

## MÉTODOS EXATOS E UMA ABORDAGEM HEURÍSTICA PARA O PROBLEMA DE ALOCAÇÃO DE BERÇOS

**Mateus Pereira Martin, Ellen Marianne Bernal Cavalheiro,  
Flávia Barbosa, Christiano Lyra Filho**

Faculdade de Engenharia Elétrica e de Computação - UNICAMP  
Cidade Universitária Zeferino Vaz. Av. Albert Einstein, 400. CEP: 13083-852, Campinas - SP  
chrlyra@densis.fee.unicamp.br

**Antonio Carlos Moretti**

Faculdade de Ciências Aplicadas - UNICAMP  
R. Pedro Zaccaria, 1300. CEP 13484-350, Limeira - SP

### RESUMO

O Problema de Alocação de Berços procura encontrar a melhor alocação de navios aos berços de serviços portuários em um dado horizonte de planejamento. Este trabalho estuda a aplicação de Algoritmo Genético Baseado em Chaves Aleatórias Viciadas e de métodos exatos de programação inteira para solução do problema. As abordagens por programação inteira interpretam o problema como um caso especial de cortes bidimensionais e como um problema de *scheduling*. Os estudos de casos realizados indicam que o método heurístico baseado em chaves aleatórias viciadas é a abordagem mais indicada para o problema. Indicam também que os métodos baseados em programação inteira são competitivos e que há espaço para aperfeiçoamento de métodos para abordagem de problemas exatos.

**PALAVRAS CHAVE.** Alocação de Berços, Algoritmo Genético de Chaves Viciadas Aleatórias, Programação Inteira.

**Área Principal:** Logística & Transportes.

### ABSTRACT

The Berth Allocation Problem seeks to find the best allocation of ships to port berths in a given planning horizon. This paper studies the application of Biased Random Keys Genetic Algorithm (BRKGA) and exact methods of mixed integer linear programming to solve the problem. The integer programming approaches interpret the problem as a special case of Two-Dimensional Cutting Stock Problem and as a Scheduling Problem. The case studies carried out indicate that the heuristic method based on BRKGA is the most appropriate approach to the problem. They also indicate that the integer programming based methods are competitive and that there is space for improvements in the methods for exact approaches.

**KEYWORDS.** Berth Allocation, Biased Random Keys Genetic Algorithm, Integer Programming.

**Main Area:** Logistics & Transports.

## 1. Introdução

De acordo com relatório da Conferência da Organização das Nações Unidas de Comércio e Desenvolvimento (UNCTAD 2014), a logística marítima mundial cresceu 3,8% em 2013, transportando aproximadamente 9,6 bilhões de toneladas. Segundo a Secretaria de Comércio Exterior do Brasil (SECEX 2013), o comércio brasileiro exportou US\$ 466 bilhões, equivalendo a 688 milhões de toneladas em 2012; cerca de 80% do fluxo comercial e 95% do volume de carga foi transportado por via marítima. Esses dados indicam a relevância do transporte marítimo no comércio nacional e mundial.

Para atender o aumento da demanda das últimas décadas, os portos têm procurado melhorar a infraestrutura e os processos. Bierwirth & Meisel (2010) analisaram as relações e modelos de planejamento estratégico e operacional em terminais portuários em três grupos: áreas de *seaside*, *yard* e *landside*. Em geral, a primeira decisão em *seaside* é a determinação de qual porção do berço portuário, em dado período de tempo, estará disponível para as operações de carregamento e descarregamento de determinado navio. Essa decisão é conhecida como Problema de Alocação de Berços (PAB). Ainda na área de *seaside* ocorrem outras operações, como a alocação de guindastes necessários ao manuseio das cargas. As operações de *yard* envolvem o gerenciamento do pátio portuário, como a movimentação das cargas aos seus respectivos navios. Por sua vez, as operações de *landside* lidam com o recebimento e despacho das cargas para outras vias modais.

Limitações e políticas particulares de cada porto levam a diversas extensões para o PAB. As diferenças dizem respeito, por exemplo a considerações físicas (profundidade do berço, tamanho do calado, quantidade de navios atendida pelos berços), e a restrições temporais (tempo de atendimento proporcional ao tamanho do navio e horário de chegada conhecido). Para cada uma dessas extensões modelos de Pesquisa Operacional têm sido desenvolvidos para otimizar a utilização de recursos e de instalações dos portos. É comum que o objetivo seja a minimização dos tempos de espera dos navios para atendimento ou algum objetivo similar.

O PAB pode ser descrito como um problema de *scheduling* e de *packing*. É um problema *NP-Difícil*, como indicado por Hansen & Oğuz (2003) e Imai et al. (2008). As variações desses problemas têm sido abordados por métodos exatos e por metaheurísticas (Stahlbock & Voß 2008, Bierwirth & Meisel 2010). Abordagens como Cordeau et al. (2005) e Lalla-Ruiz et al. (2014) indicam que em grande quantidade de cenários, as metaheurísticas obtêm melhores soluções e em menor tempo computacional que os métodos exatos, que utilizam pacotes comerciais de resolução (*solvers*).

Este trabalho considera duas abordagens distintas ao Problema de Alocação de Berços. A primeira abordagem é baseada em Programação Inteira, a partir das metodologias de “*scheduling*” e de “*cutting e packing*”, desenvolvidas por Barbosa (2014). A segunda abordagem é baseada em Algoritmos Genéticos com Chaves Aleatórias Viciadas (do inglês, *Biased Random Key Genetic Algorithm* – BRKGA).

