



Metaheurística híbrida aplicada ao problema de alocação tática de berços

Mateus Matos Rizzi

Universidade Federal do ABC – UFABC
m.rizzi@ufabc.edu.br

Carlos Zaca Pomari

Universidade Federal de São Paulo - UNIFESP
cpomari@sabesp.com.br

Rudinei Martins de Oliveira

Universidade Federal de São Paulo - UNIFESP
São José dos Campos, SP
rudmart@gmail.com

Antonio Augusto Chaves

Universidade Federal de São Paulo – UNIFESP
antonio.chaves@unifesp.br

Luiz Antonio Nogueira Lorena

Instituto Nacional de Pesquisas Espaciais - INPE
São José dos Campos, SP
luiz.lorena@inpe.br

RESUMO

O objetivo deste trabalho é aplicar o método híbrido *Clustering Search* (CS) em conjunto com o *Biased Random Key Genetic Algorithm* (BRKGA) ao problema de alocação tática de berços (TBAP). O TBAP consiste em alocar navios nos berços e definir o perfil de guindastes que serão usados para descarregamento e carregamento das cargas, buscando minimizar o custo de serviço em transferir as cargas entre os navios atracados e otimizar a utilização dos guindastes disponíveis para as tarefas. Para a validação dos resultados, foram utilizados três conjuntos de instâncias e as soluções foram comparadas com outros métodos disponíveis na literatura.

PALAVRAS CHAVE. Metaheurística híbrida, Busca local, Sequenciamento.

Área Principal: MH - Metaheurísticas

ABSTRACT

This paper aims to apply the hybrid method *Clustering Search* (CS) with the *Biased Random Key Genetic Algorithm* (BRKGA) on the *Tactical Berth Allocation Problem* (TPAB). The TPAB consists of allocate incoming ships to berthing positions and assign quay crane profile to them. The goals of the TBAP are both the minimization of the housekeeping costs derived from the transshipment container flows between ships, and the maximization of the total value of the quay crane profiles assigned to the ships. To validate the computational results, it was used three sets of instances available on literature and the proposed method is compared with other solution approaches.

KEYWORDS. Hybrid metaheuristic, Local search, Scheduling.

Main Area: MH - Metaheuristics



1. Introdução

O transporte marítimo de mercadorias cresceu muito nos últimos anos em decorrência do aumento do comércio e do crescimento econômico internacional [Hansen et al., 2008]. O comércio internacional tem como pilar central o transporte marítimo. Aproximadamente 80% do comércio global de mercadorias é realizado através do mar [Buhkral et al., 2009]. No Brasil, no ano de 2010 a movimentação de cargas no sistema portuário nacional alcançou 834 milhões de toneladas, um aumento de 13,8% em relação à 2009 [ABT, 2011]. Além disso, em pesquisa realizada pela Secretaria de Comércio Exterior (SECEX), o transporte marítimo apresentou crescimento de 130% nos últimos trinta anos. Isto se caracteriza pela relação entre a natureza (mar/oceano) e a melhorias na qualidade e rapidez no atendimento dos navios nos portos, visto que ambos são imprescindíveis para o escoamento de mercadorias.

Para um sistema de transporte eficiente, é necessário otimizar os recursos para o escoamento e recebimento de mercadoria, como minimizar o tempo em que os berços ficam ociosos e otimizar a utilização dos guindastes. O funcionamento de carregamento e descarregamento nos terminais marítimos consistem em uma série de etapas como indicadas em Steenken et al. [2004] e Stahlbock e Voß [2008]. Quando um navio atraca em um porto, ele é alocado para um berço específico, e em seguida, alguns guindastes são direcionados para atendê-lo. Após o descarregamento a sua carga é armazenada em um depósito e em seguida o navio é carregado com os contêineres que seguirão aos seus próximos destinos. As operações de carregamento e descarregamento são abordadas no problema de alocação tática de berços (TBAP, do inglês *Tactical Berth Allocation Problem*), tal que, os navios são atracados em berços e o tempo de permanência é definido pela quantidade de guindastes atribuídos a ele para realizar o carregamento e descarregamento. A quantidade de guindastes disponíveis é limitada, os berços possuem tempo de serviço e os navios têm horários de chegadas e partidas previstos.

Neste contexto, este trabalho propõe uma nova alternativa para resolver o TBAP por meio da aplicação do método híbrido *Clustering Search* - CS [Oliveira e Lorena, 2013] e da metaheurística *Biased Random Key Genetic Algorithm* - BRKGA [Gonçalves e Resende, 2011]. O CS é um método híbrido que combina metaheurísticas e heurísticas de busca local, em que a busca é intensificada somente em regiões do espaço de soluções que mereçam atenção especial (regiões promissoras). O BRKGA será utilizado como gerador de soluções para o CS. O método proposto CS+BRKGA é comparado a métodos recentes encontrados na literatura, permitindo assim verificar sua eficiência na resolução do TBAP. O restante deste trabalho está organizado da seguinte forma. A seção 2 apresenta uma revisão da literatura sobre o problema tratado. A seção 3 descreve o problema TBAP. A seção 4 explica o funcionamento da metaheurística híbrida CS+BRKGA. A seção 5 apresenta os resultados computacionais e a seção 6 descreve as conclusões finais.

