárias (campeiras), abelhas seguidoras (observadoras) e abelhas escudeiras. As abelhas operárias são designadas para realizarem a procura por fontes de alimento, as abelhas seguidoras escolhem as fontes que irão seguir e as abelhas escudeiras procuram fontes de alimento aleatoriamente.

Inicialmente, as abelhas operárias saem da colmeia em busca de uma fonte de alimento particular. Ao retornarem, compartilham informações, com certa probabilidade, com as abelhas seguidoras. Para a escolha de uma fonte por parte das abelhas seguidoras, a quantidade de néctar de cada fonte é avaliada, ou seja, quanto maior a quantidade mais chances das seguidoras escolherem tal fonte. Esta exploração contínua das fontes de alimento leva ao seu esgotamento. Quando a fonte da abelha operária for esgotada, a abelha operária torna-se uma abelha escudeira e volta a explorar o ambiente em busca de novas fontes de alimento de maneira aleatória.

Relacionando o algoritmo ABC ao problema de otimização em estudo, cada fonte de alimento representa uma possível solução cuja quantidade de néctar representa o valor correspondente da função objetivo. As abelhas operárias executam uma busca local em torno das soluções atuais, através do emprego de estruturas de vizinhança, identificando regiões promissoras, que por sua vez serão selecionadas de acordo com a sua qualidade para serem exploradas pelas abelhas seguidoras, também através de uma busca local. Após todas as abelhas operárias e seguidoras terem completado suas buscas, se uma posição não puder ser melhorada, a fonte de alimento (solução) é abandonada sendo substituída por novas fontes de alimento geradas aleatoriamente e exploradas pelas abelhas escudeiras.

Uma consideração importante para o desenvolvimento do algoritmo é que, inicialmente, na colmeia, 50% das abelhas são operárias e a outra metade são abelhas seguidoras. Para toda abelha operária existe uma fonte de alimento, ou seja, o número de abelhas operárias é o mesmo que o de fontes, uma vez que cada abelha é associada a apenas uma fonte de alimento.

No algoritmo da Colônia Artificial de Abelhas (ABC), é necessária a definição de alguns parâmetros como, o tamanho da população e o número máximo de ciclos, adotado como critério de parada (Karaboga e Akay, 2010). O tamanho da população aplicado neste trabalho foi de 100 fontes de alimentos, e o número máximo de ciclos foi definido em 1000. O nível de aptidão será avaliado conforme o néctar de cada fonte de alimento, determinando a qualidade da fonte de alimento para que as abelhas observadoras efetuem a sua escolha posteriormente.

Para determinação de fontes de alimento vizinhas às soluções atuais, foram aplicadas duas estruturas de vizinhança. Na primeira delas, denominada Troca Aleatória, são selecionadas posições no indivíduo e efetuada uma troca (*swap*) dos valores nas posições selecionadas aleatoriamente. Seguindo a aplicação desta estrutura de vizinhança, avalia-se a probabilidade de aplicação de uma estrutura de vizinhança idêntica ao operador de mutação uniforme empregado no AG.

4. Experimentos e Resultados

Para validação das técnicas de solução, foram utilizados três tipos de matrizes de transporte e serão assumidos os valores de 10 e 30 para a quantidade de portos. Os três tipos de matrizes de transporte são assim definidas (Avriel *et al.*, 1998):

- **Curta Distância (3):** curto período de permanência a bordo.
- **Distância Mista (1):** os elementos são determinados de forma aleatória.
- **Longa Distância (2):** longo período de permanência a bordo

Foram utilizados os problemas-teste disponibilizados em (Azevedo, 2009). Todos os problemas-teste a serem usados na avaliação dos métodos de solução assumem que a capacidade máxima do navio seja de 300 contêineres, onde as dimensões do navio são definidas com $R = 6$ linhas e $C = 50$ colunas (Azevedo *et al.*, 2010). Com base na matriz de transporte empregada em cada problema-teste, é possível determinar o número mínimo de movimentos desejado. A implementação foi desenvolvida na ferramenta Matlab, e os testes foram realizados em um computador com processador Dual Core 2,0GHz, memória RAM de 3GB e Sistema Operacional Windows 7.

Para os experimentos com o AG, foram avaliados diferentes percentuais para a taxa de cruzamento (100%, 90% e 80%) e mutação (5%, 10% e 15%), tendo sido adotado o critério de parada baseado no número máximo de 1000 gerações.

Nos experimentos com o ABC, foi empregada a estrutura de vizinhança por troca e, em 30% dos casos, a mutação uniforme com o objetivo de introduzir diversidade na população, assumindo diferentes quantidades para as fontes de alimento (50 e 100), e tendo sido adotado o critério de parada de 1000 ciclos.

Os experimentos de validação foram realizados utilizando-se 4 combinações de regras, e empregando ambas as estratégias de solução propostas. Os resultados obtidos foram comparados com os resultados encontrados por Azevedo *et al.* (2010), quando empregando o Algoritmo Genético, também com apenas 4 combinações de regras. A Tabela 2 apresenta os resultados obtidos.

