UMA METODOLOGIA DE RELAXAÇÃO COMBINATÓRIA PARA UM PROBLEMA DE LOGÍSTICA EM PORTOS

Evellyn Cavalcante, Johan Oppen, Phillippe Samer
Molde University College
P.O. box 2110, N-6402, Molde, Noruega
evellyn.cavalcante@gmail.com, johan.oppen@himolde.no,
phillippes@gmail.com

Sebastián Urrutia
Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627, 31270-010, Belo Horizonte, MG, Brasil
surrutia@dcc.ufmg.br

RESUMO

Investigamos nesse trabalho um problema de otimização em portos de contêineres, para o qual formulações generalizadas de particionamento de conjuntos foram estudadas. Apresentamos duas relaxações combinatórias baseadas no cálculo de emparelhamentos de custo máximo em grafos apropriados. Além de limites duais, tais relaxações fornecem uma técnica de redução de variáveis e uma matheurística. Resultados computationais indicam que o algoritmo é uma alternativa eficiente, alcançando o ótimo conhecido em 83% das instâncias padrão do problema. Mesmo no conjunto de instâncias mais difíceis, obtém-se soluções a menos de 4% do ótimo, com uma média de 30% do tempo necessário para métodos anteriores da literatura.

PALAVRAS CHAVE. Relaxação combinatória. Matheurísticas. Operações em portos.

Otimização Combinatória. Teoria e Algoritmos em Grafos. Programação Matemática. Logística e Transportes.

ABSTRACT

We investigate an optimization problem in container ports, for which previous models based on generalized set partitioning formulations have been studied. We describe two combinatorial relaxations based on computing maximum weighted matchings in suitable graphs. Besides dual bounds, those also provide a variable reduction technique and a matheuristic. Computational results indicate that the algorithm is an efficient alternative capable of finding the known optimal solutions on 83% of the benchmark instances. Even in the most difficult set of instances, solutions within 4% of the optimal value are provided, with an average of 30% of the time required to methods previously described in the literature.

KEYWORDS. Combinatorial relaxation. Matheuristics. Port operations.

Combinatorial optimization. Theory and algorithms on graphs. Mathematical programming. Logistics and Transportation.

1. Introdução

Nesse trabalho, investigamos um problema de otimização discreta aplicado à logística de portos marítimos. O Problema de Alocação de Berços e Atribuição de Guindastes Portuários (BACAP, do inglês *berth allocation and quay crane assignment problem)* busca determinar uma posição no espaço (um berço no cais) e no tempo, além de um número de guindastes, para atender as chegadas de navios previstas para o terminal de contêineres de um porto naval. Um planejamento viável para esse problema deve atender requisitos exigidos por cada navio quanto à sua posição no cais e ao seu horário de serviço, como também um número de guindastes portuários para atendê-lo.

O Problema de Alocação de Berços (BAP) foi introduzido por [Hoffarth e Voß, 1994] e consiste de um dos principais problemas nas operações de um terminal de contêineres [Stahlbock e Voß, 2008]. O Problema de Atribuição de Guindastes Portuários (QCAP, do inglês *quay crane assignment problem)* é geralmente resolvido de forma integrada ao BAP, por ser computacionalmente mais tratável [Bierwirth e Meisel, 2010]. O BACAP, que investigamos nesse trabalho, consiste da combinação de ambos problemas e foi estudado pela primeira vez por [Park e Kim, 2003]. Os autores propõem métodos de soluções para o problema utilizando programação inteira e uma heurística baseada em relaxação Lagrangeana. Uma crítica a esse trabalho é que, por simplicidade, a produtividade do transporte da carga de um navio é considerada proporcional ao número de guindastes atribuídos a ele, o que não é verdade na prática [Meisel e Bierwirth, 2009].