2. Revisão da literatura

Devido ao crescimento econômico mundial nos últimos anos, o comércio marítimo tem motivado o aumento nos estudos sobre o funcionamento dos portos, focando a atenção em problemas nas áreas estratégica, tática e operacional. A maioria dos problemas enfrentados nessas situações são do tipo NP-Hard, tais como, o Problema de alocação de berços (BAP) e o problema de alocação de guindastes (QCAP). O BAP consiste em alocar navios a posições de atracação, de forma que seja utilizado o máximo de espaço do cais minimizando o tempo de serviço. As decisões a serem tomadas dizem respeito a posição e ao instante em que o navio deverá atracar [Imai et al., 2001]. O BAP possui grande quantidade de restrições físicas, técnicas, entre outras. Isto faz com que seja possível modelá-lo de diferentes maneiras. Quanto aos aspectos espaciais dos berços, o BAP pode ser modelado como discreto ou contínuo [Imai et al., 2005]. Além disso, se for levado em conta a chegada dos navios, o problema pode ser tratado como estático ou dinâmico [Imai et al., 2001]. O caso estático assume que todos os navios já estão no porto para o atendimento, o caso dinâmico permite aos navios chegarem a qualquer momento.



Inúmeras abordagens já foram propostas para a solução do BAP, Cordeau e Laporte [2005] propôs minimizar o tempo de serviço total para todos os navios, definindo o tempo decorrido entre a chegada no porto, atracação e o tempo de serviço, formulando o problema como análogo ao problema de roteamento de veículos com múltiplas garagem e janelas de tempo. [Mauri et al., 2008a] propõem uma abordagem baseada na aplicação do *Simulated Annealing* para resolução do caso discreto do BAP. Os autores tratam o problema como um problema de roteamento de veículos com múltiplas garagens e janelas de tempo. Os resultados computacionais superam os obtidos pelo CPLEX e pela *Busca Tabu* proposta por [Cordeau e Laporte, 2005].

O QCAP recebeu pouca atenção dos pesquisadores até então, ganhando mais visibilidade quando foi proposto juntamente com BAP, criando assim o problema tático de alocação de berços (TBAP). Essa junção foi proposta por [Imai et al., 2008], tal que, na função objetivo busca-se a minimização da janela de tempo para o funcionamento de todos os navios, onde um navio só poderia ser atendido, se uma quantidade mínima de guindastes estivesse disponível. Essa solução não levava em conta que a operação do navio poderia acontecer com diferentes números de guindastes, fazendo com que navios esperassem para poder atracar no berço. A proposta que integrava um perfil de guindastes junto com o objetivo de minimizar a janela de tempo veio com Liang et al. [2009]. Para resolver o problema os autores utilizam um algoritmo genético e usaram dados reais de uma das empresas de contêineres de Xangai na China. Anos após, [Raa et al., 2011] apresentou um modelo que integrava o BAP ao QCAP e considerava prioridade de navios, localização de berços e tempo de operação. Este trabalho se baseou na solução proposta de [Lalla-Ruiz et al., 2014]. Neste artigo, os autores propuseram o "*Biased Random Key Genetic Algorithm*" para resolver o TBAP, levando em conta não somente a alocação dos berços, mas também a atribuição dos guindastes.

3. Alocação tática de berços

O problema de alocação tática de berços (TBAP) busca determinar o berço, o tempo de operação e a alocação de guindastes através de perfis, para cada navio que chega ao porto com horário previsto [Giallombardo et al., 2010]. O tempo de operação de cada navio nos berços é definido através do perfil utilizado para o atendimento, cada perfil define a quantidade de guindaste associada a cada 2 horas de operação. Quanto mais guindastes por 2 horas se utiliza, maior o custo da operação, e com isso, mais rápido o navio é atendido. Cada navio tem um conjunto de perfis associados a ele. Todos os perfis têm um custo associado ao número de guindastes utilizado e navio atendido. Portanto, um mesmo perfil pode ter custo diferente devido á estar relacionado à diferentes navios. Quando um navio chega no porto para ser atendido, alguns dos contêineres que ele vai carregar podem ainda não ter chegado ou estar em algum outro berço e isto deve ser levado em conta. Cada navio traz a informação do fluxo de carga que o outro navio irá transportar. Assim, o custo de armazenamento é proporcional à carga armazenada, e dependendo de qual navio receberá a carga, um custo relacionado ao transporte entre navios é contabilizado. Tal que:

- N : Conjunto de navios, $n = |N|$;
- M : Conjunto de berços, $m = |M|$;
- H^s : Total de horas para o atendimento;
- P_i^s : Conjunto de perfis viáveis atribuído para um navio $i \in N$ quando o navio chega a um passo de tempo com índice $s \in S$ dentro de um turno de trabalho;
- P_i : Quantidade de perfis de guindastes para cada navio $i \in N$;
- t_i^p : Tempo de serviço do navio $i \in N$ para cada perfil de guindastes $p \in P_i$;
- v_i^p : Valor do serviço prestado ao navio $i \in N$ para cada perfil de guindastes $p \in P_i$;



- q_i^{pu} : Número de guindastes atribuídos ao navio $i \in N$ sob o perfil $p \in P_i$ no passo de tempo $u \in (1, \dots, t_i^p)$, tal que $u = 1$ corresponde ao tempo de chegada do navio;
- Q^h : Quantidade de guindastes disponíveis em um passo de tempo $h \in H$;
- f_{ij} : Fluxo de contêineres trocados entre os navios $i, j \in N$;
- d_{kw} : Custo do deposito de contêiner por unidade nos berços $k, w \in M$;
- a_i, b_j : Tempo de Chegada e partida dos navios $i \in N$;
- a^k, b^k : Abertura e fechamento do funcionamento dos berços $k \in M$;
- $y_i^k \in \{0, 1\}, \forall k \in M, \forall i \in N$, 1 se o navio estiver atracado no berço k e 0 caso contrário;
- $L_i^p \in \{0, 1\}, \forall p \in P_i, \forall i \in N$, 1 se o navio estiver funcionando com o perfil p e 0 caso contrário.