As metodologias são avaliadas com duas análises comparativas de desempenho para o PAB discreto e dinâmico. A primeira análise destaca os melhores resultados obtidos pelas duas abordagens utilizadas. A segunda análise ressalta a evolução das soluções obtidas pelas duas abordagens ao longo de seus processos de busca. Os estudos de casos realizados obtiveram melhores resultados com a metaheurística. Eles mostram também que as abordagens exatas são competitivas e que há perspectivas para aperfeiçoamento dessas técnicas.

## 2. Revisão da literatura

Ao longo das últimas duas décadas diversos modelos foram propostos para o PAB. Bierwirth & Meisel (2010) sugeriram a classificação do PAB segundo os espaços dos berços, a chegada de navios e os tempos de processamento. Em relação às características espaciais, os berços

podem ser classificados como discretos, contínuos ou híbridos (Imai et al. 2005). Na abordagem discreta, a plataforma de embarque e desembarque é compreendida como um conjunto finito de berços com comprimentos fixos. Assim, pode ser interpretado como um problema de *scheduling*, em que os berços representam máquinas e os navios representam tarefas. Na abordagem contínua, os navios podem atracar em qualquer lugar na plataforma, neste caso, o PAB é visto como um problema de corte bidimensional. O caso híbrido é uma composição de ambas as abordagens.

Em relação à chegada dos navios, Imai et al. (2001) indicaram os casos estático e dinâmico. Na forma estática, todos os navios estão disponíveis no início do horizonte de planejamento. Na forma dinâmica, nem todos os navios estão disponíveis no horizonte de planejamento, porém os tempos de chegada são previamente conhecidos – naturalmente, os navios só poderão atracar após sua chegada ao porto. Segundo Bierwirth & Meisel (2010), os tempos de processamento dos navios podem ser considerados fixos ou dependentes de aspectos como o número de guindastes utilizados e a posição dos berços.

Cordeau et al. (2005) consideraram duas formulações matemáticas e uma metaheurística baseada em busca tabu para o PAB discreto e dinâmico. Os autores desenvolveram um modelo baseado em roteamento de veículos (do inglês, *Multidepot Vehicle-Routing Problem with Time Windows* – MDVRPTW) capaz de lidar com ponderamentos nos tempos de serviço, assim como janelas de tempo na atracação dos navios. A outra formulação matemática analisada foi um modelo de Imai et al. (2001). A heurística busca tabu desenvolvida foi baseada no modelo discreto proposto (MDVRPTW) e também foi estendida para o PAB contínuo e dinâmico. Os autores propuseram 90 instâncias baseadas no porto de Gioia Tauro, na Itália. Os modelos de programação linear inteira foram executados por duas horas, isto é, o algoritmo *Branch & Bound* foi truncado. A metaheurística obteve melhores resultados e em tempo computacional menor que os métodos exatos.

O Problema de Alocação de Guindastes (do inglês, *Quay Crane Allocation Problem* – QCAP) está associado ao PAB, pois os tempos de processamento dos navios estão associados ao número de guindastes utilizados. O emprego ineficiente desses recursos pode criar gargalos para as operações de *seaside*. Em portos de grandes dimensões, é comum a aplicação de penalidades financeiras, caso determinado navio não seja atendido antes de período limite acordado em contrato. Assim, um terminal competitivo precisa maximizar a eficiência das operações sobre um número limitado de guindastes. Imai et al. (2008) propuseram uma heurística baseada em algoritmos genéticos para lidar com o PAB e o QCAP, originando um problema de alocação simultânea (B&CAP).

Giallombardo et al. (2010) propuseram modelos de programação quadrática e linear para o PAB tático (do inglês, *Tactical Berth Allocation Problem* – TBAP); trata-se de um problema similar ao B&CAP. Os autores também desenvolveram um modelo heurístico para resolver instâncias reais. Lalla-Ruiz et al. (2014) propuseram um algoritmo genético baseado em chaves aleatórias viciadas (BRKGA) ao TBAP. A metaheurística foi comparada às abordagens de Giallombardo et al. (2010) e Vacca et al. (2013). Os autores concluíram que, mesmo para instâncias de pequeno porte, o problema foi difícil de ser resolvido pelo *solver* CPLEX; o BRKGA proposto foi capaz de encontrar soluções de qualidade em tempo computacional melhor.

Pomari & Chaves (2014) propuseram um algoritmo híbrido baseado na metaheurística BRKGA para geração de soluções e na técnica de *Clustering Search* (CS) para agrupamento e busca local para o PAB. Os autores destacaram que a implementação do CS foi simplificada pelo uso do BRKGA como gerador de soluções, visto que as soluções candidatas são codificadas como chaves aleatórias.

Barbosa (2014) apresentou duas formulações para abordagens do PAB por programação inteira, bem como uma heurística baseada no conceito de “*First Come, First Served*”. Os modelos de programação inteira utilizam os *solvers* GLPK e CPLEX. Seis classes de instâncias foram propostos, considerando 10, 15, 20, 25, 30, 35 e 40 navios, com 10 problemas em cada classe. Soluções ótimas (exatas) foram obtidas apenas nos cenários com 10 navios. O modelo baseado

em *scheduling* levou a melhores resultados do que os obtidos com os de cortes para a maioria das instâncias.

### 3. Modelos Matemáticos e Metaheurística BRKGA

Os modelos de programação inteira de Barbosa (2014) são apresentados na Seção 3.1. A metaheurística BRKGA desenvolvida para o PAB é apresentada na Seção 3.2.