Em [Meisel e Bierwirth, 2009], o problema foi formulado com um modelo compacto que inclui os efeitos do número de guindastes na produtividade e obtém soluções ótimas para instâncias pequenas com 20 navios. Para instâncias grandes (com 30 e 40 navios), limites primais são gerados utilizando diferentes heurísticas. Em particular, o método baseado em *squeaky wheel optimization* é considerado o estado da arte dentre as heurísticas disponíveis para o BACAP. Considerando uma modelagem da aplicação e abordagem de solução diferentes, [Vacca et al., 2013] introduz uma formulação alternativa com um número exponencial de variáveis e um sofisticado algoritmo de *branch and price*.

Trabalhos mais recentes formulam variações do BACAP como uma generalização do problema de particionamento de conjuntos [Iris et al., 2015; Türkoğulları et al., 2014]. Nesse modelo, cada coluna representa uma alocação/atribuição viável para um navio e seu custo é uma combinação linear dos desvios das posições desejadas de berço e da alocação de guindastes. O trabalho de [Iris et al., 2015] resolve os modelos propostos depois de gerar todas as variáveis *a priori*. Nessa abordagem, é utilizado um conjunto de limites para a solução, e técnicas eficazes para a redução de variáveis. Os autores alcançam resultados satisfatórios para as mesmas instâncias introduzidas em [Meisel e Bierwirth, 2009]. O trabalho de [Türkoğulları et al., 2014] segue com a mesma estratégia, embora assuma modelagem da aplicação e instâncias diferentes.

Nossa contribuição nesse trabalho consiste de duas relaxações combinatórias para a formulação generalizada de particionamento de conjuntos do BACAP, utilizada por [Iris et al., 2015]. Além de mostrar que emparelhamentos em dois grafos adequados ao problema fornecem limites duais para o problema, descrevemos duas formas de empregar essas relaxações em algoritmos para a solução do problema que sigam a abordagem de enumeração de variáveis. Assim como é feito no trabalho de [Iris et al., 2015], derivamos um método de pré-processamento baseado na remoção de colunas que implicam numa solução sub-ótima. Finalmente, apresentamos uma matheurística alcançando resultados a menos de 4% do ótimo conhecido para o conjunto de instâncias mais difíceis do problema, em apenas 30%, em média, do tempo necessário pelo método de [Iris et al., 2015].

O trabalho é organizado da seguinte forma. Na seção 2, descrevemos a formulação generalizada de particionamento de conjuntos para o BACAP. Na seção 3, demonstramos duas relaxações combinatórias para o problema utilizando emparelhamentos em grafos, além de derivar o algoritmo de redução de variáveis e a matheurística. Resultados computacionais são apresentados na seção 4 e, por fim, concluímos na seção 5.

2. Formulação generalizada de particionamento de conjuntos

Por clareza, iniciamos com uma compilação de características da aplicação que assume-se nesta modelagem do BACAP.

1. O horizonte de planejamento é dividido em períodos de igual tamanho.
2. O tempo de deslocamento de guindastes portuários é desprezível, uma vez que a duração de cada período de tempo é muito maior do que tal tempo de deslocamento.
3. O cais é discretizado em seções de igual tamanho.
4. Cada seção do cais é ocupada por, no máximo, um navio em cada período de tempo.
5. Cada guindaste pode ser atribuído a, no máximo, um navio por período de tempo.
6. Cada navio tem um número mínimo e máximo de guindastes que devem ser atribuídos ao seu atendimento.
7. O atendimento de um navio pelos guindastes começa após a sua atracagem no terminal e não é interrompida até que o navio parta.
8. O número de guindastes atribuídos a um navio não muda durante a sua estadia no cais.

Existem trabalhos investigando variações dessa modelagem do problema. Em particular, questões tratadas com maior atenção na literatura incluem a atribuição de um número variável de guindastes (também por [Iris et al., 2015]), a atribuição de guindastes identificados especificamente [Türkoğulları et al., 2014], ou ainda a modelagem de berços contínuos no cais [Li et al., 2015]. O leitor interessado numa classificação cuidadosa de trabalhos relacionados pode verificar as revisões de [Stahlbock e Voß, 2008], [Bierwirth e Meisel, 2010] e [Bierwirth e Meisel, 2015].