As variáveis de decisão são definidas como:

- $x_{ij}^k \in \{0, 1\} \forall k \in M, \forall (i, j) \in A^k$, x_{ij}^k é igual a 1 se o navio j é atribuído ao berço k após o navio i , e 0 caso contrário;
- $y_i^k \in \{0, 1\} \forall k \in M, \forall i \in N$, se o navio i é atribuído ao berço k , e 0 caso contrário;
- $\lambda_i^h \in \{0, 1\} \forall h \in H, \forall i \in N$, λ_i^h é igual a 1 se o navio i é chega ao berço em um tempo h , e 0 caso contrário;
- $\lambda_i^p \in \{0, 1\} \forall p \in P_i, \forall i \in N$, λ_i^p é igual a 1 se o navio i for servido pelo perfil p e 0 caso contrário;
- $\rho_i^{ph} \in \{0, 1\} \forall p \in P_i, \forall h \in H, \forall i \in N$, ρ_i^{ph} é igual a 1 se o navio i for servido pelo perfil p e chega no passo h , e 0 caso contrário;
- $T_i^k \geq 0 \forall k \in M, \forall i \in N$, é o tempo de atracação de navio i no berço k ;
- $T_{o(k)}^k \geq 0 \forall k \in M$, é o tempo que o primeiro navio atraca no berço k ;
- $T_{d(k)}^k \geq 0 \forall k \in M$, é o tempo de operação final do berço k , quando o último navio deixa o berço.

O modelo matemático para resolver o TBAP proposto por Giallombardo et al. [2010] é apresentado em seguida:

$$\max \sum_{i \in N} \sum_{p \in P_i} \nu_i^p \lambda_i^p - \frac{1}{2} \sum_{i \in N} \sum_{k \in M} y_i^k \sum_{j \in N} \sum_{w \in M} f_{ij} d_{kw} y_j^w \quad (1)$$

Sujeito à:

$$\sum_{k \in M} y_i^k, \forall i \in N, \quad (2)$$

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

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



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

$$\sum_{j \in N \cup \{d(k)\}} X_{ij}^k = \sum_{k \in M} y_i^k, \forall k \in M, \forall i \in N, \quad (6)$$

$$T_i^k + \sum_{p \in P_i} t_i^p \lambda_i^p - T_j^k \leq (1 - X_{ij}^k) M1, \forall k \in M, \forall i \in N, j \in N \cup \{d(k)\}, \quad (7)$$

$$T_{o(k)}^k - T_j^k \leq (1 - X_{o(k),j}) M2, \forall k \in M, \forall j \in N, \quad (8)$$

$$a_i y_i^k \leq T_i^k, \forall k \in M, \forall i \in N, \quad (9)$$

$$T_i^k \leq b_i y_i^k, \forall k \in M, \forall i \in N, \quad (10)$$

$$a^k \leq T_{o(k)}^k, \forall k \in M, \quad (11)$$

$$T_{d(k)}^k \leq b_k, \forall k \in M, \quad (12)$$

$$\sum_{p \in P_i} \lambda_i^p = 1, \forall i \in N, \quad (13)$$

$$\sum_{h \in H^s} \gamma_i^h = \sum_{\lambda_i^p} \gamma_i^p, \forall i \in N, \forall s \in S, \quad (14)$$

$$\sum_{k \in M} T_i^k - b^h \leq (1 - \gamma_i^h) M3, \forall h \in H, \forall i \in N, \quad (15)$$

$$a^h - \sum_{k \in M} T_i^k \leq (1 - \gamma_i^h) M4, \forall h \in H, \forall i \in N, \quad (16)$$

$$\rho_i^{ph} \geq \lambda_i^p - b^h + \gamma_i^h - 1, \forall h \in H, \forall i \in N, \forall p \in P_i, \quad (17)$$

$$\sum_{i \in N} \sum_{p \in P_i} \sum_{u=\max\{h-t_i^p+1;1\}}^h \rho_i^{pu} q_i^{p(h-u+1)} \geq Q^h, \forall h \in H^s, \quad (18)$$

$$X_{ij}^k \in \{0, 1\}, \forall k \in M, \forall (i, j) \in A^k, \quad (19)$$

$$y_i^k \in \{0, 1\}, \forall k \in M, \forall i \in N, \quad (20)$$

$$\gamma_i^h \in \{0, 1\}, \forall h \in H, \forall i \in N, \quad (21)$$

$$\lambda_i^k \in \{0, 1\}, \forall p \in P_i, \forall i \in N, \quad (22)$$

$$\rho_i^{ph} \in \{0, 1\}, \forall p \in P_i, \forall h \in H, \forall i \in N, \quad (23)$$



$$T_i^k \geq 0 \forall k \in M, \forall i \in N \cup \{o(k), d(k)\}, \quad (24)$$

Tal que M1, M2, M3 e M4 são constantes suficientemente grandes. A função objetivo (1) maximiza a soma dos custos dos perfis de guindastes escolhidos ν_i^p associado pelos navios $i \in N$, ao mesmo tempo que, visa minimizar a somatória do custo de transporte de carga entre navios e armazenamento de contêiner nos depósitos de cada berço. As restrições (2) afirmam que cada navio i deve ser atribuído para um e somente um berço k . As restrições (3) e (4) definem os fluxos de saída e entrada para os berços. A conservação de fluxo para os vértices restantes é assegurada pelas restrições (5). As restrições (6) indicam a ligação entre variáveis X_{ij}^k e y_i^k . As precedências em cada sequência são asseguradas pelas restrições (7) e (8), que definem coerentemente as variáveis de tempo T_i^k . As janelas de tempo na hora de chegada são indicadas para cada navio pelas restrições (9) e (10), enquanto as janelas de tempo nas disponibilidades dos berços são indicadas pelas restrições (11) e (12). As restrições (13) garantem que um e somente um perfil de guindaste é atribuído a cada navio. As restrições (14) definem a ligação entre as variáveis γ_i^h e λ_i^p , enquanto as restrições (15) e (16) ligam as variáveis binárias γ_i^h ao tempo de chegada T_i^k . Observe que as restrições (10) implicam que $T_i^k = 0$ quando o navio não atracar no berço $K \in K$. As variáveis ρ_i^{ph} são ligadas às variáveis γ_i^h e λ_i^p pelas restrições (17): em particular, ρ_i^{ph} é igual a 1 se e somente se $\lambda_i^p = \gamma_i^h = 1$. Finalmente, as restrições (18) asseguram que, a cada passo de tempo, o número total de guindastes atribuídos não exceda o número de guindastes que estão disponíveis no terminal.