#### 3.1. Modelos de Programação Linear Inteira

Inicialmente, Barbosa (2014) considerou o PAB através de um Problema de Corte Bidimensional. Os modelos foram desenvolvidos para um terminal portuário de carga granel com dois berços, onde os navios poderiam atracar em qualquer berço disponível. Assim, no modelo proposto, os navios são considerados retângulos, cujos comprimentos são os tempos de processamento e as larguras dos retângulos estão associados aos berços. A função objetivo procura minimizar os tempos de espera dos navios para atendimento – este objetivo comumente usado na literatura.

A Tabela 1 apresenta a notação utilizada no modelo de programação linear inteira baseado em corte.

Tabela 1: Modelo de Corte - Variáveis e Parâmetros

$a_{ik}$	variável binária cujo valor é (1) se o navio $i$ está à esquerda do navio $k$ , (0) caso contrário.
$b_{ik}$	variável binária cujo valor é (1) se o navio $i$ está à direita do navio $k$ , (0) caso contrário.
$c_{ik}$	variável binária cujo valor é (1) se o navio $i$ está atrás do navio $k$ , (0) caso contrário.
$d_{ik}$	variável binária cujo valor é (1) se o navio $i$ está em frente ao navio $k$ , (0) caso contrário.
$x_i$	posição do início do atendimento do navio no horizonte de planejamento.
$y_i$	variável inteira cujo valor é (1) ou (2) e associa o navio $i$ a um dos dois possíveis berços.
$p_i$	tempo de processamento do navio $i$ .
$e_i$	período de chegada do do navio $i$ .
$M$	número grande arbitrário.

Usando a notação da Tabela 1, o PAB pode ser caracterizado pelas Equações (1) - (10)

$$\min \sum_i (x_i - e_i) \quad (1)$$

$$\text{s.a. } x_i + p_i \leq x_k + (1 - a_{ik})M \quad \forall (i, k), i < k \quad (2)$$

$$x_k + p_k \leq x_i + (1 - b_{ik})M \quad \forall (i, k), i < k \quad (3)$$

$$y_i + 1 \leq y_k + (1 - c_{ik})M \quad \forall (i, k), i < k \quad (4)$$

$$y_k + 1 \leq y_i + (1 - d_{ik})M \quad \forall (i, k), i < k \quad (5)$$

$$a_{ik} + b_{ik} + c_{ik} + d_{ik} \geq 1 \quad \forall (i, k), i < k \quad (6)$$

$$x_i \geq e_i \quad \forall i \quad (7)$$

$$1 \leq y_i \leq 2 \quad \forall i \quad (8)$$

$$a_{ik}, b_{ik}, c_{ik}, d_{ik} \in \{0, 1\} \quad \forall (i, k) \quad (9)$$

$$y_i \geq 0 \text{ e inteiro.} \quad \forall i \quad (10)$$

$$x_i \geq 0 \quad \forall i \quad (11)$$

A Função-objetivo (1) calcula a soma dos tempos de espera de todos os navios. As Restrições (2) e (3) garantem que não haverá sobreposição de navios na direção  $x$ , enquanto as

Restrições (4) e a (5) evitam a sobreposição na direção  $y$ . A Restrição (6) garante que a verificação de não sobreposição feita para restrições anteriores ocorra somente para navios no mesmo berço. A Restrição (7) não permite que o navio seja atracado antes de sua chegada. A Restrição (8) é uma canalização da variável  $y$  que indica em qual berço o navio foi alocado. As Restrições (9), (10) e (11) especificam a natureza das variáveis utilizadas.

O segundo modelo proposto por Barbosa (2014) foi baseado na ideia de sequenciamento de máquinas (*scheduling*), onde os berços são vistos como máquinas e os navios como tarefas. A Tabela 2 apresenta a notação adicional utilizada no modelo. As demais variáveis estão descritas na Tabela 1.

Tabela 2: Modelo de Scheduling - Variáveis

$z_{ij}$	variável binária cujo valor é (1) se o navio $i$ é alocado ao berço $j$ , (0) caso contrário.
$a_{ik}$	variável binária cujo valor é (1) se o navio $i$ está à esquerda do navio $k$ , (0) caso contrário.
$x_i$	posição do início do atendimento do navio $i$ no horizonte de planejamento.

Utilizando a notação descrita nas Tabelas 1 e 2, a representação de PAB em um modelo de *scheduling* é caracterizado pelas Equações (12) - (19).

$$\min \sum_i (x_i - e_i) \quad (12)$$

$$\text{s.a.} \quad \sum_j z_{ij} = 1 \quad \forall i \quad (13)$$

$$z_{ij} + z_{kj} - a_{ik} - a_{ki} \leq 1 \quad \forall i \neq k, j \quad (14)$$

$$x_i + p_i - (1 - a_{ik})M \leq x_k \quad \forall (i, k) \quad (15)$$

$$z_{ij} + z_{kh} + a_{ik} + a_{ki} \leq 2 \quad \forall i \neq k, \forall j \neq h \quad (16)$$

$$x_i \geq e_i \quad \forall i \quad (17)$$

$$x_i \geq 0 \quad \forall i. \quad (18)$$

$$a_{i,k}, z_{i,j} \in \{0, 1\}. \quad \forall (i, k, j) \quad (19)$$

A Função-objetivo (12) minimiza a soma dos tempos de espera. A Restrição (13) garante que cada navio será alocado em apenas um berço. A Restrição (14) mostra que se os navios  $i$  e  $k$  estão no mesmo berço, então ou o navio  $i$  está à direita do navio  $k$  ou o contrário ocorre. A Restrição (15) não permite que os navios estejam sobrepostos sobre o horizonte de planejamento. A (16) reforça que a variável  $a_{i,k}$  só existe para navios que estão no mesmo berço. Por fim, a Restrição (17) não permite que os navios sejam atracados antes de sua chegada ao cais. As Restrições (18) e (19) especificam a natureza das variáveis utilizadas.