Apresentamos a seguir a formulação para o BACAP descrita por [Iris et al., 2015], generalizando o problema de particionamento de conjuntos (GSPP, do inglês *generalized set partitioning problem*). Seja V o conjunto de navios, T o conjunto de intervalos de tempo no horizonte, L o conjunto de posições físicas no cais e K o número total de guindastes disponíveis.

Define-se ainda $P = T \times L$, e Ω como o conjunto completo de alocações (ou atribuições) viáveis: cada elemento de Ω atribui uma posição no cais, um intervalo de tempo e um número de guindastes para o atendimento de um único navio. Note que $|\Omega| \leq (|V| \times |P| \times K)$, uma vez que as alocações viáveis necessitam respeitar os requisitos acordados sobre berços e guindastes disponibilizados para cada navio.

Variáveis de decisão $y \in \mathbb{B}^{|\Omega|}$ indicam quais atribuições constituem uma solução do problema. O custo c_j de uma atribuição individual y_j, para $j \in \Omega$, consiste de uma combinação linear entre o número de horas de guindaste portuário utilizadas naquela atribuição e os desvios em relação a parâmetros esperados sobre o posicionamento no cais e horários de início e término do atendimento. Por maior generalidade na nossa contribuição, omitimos a expressão exata para o custo c_j, que varia sutilmente nos trabalhos relacionados.

Finalmente, sejam as seguintes matrizes de coeficientes, cada uma contendo $|\Omega|$ colunas. $A \in \mathbb{B}^{|V|\times|\Omega|}$ associa cada coluna a um único navio: a_{ij} é igual a 1 se a coluna j se refere a uma atribuição do navio i, 0 caso contrário. $B \in \mathbb{B}^{|P|\times|\Omega|}$ representa berços como combinações de intervalos de tempo e espaço físico: b_{pj} é igual a 1 sse o dado par (tempo, espaço) denotado $p \in P$ é utilizado no atendimento referente à atribuição y_j. Já um elemento de $Q \in \mathbb{Z}^{|T|\times|\Omega|}$ determina quantos guindastes são utilizados no período de tempo t pela atribuição y_j. Assim, a formulação para o BACAP descrita por [Iris et al., 2015] corresponde a

$$z = \min \left\{ \sum_{j\in\Omega} c_j y_j : \mathbf{y} \in \mathcal{P}_{\text{gspp}} \cap \mathbb{B}^{|\Omega|} \right\}, \tag{1}$$

em que $\mathcal{P}_{\text{gspp}}$ é a região poliédrica definida por:

$$\sum_{j \in \Omega} a_{ij} y_j = 1 \qquad \forall i \in V \tag{2}$$

$$\sum_{j \in \Omega} b_{pj} y_j \leq 1 \qquad \forall p \in P \tag{3}$$

$$\sum_{j \in \Omega} q_{tj} y_j \leq K \qquad \forall t \in T \tag{4}$$

$$y_j \leq 1 \qquad \forall j \in \Omega \tag{5}$$

$$y_j \geq 0 \qquad \forall j \in \Omega \tag{6}$$

As restrições de partição de conjuntos (2) garantem que cada navio seja contemplado em exatamente uma alocação, enquanto as desigualdades de empacotamento de conjuntos (3) evitam a sobreposição de alocações em um único intervalo de tempo e espaço físico. Já as restrições (4) garantem que a disponibilidade de guindastes portuários no terminal seja respeitada. As desigualdades (5, 6) correspondem à relaxação linear da formulação em programação binária.

A descrição de $\mathcal{P}_{\text{gspp}}$ contém um número de variáveis que cresce exponencialmente com as dimensões da entrada. Por um lado, tal situação sugere o emprego de algoritmos de geração de colunas, como [Saadaoui et al., 2015] avaliam para o subproblema de BAP. Por outro, o presente trabalho e aqueles de [Iris et al., 2015] e [Türkoğulları et al., 2014] apresentam sucesso em avaliações computacionais baseadas em uma estratégia de enumeração *a priori* das variáveis, seguidas da resolução do modelo resultante por um *solver* de programação linear inteira. Um ponto chave nessa abordagem, conforme descrevemos na seção 4, é a utilização dos métodos de redução descritos por [Iris et al., 2015] e aquele que introduzimos na seção a seguir. Mesmo as maiores instâncias padrão do BACAP utilizadas por [Iris et al., 2015] e [Meisel e Bierwirth, 2009] podem ser resolvidas com a quantidade de memória de uma estação de trabalho atual.