4. Biased Random Key Genetic Algorithm (BRKGA) com o *Clustering Search* (CS)

Nesta seção serão apresentados a metaheurística BRKGA, o método híbrido CS, e a abordagem combinando estes métodos para resolver o TBAP.

4.1. Biased Random Key Genetic Algorithm (BRKGA)

O *Biased Random Key Genetic Algorithm* (BRKGA) foi proposto por [Gonçalves e Resende, 2011]. É uma variação do Random Key Genetic Algorithms (RKGA) [Bean, 1994] que é baseado no algoritmo genético [Holland, 1975]. O BRKGA representa uma solução como um vetor de chaves aleatórias. Cada posição do vetor é composta por números reais no intervalo $[0, 1]$. Este vetor (também chamado de cromossomo) não é considerado uma solução do problema. Então, é necessário decodificá-lo na solução do problema que está sendo resolvido. Para cada problema é definido um decodificador (*decoder*) específico. O *decoder* é um algoritmo determinístico que pega as informações do cromossomo e retorna uma solução viável do problema. A aptidão (função objetivo) de cada solução também é calculada pelo *decoder*. O processo de evolução do BRKGA é independente do problema. A população de P vetores de chave aleatórias evolui ao longo das gerações. Em cada geração, a população é ordenada pelo valor da função objetivo. Em seguida, um pequeno grupo com as melhores soluções (p_e) (grupo elite) são copiados sem modificação para a população da próxima geração. Um número (p_m) de vetores de chaves aleatórias (mutantes, gerados aleatoriamente) também são introduzidos nesta população. O restante da população ($P - p_e - p_m$ soluções) é produzido através do cruzamento, combinando os pais dos vetores elites com os não elites na solução corrente. A Figura 2 apresenta o fluxograma do BRKGA.

O BRKGA foi adaptado para resolver o TBAP. O método cria aleatoriamente uma população inicial de vetores de chaves aleatórias. Para a decodificação em chaves aleatórias do problema foi utilizado um vetor de tamanho $2 * N$ que recebe valores randômicos entre $[0, 1]$, sendo as posições de 0 a $N - 1$ representando os navios, e as posições de N a $2N - 1$, representando o perfil utilizado pelos navios. A primeira parte da subsequência é ordenada de forma crescente, definindo a sequência de navios que terá a prioridade no atendimento. A segunda subsequência divide o intervalo $[0, 1]$ em subintervalos com o número de possibilidade de perfis, sendo cada intervalo representando um perfil de guindaste.

O decodificador associa os navios aos berços, ou seja, os navios são alocados em um berço até que ele não suporte mais, e para se determinar a ordem de berços que serão preenchidos,



é utilizado as funções 25 e 26. A função 25 determina o berço mais barato em termos de custo de armazenamento de carga, para se determinar o primeiro berço a ser preenchido. Os parâmetros α_1 e α_2 são usados para assegurar que o berço escolhido seja o mais barato em relação ao custo de armazenamento. Determinado o berço mais barato, os navios são alocados até que não haja mais espaço. Para a determinação do próximo berço, se utiliza a função 26, que considera o valor de custo de armazenamento e o custo de trocas de cargas entres os navios alocados e os navios que ainda serão alocados para a determinação do berço mais barato.

$$C(j) = \alpha_1 d_{jj} + \alpha_2 \frac{\sum_{k \in M \setminus \{j\}} d_{jk}}{M-1}, \forall j \in M, \quad (25)$$

$$C(i, j) = \sum_{k \in N} \sum_{w \in M} f_{ik} d_{jw} \gamma_{kw} \quad (26)$$

4.2. Clustering Search (CS)

O método híbrido Clustering Search (CS) [Oliveira e Lorena, 2013], se baseia no conceito de regiões promissoras e aplica heurísticas de busca local nestas regiões para intensificar o processo de convergência para boas soluções. O CS é composto por três componentes principais: uma metaheurística geradora de soluções, um processo de agrupamento e uma heurística de busca local. A cada iteração do CS, uma solução S é gerada pela meta-heurística e enviada para o processo de agrupamento. Essa solução é então agrupada no *cluster* mais similar C_j e o centro desse *cluster* c_i é atualizado com informações contidas na nova solução agrupada, fazendo com que o centro se desloque no espaço de busca. Em seguida, é analisado o volume v_i do *cluster* e, caso esse volume atinja um limitante λ ($v_i \geq \lambda$), percebe-se que algum padrão de solução está sendo predominantemente gerado pela metaheurística. Portanto, esse *cluster* pode estar em uma região de busca promissora.