### 3.2. Algoritmo Genético Baseado em Chaves Aleatórias Viciadas

A metaheurística BRKGA, proposta por Gonçalves & Resende (2011) para problemas de otimização combinatória, é baseada no algoritmo genético de chaves aleatórias (do inglês, *Random-Key Genetic Algorithm* - RKGA) de Bean (1994). O RKGA e o BRKGA representam uma solução do problema com um vetor de  $n$  chaves, onde cada chave é um número real gerado aleatoriamente, no intervalo contínuo  $[0, 1)$ . Para calcular o custo da solução (*fitness*), utiliza-se um decodificador que ordena os elementos do vetor de chaves e gera uma permutação correspondente aos índices dos elementos ordenados. O algoritmo começa com uma população inicial de  $p$  vetores de  $n$  chaves aleatórias e a cada iteração a população é dividida em duas subpopulações. A primeira

subpopulação corresponde ao pequeno conjunto de  $|p_e|$  elementos que possuem os melhores valores de *fitness*, denominado *conjunto elite*, e o restante da população compõem o *conjunto não-elite*. O conjunto elite é automaticamente copiado para a geração seguinte, juntamente com novos vetores de chaves aleatórias, chamados de mutantes  $p_m$ . O restante da população é completada com  $|p - p_e - p_m|$  vetores gerados via *crossover*. Segundo Resende (2013), os mutantes têm o mesmo papel dos operadores de mutação nos algoritmos genéticos clássicos; sendo assim, procuram evitar convergência para um ótimo local.

O que difere o BRKGA ao RKGA é a seleção dos progenitores no processo de realização de *crossover*. No RKGA, os dois progenitores são escolhidos entre os indivíduos da população. No BRKGA, um progenitor é proveniente do conjunto elite e outro progenitor do conjunto não-elite. A Figura 1 apresenta o fluxograma do algoritmo BRKGA<sup>1</sup>.

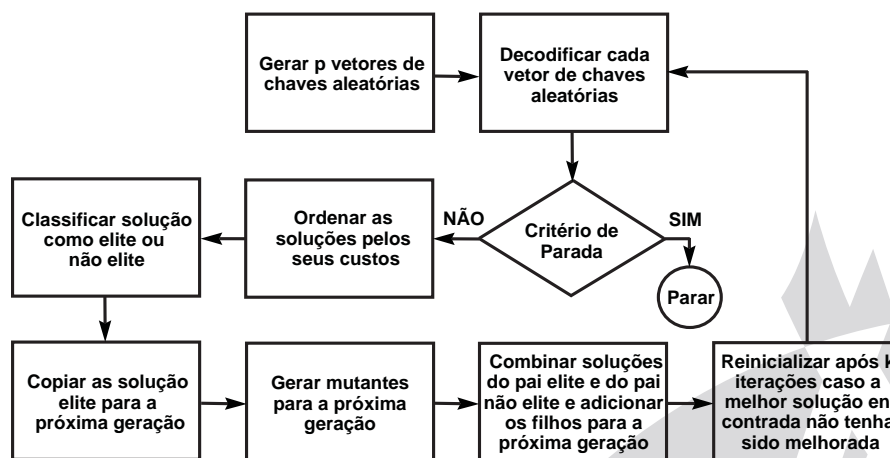


Figura 1: Fluxograma do algoritmo BRKGA

### 3.2.1. BRKGA Proposto

As soluções candidatas são cromossomos representados como vetores aleatórios de números reais, gerados uniformemente no intervalo de  $[0, 1)$  – e, o tamanho de cada cromossomo corresponde a quantidade de navios  $N$ . As  $N$  posições do cromossomo estão associadas à ordem em que os navios serão alocados aos berços. Para decodificar a solução, as chaves aleatórias do cromossomo são organizados em ordem crescente. Assim, também se obtém uma ordem de alocação. A Figura 2 apresenta uma solução  $N = \{1, 2, 3, 4, 5, 6\}$ , que representa a quantidade de chaves aleatórias, isto é, de navios existentes na solução. Em seguida, esta solução é decodificada e o novo vetor de alocação aos berços obtido é  $N = \{6, 4, 1, 5, 2, 3\}$ .

		número de navios					
		1	2	3	4	5	6
cromossomo		0.583	0.897	0.932	0.369	0.734	0.183
	decodificação	0.183	0.369	0.583	0.734	0.897	0.932
		6	4	1	5	2	3

Figura 2: Representação computacional do BRKGA

<sup>1</sup>A figura é baseada em slides de Resende: [www.mauricio.resende.info/](http://www.mauricio.resende.info/)

A Tabela 3 apresenta os tempos de chegada e de processamento para uma instância de 6 navios e 2 berços. A Figura 3 exemplifica uma solução para a situação apresentada na Figura 2. Pode-se notar que, com exceção dos navios  $N4$  e  $N6$  que foram alocados no seu momento de chegada, os demais navios tiveram de esperar ao início de suas atividades. É importante destacar que a decodificação do cromossomo fornece um vetor de ordem de alocação dos navios. Para realizar a alocação navio-berço, uma programação é realizada: o navio  $i$  é alocado ao primeiro berço  $k$  disponível.