3. Relaxações combinatórias e algoritmos

As relaxações combinatórias para a formulação GSPP do BACAP assumem a enumeração *a priori* de atribuições/alocações viáveis para cada navio. Duas atribuições de navios diferentes são ditas *compatíveis* caso não haja sobreposição concomitante de tempo e espaço nos momentos de atracagem. De maneira equivalente, ao representar as duas atribuições em um plano Cartesiano (usando tempo e espaço como coordenadas), elas são compatíveis se os retângulos correspondentes não possuem interseção.

Iniciamos com a apresentação das relaxações e os limites duais que fornecem ao BACAP, na seção 3.1. Em seguida, estendemos tais resultados em um método de pré-processamento (seção 3.2) e uma matheurística para a solução do problema (seção 3.3).

3.1. Emparelhamentos em dois grafos interessantes

Construímos a seguir dois grafos (simples, não-direcionados) adequados ao problema, que representam um subconjunto das atribuições enumeradas. Seja $\Omega_i \subseteq \Omega$ o subconjunto de atribuições para um dado navio $i \in V$.

Seja o grafo $G_1(V, E_1)$, com um vértice para cada navio. O conjunto E_1 inclui uma aresta (i, j) se as atribuições de melhor custo para os barcos i e j não são compatíveis entre si. Denotamos por c'_j o menor custo de uma atribuição para o navio j, *i.e.* $c'_j = \min\{c(y_j) : y_j \in \Omega_j\}$. Analogamente, seja c''_j o segundo menor custo de uma atribuição para j. O custo $c_1(i, j)$ de uma aresta em G_1 é definido como a menor diferença entre esses custos, correspondentes aos navios i e j.

Isto é, $c_1(i,j) = \min\{(c''_i - c'_i), (c''_j - c'_j)\}$. Assim, o seguinte limite sobre o custo é válido para qualquer solução viável.

Teorema 1. *Seja $M \subseteq E_1$ um emparelhamento de custo máximo em G_1 e $w(M) = \sum_{e \in M} c_1(e)$ seu custo. Então $LB_1 \triangleq w(M) + \sum_{j \in V} c'_j$ é um limite inferior para o valor ótimo z em (1).*

Demonstração. Selecionar a melhor atribuição para cada navio corresponde a relaxar as restrições (3) and (4). Portanto, é um limite inferior trivial para o custo de qualquer solução viável e equivale a $\sum_{j \in V} c'_j$.

Tendo como ponto de partida a seleção trivial das melhores atribuições de cada navio, o peso de uma aresta $(i,j) \in E_1$ corresponde ao aumento mínimo no custo devido à troca de uma dessas atribuições pela segunda melhor. Claramente, o novo par de atribuições para os navios i e j pode permanecer inviável, mas a soma dos seus custos é um limite inferior para o custo de quaisquer atribuições viáveis a esses navios.

Note que não se pode implicar que os custos acumulados de arestas incidentes em um mesmo vértice sejam necessários, pois o grafo não fornece informação sobre qual dos extremos de uma aresta assume a segunda melhor atribuição; é possível até que, diante de uma troca, outras arestas sequer existam. Porém, pode-se considerar qualquer emparelhamento em G_1, que corresponde a pares disjuntos de navios cujas melhores atribuições não são compatíveis. Assim, o peso de qualquer emparelhamento é o aumento necessário sobre o custo $\sum_{j \in V} c'_j$ devido à sobreposição par-a-par das atribuições dos navios correspondentes. Em particular, o emparelhamento de custo máximo corresponde ao limite mais forte possível em G_1. □

Nosso segundo limite dual leva em consideração a informação do custo das atribuições compatíveis entre pares de navios. Seja $G_2(V, E_2)$ o grafo completo com um vértice para cada navio. Defina o peso $c_2(i,j)$ de uma aresta em E_2 como a menor soma de custos de duas atribuições compatíveis entre os navios i and j, *i.e.* $c_2(i,j) = \min\{c(y_i) + c(y_j) : y_i \in \Omega_i, y_j \in \Omega_j,\ y_i \text{ e } y_j \text{ são compatíveis}\}$. Assim, verificamos o seguinte resultado.