Por fim, é analisado o índice de ineficácia r_j , ou seja, caso a heurística de busca local não melhore a solução por r_{max} vezes consecutivas ($r_j \geq r_{max}$), é aplicada uma perturbação aleatória no centro c_j , objetivando escapar de um possível ótimo local. Por outro lado, se $r_j < r_{max}$, a heurística de busca local é aplicada no centro c_j analisando a vizinhança do *cluster*. Encerrado esse processo, retorna-se para a metaheurística que irá gerar uma nova solução. O critério de parada do CS é geralmente definido pela metaheurística escolhida. A Figura 1 apresenta o fluxograma de execução do CS.

4.3. CS+BRKGA

Seguindo o fluxograma do CS (Figura 2), são criados então os *clusters* iniciais. Assim, para cada *cluster* é criada uma solução gerada aleatoriamente. As soluções são formadas por chaves aleatórias. A partir de então, o BRKGA é executado e apenas as soluções filhas são enviadas para o processo de agrupamento do CS. Para que as soluções possam ser agrupadas em *clusters* similares, deve-se definir uma métrica para medir a distância entre duas soluções. Para isso, utiliza-se neste trabalho a distância Euclidiana. O processo de agrupamento de cada filho ao *cluster* mais similar é de acordo com a menor distância calculada. A atualização do centro do *cluster* é dada pela execução do *Path-Relinking* (PR) entre a solução dada S e a solução c_j centro do *cluster* C_j . A atualização do centro do *cluster* é feita com a combinação entre suas chaves aleatórias. O algoritmo PR é usado para intensificar e diversificar a busca em um grupo de soluções. Ele realiza movimentos exploratórios em uma vizinhança a partir de uma solução inicial buscando atingir uma solução guia. O objetivo é encontrar as melhores soluções no caminho que conectam estas soluções. Os movimentos gradualmente introduzem informações dos atributos da solução guia para a solução inicial [Resende e Ribeiro, 2005].

Quando um *cluster* atinge um número de soluções superior ao parâmetro λ , o centro se torna promissor, sendo realizada uma busca local nesta solução. A busca local é feita usando três heurísticas distintas, *Swap*, *Shift* e *Swap* perfil:

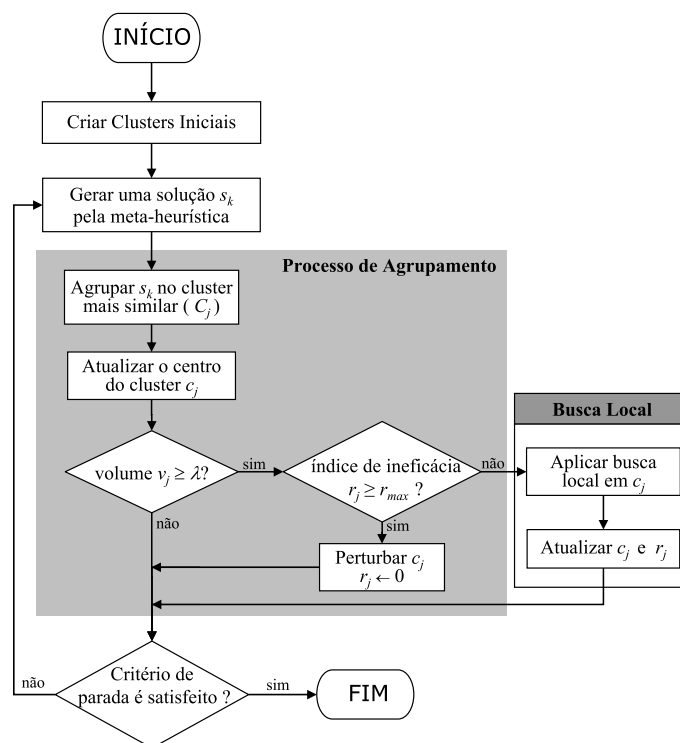


Figura 1: Fluxograma do CS [Oliveira e Lorena, 2013].

- Swap navio/berço: realiza todas as trocas possíveis entre navios de berços diferentes;
- Shift navios: realiza todas as trocas de dois navios dentro do mesmo berço;
- Swap perfil: analisam-se todos os perfis não utilizados pelos navios alocados, verificando se a troca de perfil utilizado melhora a solução sem comprometer sua viabilidade.

Outra busca local é examinada apenas para soluções que possuam inviabilidades. A troca da sequência de navios no berço não altera o valor da solução, mas pode tornar uma solução viável em inviável e vice-versa. Essa busca local é aplicada apenas em soluções inviáveis, buscando diminuir o número de penalizações ou que as soluções se tornem viáveis.

5. Resultados computacionais

O método heurístico foi codificado na linguagem C++ e os testes computacionais foram executados em um PC com processador Intel core i5, 2.5 GHz e memória de 6 GB. Para verificar a eficiência do método proposto foram utilizadas as instâncias de Giallombardo et al. [2010], Vacca et al. [2013] e Lalla-Ruiz et al. [2014]. Os resultados obtidos foram comparados com os resultados do BRKGA proposto por Lalla-Ruiz et al. [2014] e também com os “upper bound” determinados pelo CPLEX, em Lalla-Ruiz et al. [2014]. Os parâmetros do CS+BRKGA foram calibrados através de testes empíricos realizados sobre um subconjunto de instâncias. Os parâmetros adotados neste trabalho são:

- Número de *clusters* do CS (NC) = 20.
- Limitante de *cluster* promissor (λ) = 10.
- Intensidade da perturbação de um *cluster* (β) = 4.
- População(p) = 1000.