Tabela 3: Exemplo de instância do PAB

Navio	N1	N2	N3	N4	N5	N6
Chegada	12	22	27	11	21	5
Processamento	7	22	13	14	7	18

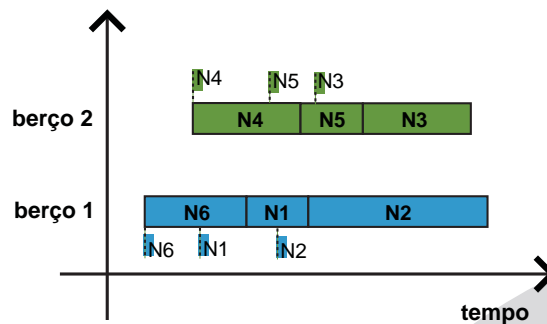


Figura 3: Possível solução da instância de exemplo

Assim como os modelos da Seção 3.1, o objetivo do BRKGA é minimizar o tempo de espera dos navios para atendimento. Dessa forma, para a solução candidata ser avaliada, deve-se calcular o momento de início de processamento ( $t_i$ ) e tempo de espera de cada navio ( $e_i$ ), de acordo com os tempos de chegada ( $t_{cheg_i}$ ) e de processamento ( $t_{proc_i}$ ). As Equações (20) e (21) apresentam a forma de cálculo. Este procedimento considera os berços separadamente, isto é, para cada berço, o primeiro navio alocado iniciará suas atividades no seu tempo de chegada ao porto. Essa forma de cálculo evita infactibilidades em relação ao início do atendimento de dado navio antes de sua chegada ao porto.

$$t_i = \max\{t_{cheg_i}; t_{i-1} + t_{proc_{i-1}}\} \quad (20)$$

$$e_i = t_i - t_{cheg_i} \quad (21)$$

A função *fitness* adotada apresenta um escalonamento da função objetivo à solução  $j$  e é indicada na Equação (22). Portanto, a função deve ser maximizada.

$$fitness(j) = \frac{1}{1 + \sum_{i=1}^N e_i} \quad (22)$$

A população do BRKGA é composta por  $n$  indivíduos e representada por uma matriz de dimensão  $[n \times N]$ , onde  $N$  indica o número de navios. As soluções candidatas da população são

vetores aleatórios com  $N$  elementos. A avaliação do *fitness* de cada solução é feita de acordo com o procedimento apresentado pelas Equações (20), (21) e (22). O operador de *crossover* utilizado é o *crossover* uniforme, onde a probabilidade ( $prob_{el}$ ) do novo indivíduo herdar características do progenitor do conjunto elite deve ser maior.

O método de reinicialização proposto por Resende (2013) também foi utilizado para contribuir na busca de melhores soluções. O algoritmo é reinicializado a cada  $T_1$  gerações. O critério de parada adotado foi a estabilização da melhor solução encontrada, isto é, se não houver melhora em  $T_2$  gerações, o algoritmo é encerrado.

#### 4. Estudos de Casos

As implementações foram executadas em um computador com processador Intel Core 2 Quad (3.30 GHz), de 4 GB de memória RAM e com sistema operacional *Windows*. Os modelos exatos foram implementados na plataforma *AIMMS 4.0*<sup>2</sup>, utilizando o *solver GUROBI 6.0*. A metaheurística BRKGA proposta foi implementada em ambiente *Matlab R2014a*.

Os cenários utilizados são os mesmos de Barbosa (2014) para as classes de 10, 20, 30 e 40 navios. A autora gerou as instâncias em ambiente *Matlab*, considerando a chegada dos navios como uma distribuição uniforme no intervalo  $[0, 25]$  e os tempos de processamento como uma distribuição binomial com 16 tentativas e probabilidade  $p = 0.5$ . Para permitir comparação com os resultados de Barbosa (2014), a alocação de navios é realizada para dois berços.

Os parâmetros indicados na Tabela 4 (obtidos na fase de ajuste do BRKGA) foram utilizados para obtenção dos resultados desta seção. A Figura 4 apresenta uma comparação entre as melhores soluções obtidas pela metaheurística BRKGA proposta e os melhores resultados apresentados por Barbosa (2014) em seu trabalho, que foram obtidos com seu modelo de “scheduling” — a autora utilizou para execução um computador Intel Core I5 (1.80 GHz), de 4 GB de memória RAM no sistema operacional *Windows* com plataforma *AIMMS 3.12* e *solver CPLEX* e limitou a execução do modelo a aproximadamente 1 hora. Os valores do BRKGA são médias de 10 execuções.

Tabela 4: Parâmetros e Taxas da Metaheurística BRKGA

Parâmetros	BRKGA
Tamanho da população ( $n$ )	100
Tamanho do conjunto elite ( $p_e$ )	25%
Tamanho do conjunto mutante ( $p_m$ )	15%
Probabilidade de herança ao progenitor elite ( $prob_{el}$ )	70%
Reinicialização a cada $T_1$ gerações	200
Critério de Parada ( $T_2$ )	400

Os modelos de Barbosa (2014) obtiveram otimalidade apenas para os cenários de 10 navios. Na Figura 4(a), pode-se notar que o BRKGA também atingiu a otimalidade em todos os cenários dessa classe. Para as demais classes de cenários, o BRKGA obteve sempre resultados iguais ou superiores aos de Barbosa (2014). A Tabela 5 apresenta os tempos médios de processamento referentes ao BRKGA para obtenção das soluções da Figura 4.