Tabela 2 – Resultados dos Experimentos de Validação (4 combinações de regras)

Inst.	AG 4R	ABC 4R	AG 4R*	ÓTIMO
	FO (σ)	FO (σ)	FO	FO
10/1	1438,00 (0,00)	1462,00 (11,95)	1438,00	1322
10/2	1236,00 (0,00)	1256,20 (8,13)	1236,00	750
10/3	2288,00 (0,00)	2288,00 (0,00)	2288,00	2282

*(Azevedo *et al.*, 2010)

Na Tabela 2, são apresentados valores médios da Função Objetivo após 30 execuções do algoritmo. Pode ser observado que, tanto o AG quanto o ABC desenvolvidos, forneceram resultados semelhantes àqueles encontrados na literatura, demonstrando a validação das implementações realizadas. Além disso, o AG mostrou-se mais eficiente que o ABC, em virtude dos melhores valores médios obtidos, mas também pelo valor de desvio padrão menor. Através destes experimentos de validação, foi possível identificar a melhor configuração de operadores e parâmetros para o AG, tendo sido empregados o cruzamento uniforme e a mutação uniforme com taxas de 90% e 15%. Para o ABC, foram obtidos bons resultados quando utilizadas 100 fontes de alimento, vizinhança por troca e 30% para a taxa de emprego da mutação uniforme. Com relação ao tempo computacional despendido nos experimentos de validação, o AG consumiu 1,5 min. em média, enquanto o ABC exigiu 60 min. em média.

Uma vez definidas as melhores configurações de cada algoritmo, foram realizados experimentos a fim de avaliar a contribuição do emprego de um número maior de combinações de regras. A Tabela 3 apresenta os resultados do AG e do ABC, quando utilizadas 18 combinações de regras, e aplicados ao seis conjuntos de problema teste, diferindo entre si pela quantidade de portos presentes na rota (10 ou 30) e o tipo de matriz de transporte empregada (tipo 1, 2 ou 3). Os resultados médios, obtidos de um conjunto de 30 execuções do algoritmo, foram

comparados com o valor ótimo conhecido, conforme apresentado na Tabela 3.

Tabela 3 – Resultados dos Experimentos de Teste (18 combinações de regras)

Inst.	AG 18R	ABC 18R	ÓTIMO	GAP (%)	
	FO (σ)	FO (σ)	FO	AG 18R	ABC 18R
10/1	1335,03 (2,09)	1346,77 (3,73)	1322	0,98	1,87
10/2	760,00 (0,00)	772,55 (5,91)	750	1,33	3,01
10/3	2282,00 (0,00)	2282,00 (0,00)	2282	0,00	0,00
30/1	2311,43 (14,2)	2481,30 (48,97)	2262	2,18	9,69
30/2	1094,5 (13,92)	1248,77 (76,61)	1030	6,26	21,23
30/3	6380,00 (0,00)	6380,00 (0,00)	6380	0,00	0,00

Através de uma análise dos resultados, percebe-se que os valores médios encontrados com 18 combinações de regras (Tabela 3) mostram-se melhores que os valores médios encontrados nos testes com 4 combinações de regras (Tabela 2), tanto para o AG quanto para o ABC. Logo, o emprego das novas e mais combinações de regras produziu melhores resultados, mais próximos da solução ótima.

Comparando os resultados médios obtidos com o AG e o ABC, quando empregadas 18 combinações de regras (Tabela 3), percebe-se que o AG novamente mostrou-se mais eficiente que o ABC, com médias e desvio-padrão menores. Com o objetivo de se ter uma medida da qualidade dos resultados obtidos, utilizou-se a grandeza denominada GAP, definida como a diferença relativa percentual de cada solução média em relação à solução ótima. Desta forma, quanto mais próximo de zero for este valor, mais próximo o resultado obtido estará do resultado ótimo. Desta forma, o AG mostra-se mais próximo da solução ótima se comparado ao ABC, com exceção dos casos onde a matriz de transporte é do tipo 3 (10/3 e 30/3), onde as soluções encontradas através do emprego de ambos os métodos são idênticas. Tal exceção explica-se pela característica da matriz do tipo 3, que corresponde à matriz de transporte de curta distância, onde os contêineres embarcados permanecem um curto período a bordo, o que facilita o gerenciamento das operações de carga e descarga realizadas nos portos ao longo da rota. Com relação ao tempo computacional despendido nos experimentos empregando-se 18 combinações de regras e problemas-teste de diferentes dimensões, o AG exigiu, novamente, 1,5 min. em média para o caso com 10 portos e 6 min. em média para o caso com 30 portos. Já o ABC consumiu aproximadamente 60 min. para o caso com 10 portos e 230 min. para o caso com 30 portos. Problemas-teste assumindo navios de maiores dimensões (1000 e 5000 contêineres) foram também avaliados, tendo sido obtidos resultados que forneceram um valor de GAP inferior a 3,70% para o caso de 10 portos.