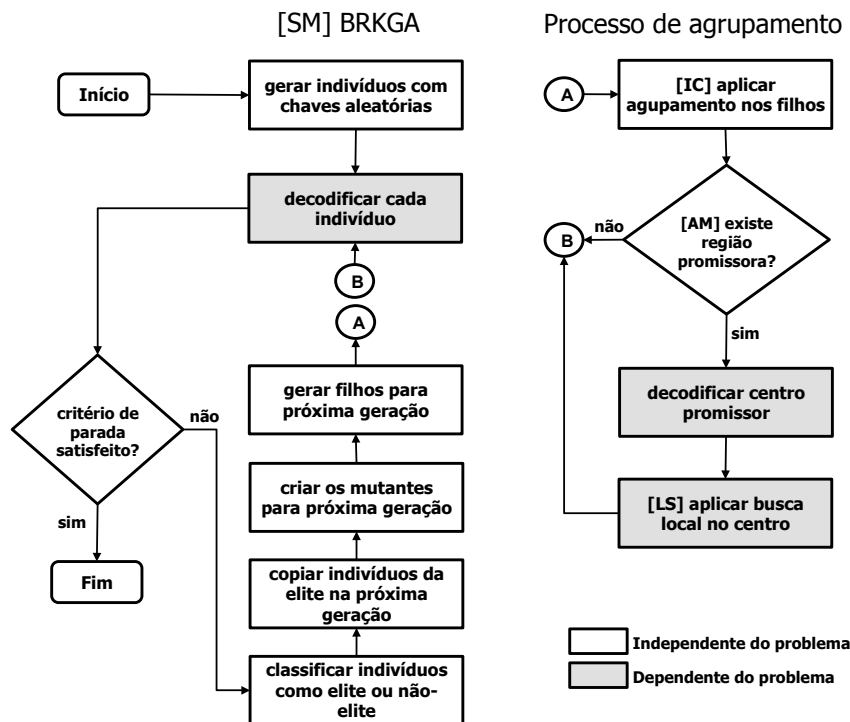


Figura 2: Fluxograma do CS+BRKGA. Adaptado de [Chaves et al., 2016]

- Fração de população elite (p_e) = 0,2.
- Fração de população mutante (p_m) = 0,15.
- Probabilidade de um alelo elite ser escolhido (ρ) = 0,6.
- Número de gerações no *BRKGA* = 200.

Os resultados computacionais estão nas Tabelas 1, 2 e 3. Os significados dos nomes das colunas nas tabelas são: **Nome:** nome da instância; **Classe:** classificação da instância; **B:** número de berços de cada instância; **N:** número de navios de cada instância; **P:** número de perfis diferentes para cada navio; **G:** número de guindaste de cada instância; **FO melhor:** valor da função objetivo com melhor solução; **FO Média:** valor da função objetivo com a média das melhores soluções; **BRKGA:** valor da solução encontrada em Lalla-Ruiz et al. [2014]; **Tempo:** Tempo total do algoritmo; **CS+BRKGA:** valor solução encontrada pelo método proposto; **Média:** valor da solução média encontrada pelo método proposto; **T*:** tempo médio para encontrar a melhor solução; **T:** tempo total médio para rodar o algoritmo; **Sobj:** Distância da solução em relação ao “upper bound” do Cplex, o valor 100, significa que encontrou o ótimo.

Na Tabela 1 os resultados obtidos pelo método proposto demonstram a robustez do CS+BRKGA ao conseguir soluções similares ao encontrado pelo BRKGA proposto em Lalla-Ruiz et al. [2014]. Vale ressaltar que as soluções do BRKGA de Lalla-Ruiz et al. [2014] são soluções ótimas para todas as instâncias de Vacca et al. [2013]. A otimalidade das soluções foi provada pelo Cplex. Os Gaps em negrito representam as soluções que o CS+BRKGA não conseguiu atingir o ótimo. O método CS+BRKGA encontrou a solução ótima em 17 das 20 instâncias, e quando não obteve a solução ótima, as soluções eram muito próximas as do BRKGA [Lalla-Ruiz et al., 2014]. Além disso, ao analisar as médias das soluções, nota-se a estabilidade do CS+BRKGA, menos para a instância c4-5 com 20 navios. Para este caso a média ficou um pouco abaixo do valor esperado. Como comentado anteriormente o CS+BRKGA usa os resultados do BRKGA como condição para



iniciar o CS. As melhorias das soluções do BRKGA obtidas pelo CS ficaram em média de 0,001% com melhora em seis soluções.

Tabela 1: Resultados para as instâncias de [Vacca et al., 2013]

Classe	Nome	Instância	B	N	P	G	BRKGA Ruiz			CS+BRKGA			
							FO Melhor	T	FO Melhor	FO Média	T*	T	Gap
B1	H1	c1-1	3	10	10	8	790735	0,40	790735	790735,00	0,024	2,268	0,000
B1	H1	c1-2	3	10	20	8	791011	0,64	791011	791011,00	0,029	2,367	0,000
B1	H1	c1-3	3	10	30	8	791045	0,54	791045	791045,00	0,006	2,275	0,000
B1	L1	c1-4	3	10	10	8	515902	0,18	515902	515902,00	0,001	2,591	0,000
B1	L1	c1-5	3	10	20	8	518049	0,26	516348	516348,00	0,012	2,235	0,328
B1	L1	c2-1	3	10	30	8	518084	0,37	518084	518084,00	0,005	2,308	0,000
B1	L2	c2-2	3	10	10	8	564831	0,23	564831	564831,00	0,001	2,356	0,000
B1	L2	c2-3	3	10	20	8	564867	0,42	564867	564867,00	0,001	2,316	0,000
B1	L2	c2-4	3	10	30	8	564903	0,59	564903	564903,00	0,001	2,246	0,000
B1	H2	c2-5	3	10	10	8	733276	0,32	733276	733276,00	0,001	2,327	0,000
B1	H2	c3-1	3	10	20	8	735646	0,58	735646	735646,00	0,005	2,449	0,000
B1	H2	c3-2	3	10	30	8	735682	0,59	735682	735682,00	0,005	2,302	0,000
B2	H1	c3-3	3	15	10	13	1170783	0,54	1170448	1170442,75	0,553	11,945	0,029
B2	H2	c3-4	3	15	10	13	1272247	1,07	1272236	1272236,00	0,292	8,318	0,001
B2	L1	c3-5	3	15	10	13	1098411	1,20	1098411	1098411,25	0,165	7,446	0,000
B2	L2	c4-1	5	15	10	13	890211	0,54	890211	890211,00	0,094	7,715	0,000
B3	H1	c4-2	5	20	10	13	1347208	2,17	1347208	1347157,50	22,263	26,890	0,000
B3	H2	c4-3	5	20	10	13	1437301	3,89	1437301	1437215,25	10,445	25,174	0,000
B3	L1	c4-4	5	20	10	13	1269229	2,93	1269228	1268226,25	8,143	28,960	0,000
B3	L2	c4-5	5	20	10	13	1078337	2,71	1077584	1077352,75	0,758	25,785	0,070
Média													0,021