Os tempos computacionais do BRKGA obtidos são muito pequenos. Esses resultados justificam a preferência de parte da literatura por abordagens heurísticas para o PAB. Na verdade, seria suficiente constatar que os modelos de Barbosa (2014) foram executados em até um hora e não obtiveram soluções superiores às soluções do BRKGA. Análises semelhantes, em relação ao tempo computacional, são realizados por Cordeau et al. (2005), Lalla-Ruiz et al. (2014).

Entretanto, não se deve inferir que os modelos exatos de Barbosa (2014) levaram uma hora para obter as soluções da Figura 4. Essa mesma consideração pode ser realizada para outros cenários

<sup>2</sup>Advanced Integrated Multidimensional Modeling Structure: [www.aimms.com](http://www.aimms.com)



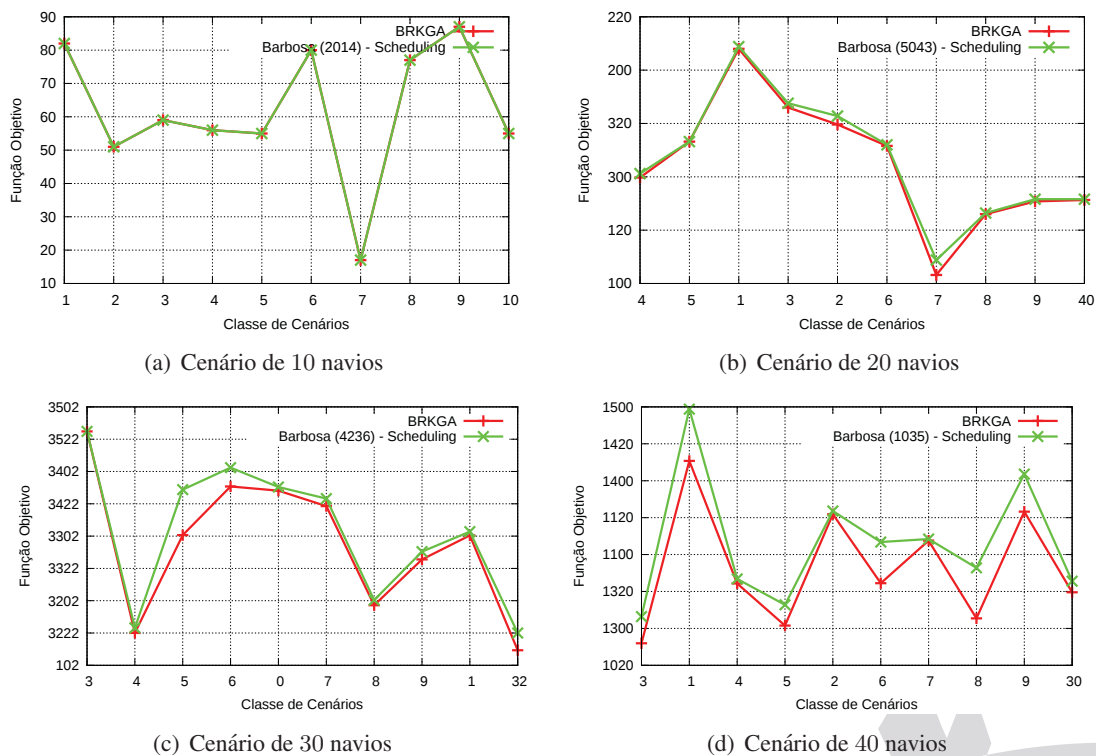

 Figura 4: Comparação de resultados: modelo de *scheduling* e o BRKGA

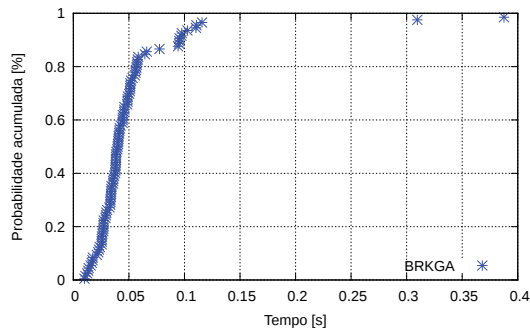
Tabela 5: Tempo computacional médio do BRKGA

Cenários	Tempos [s]			
	10 navios	20 navios	30 navios	40 navios
1	1,14	1,35	1,79	1,98
2	1,08	1,22	1,97	2,19
3	1,18	1,26	2,54	2,41
4	1,14	1,20	2,02	1,91
5	1,08	1,28	1,91	1,86
6	1,09	1,25	1,75	1,82
7	1,11	1,45	1,85	1,90
8	1,12	1,54	1,89	1,84
9	1,08	1,68	1,63	1,87
10	1,12	1,46	1,88	2,67

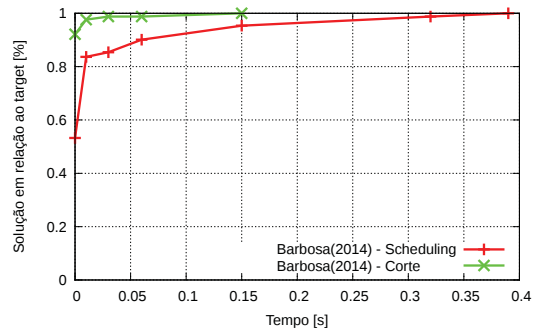
da literatura, inclusive outros problemas. A Figura 5 apresenta a comparação de desempenho do BRKGA proposto em relação aos modelos exatos de Barbosa (2014), considerando os tempos de processamento. As Figuras 5(a), 5(c), 5(e) e 5(g) apresentam curvas TTTPlots, considerando um cenário de cada classe de instância analisada (Aiex et al. 2006). Os alvos (*targets*) são as melhores soluções obtidas por Barbosa (2014) para as respectivas instâncias.