5. Conclusão

O problema de carregamento de navios-contêineres é de fato importante e merece atenção. Este problema influencia na eficiência de um porto, que é um fator essencial para a melhora de sua atratividade e operacionalidade. Neste sentido, este trabalho buscou ampliar a codificação por regras proposta no trabalho de Azevedo *et al.* (2010) para a minimização do remanejamento de contêineres, em operações de carga e descarga em terminais portuários. Para isso, foram criadas novas regras de carregamento e descarregamento de contêineres, que foram aplicadas às Metaheurísticas Algoritmo Genético (AG) e Colônia de Abelhas Artificiais (ABC). A criação de novas regras e o emprego da metaheurística Colônia de Abelhas Artificiais são destacados como as principais contribuições deste trabalho. Além disso, o trabalho também apresentou uma aplicação do Algoritmo Genético baseado na codificação da solução por regras. As simulações das metaheurísticas propostas, utilizando 4 combinações de regras, forneceram resultados semelhantes, idênticos e ligeiramente melhores que os relatados na literatura (Azevedo

et al., 2010), tendo sido observado, especificamente para o caso do AG, que a aplicação de operadores diferentes melhorou os resultados daquele estudo. Comparando os resultados obtidos através das metaheurísticas AG e ABC, empregando-se 4 combinações de regras, observa-se uma supremacia do AG. Com o emprego de 18 combinações de regras, os resultados obtidos continuaram demonstrando a vantagem do AG sobre o ABC, porém o uso de uma quantidade maior de regras possibilitou o aprimoramento acentuado das soluções, tendo sido obtidos resultados mais próximos dos ótimos globais conhecidos. Ademais, o AG exigiu um esforço computacional notavelmente menor que o ABC. Frente aos resultados obtidos, destaca-se que as novas regras possibilitaram uma melhora significativa nos resultados, e que o Enxame de Abelhas mostrou-se uma técnica viável de ser aplicada na solução do problema de carregamento de navios-contêineres, o que ainda não havia sido encontrado na literatura. apesar do esforço computacional exigido.

Agradecimentos

Os autores gostariam de agradecer à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo suporte financeiro dado para a realização desta pesquisa.

Referências

- Avriel, M., Penn, M., Shpirer, N. e Witteboon, S.** (1998), Stowage Planning for Container Ships to Reduce the Number of Shifts, *Annals of Operations Research*, 76, 55–71.
- Avriel, M., Penn, M., Shpirer, N.** (2000), Container ship stowage problem: complexity and connection to the coloring of circle graphs, *Discrete Applied Mathematics*, 103, 271–279.
- Azevedo, A. T.**, Problemas & Dados, *Instâncias empregadas nos artigos do SIMPEP 2009*, Universidade Estadual Paulista, 2009 (<https://sites.google.com/site/projetonavio/programas-dados>), 8, 2012.
- Azevedo, A. T., Ribeiro, C. M. e Deus, N. M. R.** (2010), Resolução do Problema de Carregamento e Descarregamento de Contêineres em Terminais Portuários via Algoritmo Genético. *Revista Ingerpro – Inovação, Gestão e Produção*, 2, 38-51.
- Botter, R. C. e Brinati, M. A.** (1992), Stowage Container Planning: A Model for Getting an Optimal Solution, *Proceedings of the IFIP TC5/WG5.6 Seventh International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design*, 217–229.
- Dubrovsky, O; Levitin, G. e Penn, M.** (2002), A Genetic Algorithm with a compact solution encoding for the container ship stowage Problem, *Journal of Heuristics*, 8, 585-599.
- Karaboga, D.** An idea based on honey bee swarm for numerical optimization. *Technical Report TR06*, Erciyes University, 10, 2005 (http://mf.erciyes.edu.tr/abc/pub/tr06_2005.pdf).
- Karaboga, D. e Akay, B.** (2009) Artificial Bee Colony (ABC), Harmony Search and Bees Algorithms on Numerical Optimization, *Proceedings of the Innovative Production Machines and Systems Virtual Conference*.
- Michalewicz, Z.**, *Genetic Algorithms + data structures = evolution programs*, Springer-Verlag, Berlin, 1996.
- Talbi, E-G.**, *Metaheuristics: from design to implementation*, John Wiley & Sons, New Jersey, 2009.
- Mitchell, M.**, *An introduction to genetic algorithms*, MIT Press, Cambridge, 1996.
- Sciomachen, A. e Tanfani, E.** (2007) A 3D-BPP approach for optimizing stowage plans and terminal productivity, *European Journal of Operational Research*, 183, 1433-1446.
- Wilson, I. D. e Roach, P.A.** (1999) Principles of Combinatorial Optimization Applied to Container-Ship Stowage Planning, *Journal of Heuristics*, 5, 403-418.
- Yang, X-S.**, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, Frome, 2010.