Para o conjunto de instâncias da Tabela 2, o CS+BRKGA conseguiu melhorar três soluções (soluções em negrito), e também encontrou três soluções idênticas as encontradas pelo método de [Lalla-Ruiz et al., 2014]. As soluções médias são idênticas as melhores soluções em praticamente todas as instâncias desse conjunto, mostrando a robustez do método. Em relação a coluna Subj é possível ver a robustez do método proposto, pois a distância da melhor solução em relação ao limitante do Cplex é pequena quando não se encontra o valor ótimo, tendo valor médio de 97,71. O CS+BRKGA melhorou as soluções do BRKGA implementado neste trabalho em cinco instâncias, com média de 0,011%.

Tabela 2: Resultados para as instâncias de [Lalla-Ruiz et al., 2014]

Instância	Berços	Navios	Profiles	Guindaste	Ubmilp	Melhor	BRKGA Ruiz			CS+BRKGA			
							Sobj	Tempo total(S)	Melhor	Média	Sobj	Tempo Médio	Tempo Total
c1-1	3	12	10	8	596910	596910	100,00	0,53	596909	596909,00	99,9998	0,002	3,703
c1-2	3	12	10	8	1029695	1026629	99,7022	0,21	1026628	1026628,00	99,7021	0,005	5,090
c1-3	3	12	10	8	1490528	1446182	97,02	1,84	1443230	1425432,38	96,83	0,446	9,198
c1-4	3	12	10	8	1377586	1370274	99,47	3,45	1370650	1368807,38	99,50	6,852	11,445
c1-5	3	12	10	8	1415291	1370253	96,82	4,47	1373906	1369153,63	97,08	10,259	18,220
c2-1	3	12	10	8	1044298	1044298	100,00	0,45	1044298	1044298,00	100,00	0,009	4,648
c2-2	3	12	10	8	726295	724753	99,79	0,1	724752	724752,00	99,79	0,001	4,049
c2-3	3	12	10	8	1004156	1004156	100,00	0,5	1004156	1004156,00	100,00	0,003	4,482
c2-4	3	12	10	8	1471060	1424927	96,86	8,38	1428502	1427983,50	97,11	7,802	13,603
c2-5	3	12	10	8	1622232	1480347	91,25	17,63	1412280	1352331,50	87,06	7,882	21,500
Média							98,09					97,71	

Na Tabela 3, o CS+BRKGA se mostrou competitivo melhorando as soluções de nove das onze instâncias. O método proposto também apresentou bom desempenho para os tempos computacionais, mesmo quando se aumentou o número de elementos do problema. Analisando a coluna Subj, o CS+BRKGA conseguiu valor médio de 98,27. As melhores soluções estão em negrito. A porcentagem das melhorias do CS+BRKGA em relação à condição inicial do BRKGA implementado ficou com média de 0,02%. As melhorias foram para todas as doze instâncias utilizadas.

6. Conclusão

Este trabalho propôs combinar a metaheurística BRKGA com o método CS aplicado ao problema de alocação tática de berços (TBAP). Os resultados computacionais consideraram as ins-



Tabela 3: Resultados para as instâncias de [Giallombardo et al., 2010]

Nome	Classe	Inst.	Berços	Navios	Profiles	Guindaste	Ubmilp	BRKGA Ruiz		CS+BRKGA com PR				
								Sobj	Tempo Total	Melhor	Média	Sobj	Tempo Médio	Tempo Total
C3-1	A1	H1	5	20	10	13	1.383.614	97,80	3,75	1.353.290	1.351.355,25	97,81	0,275	0,380
C3-2		H1	5	20	20	13	1.384.765	97,81	3,85	1.354.441	1.353.274,00	97,81	0,045	0,390
C3-3		H1	5	20	30	13	1.383.119	97,82	3,57	1.355.365	1.354.356,25	97,99	0,020	0,398
C3-4		H2	5	20	10	13	1.474.082	97,86	3,5	1.466.718	1.465.317,63	99,50	0,118	0,336
C3-5		H2	5	20	20	13	1.474.561	97,86	3,53	1.443.537	1.442.542,63	97,90	0,108	0,329
C3-6		H2	5	20	30	13	1.474.631	97,89	3,56	1.444.118	1.443.510,63	97,93	0,289	0,335
C4-1		L1	5	20	10	13	1.298.356	98,00	2,56	1.283.106	1.257.871,00	98,83	0,302	0,360
C4-2		L1	5	20	20	13	1.298.542	98,00	3,03	1.272.355	1.270.830,38	97,98	0,164	0,345
C4-3		L1	5	20	30	13	1.298.542	98,00	2,76	1.272.047	1.270.853,00	97,96	0,303	0,363
C4-4		L2	5	20	10	13	1.103.212	98,06	2,89	1.096.286	1.089.137,25	99,37	0,095	0,287
C4-5		L2	5	20	20	13	1.103.516	98,06	3,20	1.082.166	1.081.129,75	98,07	0,114	0,281
C4-6		L2	5	20	30	13	1.103.551	98,07	1,43	1.082.390	1.081.720,25	98,08	0,276	0,292
Média								97,94					98,27	