Para analisar as soluções obtidas pelos modelos exatos de Barbosa (2014), um procedimento análogo é proposto: o critério de parada do *solver* é também a obtenção da solução *target* (ou solução superior). Para isso, o modelo matemático foi implementado e a cada nova solução incumbente encontrada, o tempo necessário para sua obtenção foi registrado. Dessa forma, as demais figuras da Figura 5 apresentam as “curvas alvo” dos dois modelos de Barbosa (2014), para as

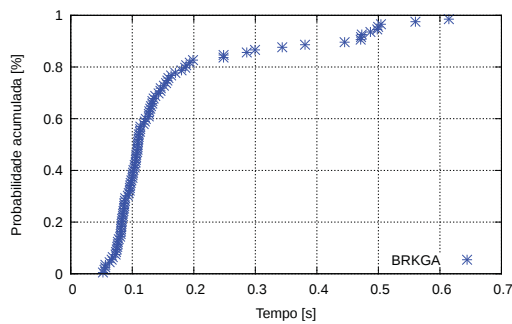
mesmas instâncias analisadas via TTTPlots.



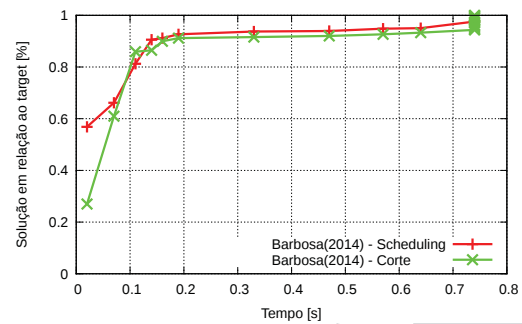
(a) BRKGA: TTTplot para 10 navios



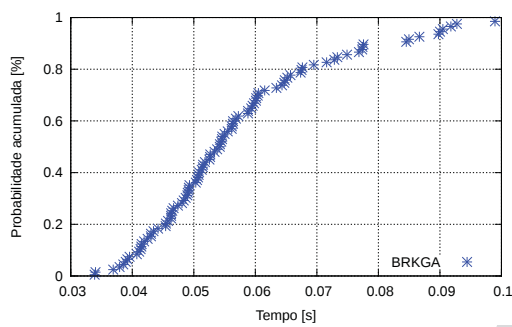
(b) PLIs: curva alvo para 10 navios



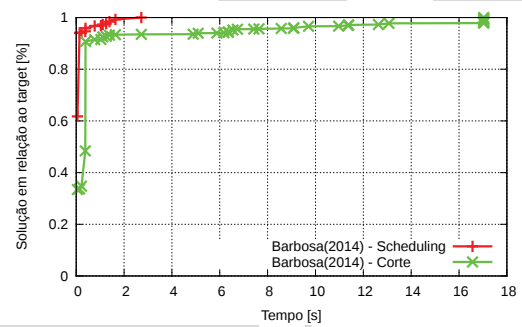
(c) BRKGA: TTTplot para 20 navios



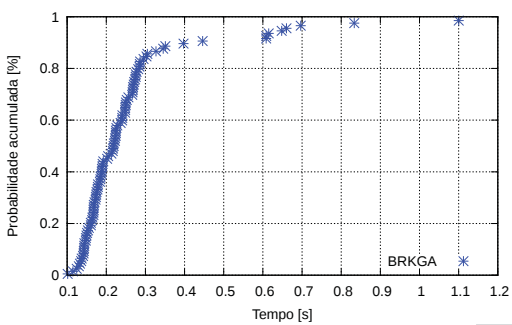
(d) PLIs: curva alvo para 20 navios



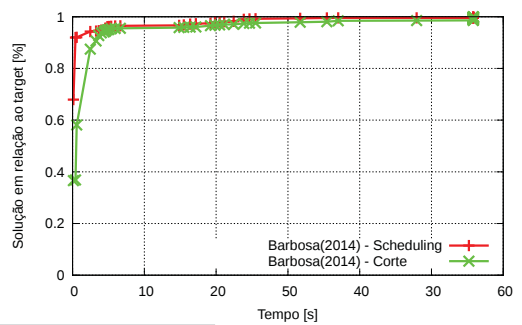
(e) PLIs: curva alvo para 30 navios



(f) Curva Target PLI para 30 navios



(g) BRKGA: TTTplot para 40 navios



(h) PLIs: curva alvo para 40 navios

Figura 5: Comparação de desempenho: BRKGA proposto e modelos matemáticos de Barbosa (2014)

É importante destacar que as curvas *time to target* são indicadas por Aiex et al. (2006) para avaliação de metaheurísticas, dado seu caráter probabilístico. Por sua vez, os modelos matemáticos foram executados uma única vez para construção das “curvas-alvo”. O que claramente pode ser notado é que os modelos exatos obtêm as soluções-alvo também na ordem de segundos, e não nos limites superiores de tempos atribuídos para as execuções.

Em outras palavras, os modelos exatos conseguem alcançar boas soluções nos primeiros segundos de execução do *solver*. Entretanto, dado a característica combinatória do problema, a árvore do *Branch & Bound* se torna muito grande e não se obtém a otimalidade. Assim, o tempo de processamento cresce na busca da solução ótima. Por consequência, o *GAP* relativo é muito alto; nas execuções realizadas esteve próximo de 90%.