tâncias de Giallombardo et al. [2010], Vacca et al. [2013] e Lalla-Ruiz et al. [2014]. De modo geral, quando se analisa os resultados obtidos o método híbrido proposto CS+BRKGA se mostra eficiente na resolução das instâncias testadas. O TBAP é um problema complexo, pois possui uma grande quantidade de variáveis e muitas restrições, assim uma decodificação da solução de chaves aleatórias mais refinada é necessária para realizar uma busca mais eficiente no espaço de soluções. Apesar da complexidade do problema, é possível observar que os resultados (Tabelas 1 a 3) foram satisfatórios. Nas instâncias de Vacca et al. [2013] o CS+BRKGA encontrou a solução ótima em 17 das 20 instâncias. Para as instâncias de Lalla-Ruiz et al. [2014] o CS+BRKGA conseguiu melhorar três soluções. Já para as instâncias de Giallombardo et al. [2010] houve melhora nas soluções, totalizando nove das onze instâncias analisadas. Observando as colunas Sobj das Tabelas 2 e 3 é possível afirmar que o método proposto é estavel na busca de soluções viáveis, além do baixo tempo computacional (Colunas “Tempo total”).

Conclui-se pelos bons resultados obtidos que o método híbrido CS+BRKGA é eficiente e robusto, podendo ser aplicado a outros problemas da literatura, uma vez que necessita apenas alterar a decodificação e a busca local para o problema específico. Para trabalhos futuros, espera-se resolver outros problemas de otimização, analisar outras heurísticas de busca local mais eficientes e estudar alguns métodos para calibragem automática de parâmetros.

Agradecimentos

Os autores agradecem a FAPESP (processos 2012/17523-5 e 2016/07135-7), CNPq (301836/2014-0, 307330/2015-0) e CAPES.

Referências

- (2011). Associação brasileira dos terminais portuários - relatório anual, exercício de 2010. Technical report.
- Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, 6(2):154–160.
- Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J., e Lusby, R. (2009). Models for the discrete berth allocation problem: A computational comparison. *Transportation Research*, 47(4):461 – 473.
- Chaves, A. A., Lorena, L. A. N., Senne, E. L. F., e Resende, M. G. (2016). Hybrid method with cs and brkga applied to the minimization of tool switches problem. *Computers & Operations Research*, 67:174–183.
- Cordeau, J.-F. e Laporte, G. (2005). Models and tabu search heuristics for the berth allocation problem. *Transportation Science*, 39(4):525–538.
- Giallombardo, G., Moccia, L., Salani, M., e Vacca, I. (2010). Modeling and solving the tactical berth allocation problem. *Transportation Research Part B*, 44(2):232–245.



- Gonçalves, J. F. e Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525.
- Hansen, P., Oguz, C., e Mladenovic, N. (2008). Variable neighborhood search for minimum cost berth allocation. *European Journal of Operational Research*, 191(3):636–649.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press.
- Imai, A., Nishimura, E., e Papadimitriou, S. (2001). The dynamic berth allocation problem for a container port. *Transportation Research Part B*, 35:401–417.
- Imai, A., Sun, X., Nishimura, E., e Papadimitriou, S. (2005). Berth allocation in a container port: using a continuous location space approach. *Transportation Research Part B*, 39(3):199–221.
- Imai, A., Chen, H. C., Nishimura, E., e Papadimitriou, S. (2008). The simultaneous berth and quay crane allocation problem. *Transportation Research Part E*, 44:900–920.
- Lalla-Ruiz, E., González-Velarde, J. L., Melián-Batista, B., e Moreno-Vega, J. M. (2014). Biased random key genetic algorithm for the tactical berth allocation problem. *Applied Soft Computing*, 22:60 – 76.
- Liang, C., Huang, Y., e Yang, Y. (2009). A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. *Computers & Industrial Engineering*, 56 (3):1021 – 1028.
- Mauri, G. R., de Oliveira, A. C. M., e Lorena, L. A. N. (2008a). Heurística baseada no simulated annealing aplicada ao problema de alocação de berços. *Gestão da Produção, Operações e Sistemas*, 3(1):113–127.
- Oliveira, A. A., A. C. M. and Chaves e Lorena, L. A. N. (2013). Clustering search. *Pesquisa Operacional*, 33(1):105–121.
- Raa, B., Dullaert, W., e Schaeren, R. V. (2011). An enriched model for the integrated berth allocation and quay crane assignment problem. *Expert Systems with Applications*, 38(11):14136 – 14147.
- Resende, M. G. C. e Ribeiro, C. C. (2005). Grasp with path-relinking: Recent advances and applications. p. 29–63.
- Stahlbock, R. e Voß, S. (2008). Operations research at container terminals: a literature update. *OR Spectrum*, 30(1):1–52.
- Steenken, D., Voß, S., e Stahlbock, R. (2004). Container terminal operation and operations research - a classification and literature review. *OR Spectrum*, 26(1):3–49.
- Vacca, I., Salani, M., e Bierlaire, M. (2013). An exact algorithm for the integrated planning of berth allocation and quay crane assignment. *Transportation Science*, 47(2):148–161. ISSN 1526-5447.