Dois experimentos adicionais foram realizados. No primeiro, as melhores soluções encontradas pelo BRKGA foram passadas como solução inicial no processo de busca do modelo de *scheduling* analisado. O segundo experimento é semelhante ao primeiro, e adicionalmente a alocação navios-berços da solução do BRKGA foi travada, permitindo assim apenas a otimização da ordem de atendimento dos navios no berço. Entretanto, ainda assim a otimalidade não foi obtida nas instâncias de 20, 30 e 40 navios. No entanto, o fato de boas soluções serem obtidas nos primeiros instantes da execução indica que há ainda perspectiva para a melhoria das abordagens exatas.

## 5. Conclusão

O Problema de Alocação de Berços é a primeira decisão a ser tomada nos terminais portuários. Os berços juntamente com os guindastes são recursos valiosos. Planejamentos eficientes conduzirão a impactos positivos na operação dos portos, abrindo possibilidade de aperfeiçoamento das demais operações portuárias, considerando peculiaridades de cada porto.

Este trabalho estudou as possibilidades de aplicação de algoritmo genético baseados em chaves aleatórias viciadas para o problema de alocação de navios a berços de serviços portuários. O algoritmo heurístico foi comparado com dois modelos exatos para o problema.

Destaca-se a complexidade combinatorial do PAB. Embora cenários reais envolvam número reduzido de variáveis e restrições, quando comparados a outros problemas clássicos da literatura, a otimalidade dificilmente é obtida por métodos exatos — o que é indicado pelo alto *GAP* relativo das diversas abordagens já realizadas.

A primeira análise mostra o bom desempenho do método heurístico em relação aos modelos matemáticos de “*scheduling*” e “*cutting and packing*”, tanto na qualidade das soluções como nos tempos computacionais. A segunda análise proposta neste trabalho avalia a qualidade das soluções obtidas ao longo do processo de busca — este tipo de abordagem pode ser aplicada para qualquer metodologia matemática exata, isto é, não está limitada ao problema e modelos aqui estudados. Nos cenários analisados do PAB, a comparação evidencia que a diferença de desempenho das duas abordagens não é crítica. Portanto, indica que há espaço para aperfeiçoamento de abordagens exatas.

## 6. Agradecimentos

Os autores agradecem o apoio financeiro da CAPES e do CNPq.

## Referências

- Aiex, R. M., Resende, M. G. C. & Ribeiro, C. C. (2006), ‘Ttplots: a perl program to create time-to-target plots’, *Optimization Letters* 1(4), 355–366.
- Barbosa, F. (2014), O problema de alocação de berços: Aspectos teóricos e computacionais, Mestrado, IMECC - Universidade Estadual de Campinas, Campinas - SP - Brasil.

- Bean, J. C. (1994), 'Genetic algorithms and random keys for sequencing and combinatorial and optimization', *Journal on Computing* **6**(2), 154–180.
- Bierwirth, C. & Meisel, F. (2010), 'A survey of berth allocation and quay crane scheduling problem in container terminals', *European Journal of Operational Research* **202**(3), 615–627.
- Cordeau, J. F., Laporte, G., Legato, P. & Moccia, L. (2005), 'Models and tabu search heuristics for the berth-allocation problem', *Transportation Science* **4**(39), 526–538.
- Giallombardo, G., Moccia, L., Salani, M. & Vacca, I. (2010), 'Modeling and solving the tactical berth allocation problem', *Transportation Research Part B* **44**(2), 232–245.
- Gonçalves, J. F. & Resende, M. G. C. (2011), 'Biased random-key genetic algorithms for combinatorial optimization', *Journal of Heuristics* **17**, 487–525.
- Hansen, P. & Oğuz, C. (2003), 'A note on formulations of static and dynamic berth allocation problems', *Les Cahiers du GERAD* (30).
- Imai, A., Chen, H. C., Nishimura, E. & Papadimitriou, S. (2008), 'The simultaneous berth and quay allocation problem', *Transportation Research Part E* **44**(5), 900–920.
- Imai, A., Nishimura, E. & Papadimitriou, S. (2001), 'The dynamic berth allocation problem for a container port', *Transportation Research Part B* **35**, 401–417.
- Imai, A., Nishimura, E. & Papadimitriou, S. (2005), 'Berth allocation in a container port: using a continuous location space approach', *Transportation Research Part B* **39**(3), 199–221.
- Lalla-Ruiz, E., González-Verlarde, J. L., Melián-Batista, B. & Moreno-Vega, J. M. (2014), 'Biased random key genetic algorithm for the tactical berth allocation problem', *Applied Soft Computing* **22**, 60–76.
- Pomari, C. Z. & Chaves, A. A. (2014), Algoritmo híbrido com cs e brkga aplicado ao problema de alocação de berços, in 'Simpósio Brasileiro de Pesquisa Operacional'.
- Resende, M. G. C. (2013), Introdução aos algoritmos genéticos de chaves aleatórias viciadas, in 'Anais do SBPO'.
- SECEX (2013), 'Secretaria de comércio exterior do brasil - a importância dos portos para o comércio exterior brasileiro', <http://www.senado.gov.br/atividade/materia/getPDF.asp?t=124363&tp=1>.
- Stahlbock, R. & Voß, S. (2008), 'Operations research at container terminals: a literature update', *OR Spectrum* **30**, 1–52.
- UNCTAD (2014), 'United nations conference on trade and development - review of maritime transport', <http://www.unctad.org>.
- Vacca, I., Salani, M. & Bierlaire, M. (2013), 'An exact algorithm for the integrated planning of berth allocation and quay crane assignment', *Transportation Science* **47**(2), 148–161.