Teorema 2. *Seja $M \subseteq E_2$ um emparelhamento de custo máximo em G_2. Então, $LB_2 \triangleq \sum_{e \in M} c_2(e)$ é um limite inferior para o valor ótimo z em (1).*

Demonstração. O peso de uma única aresta $(i,j) \in E_2$ é a soma das melhores atribuições para os navios i e j, mantendo-se suas restrições de não sobreposição. Uma seleção de arestas em G_2 que não compartilham um vértice (*i.e* um emparelhamento) corresponde, dessa forma, a parear navios e determinar suas melhores atribuições compatíveis, exigência imposta em qualquer solução que satisfaça as restrições (3). Assim, o peso de qualquer emparelhamento em G_2 claramente relaxa as restrições de sobreposição entre navios não pareados, mas é um limite inferior para o custo de qualquer solução viável. Portanto, o emparelhamento de custo máximo provê o limite mais forte possível em G_2. □

Observe que, embora esse resultado seja válido para qualquer número de navios, seria desnecessariamente mais fraco quando $|V|$ é ímpar, uma vez que algum vértice não poderia ser pareado e não contribuiria no limite inferior. Para contornar essa situação, basta adicionar a G_2 um vértice artificial s com arestas para cada outro vértice i e custo $c_2(s,i) = \min\{c(y_i) : y_i \in \Omega_i\}$.

Finalizamos com um resultado sobre a comparação dos limites obtidos nas duas relaxações. No caso mais simples em que todas as melhores atribuições individuais são compatíveis par-a-par, verificamos: (i) o grafo G_1 não possui arestas, e o limite LB_1 corresponde ao limite trivial $\sum_{j \in V} c'_j$; (ii) todo emparelhamento perfeito M no grafo G_2 tem custo máximo, igual à soma dos custos das melhores atribuições individuais. Assim, ambos limites coincidem: $LB_2 = \sum_{(i,j) \in M} c_2(i,j) = \sum_{(i,j) \in M} (c'_i + c'_j) = \sum_{u \in V} c'_u = LB_1$. Demonstramos a seguir que o segundo limite nunca é mais fraco; casos específicos em que é mais forte são intuitivos (basta um par de vértices para os quais não existem atribuições compatíveis usando a melhor atribuição de um deles).

Teorema 3. *O limite inferior obtido através do grafo $G_2(V, E_2)$ é mais forte que aquele do grafo $G_1(V, E_1)$,* i.e. *para qualquer instância do problema, verifica-se $LB_2 \geq LB_1$.*

Demonstração. Assumimos sem perda de generalidade que o número de vértices $|V|$ é par, uma vez que, conforme sugerido acima, deve-se incluir um vértice artificial no grafo correspondente a instâncias de $|V|$ ímpar para obter o limite mais forte. Suponha que existisse alguma instância em que $LB_1 > LB_2$. Vamos construir um emparelhamento em G_2 de custo pelo menos LB_1, mostrando que a hipótese é absurda.

Para qualquer aresta $(i, j) \in E_1$ (como G_2 é completo, também temos $(i, j) \in E_2$), podemos comparar as funções de custo em G_1 e em G_2. Por definição, $c_1(i, j)$ é o mínimo acréscimo implicado pela incompatibilidade entre as melhores atribuições de i e j, e $c_2(i, j)$ é a própria soma dos custos das melhores atribuições compatíveis. Segue que:

$$c_2(i, j) \geq c'_i + c'_j + c_1(i, j) \tag{7}$$

Seja $M_1 \subseteq E_1$ um emparelhamento de custo máximo em G_1 que, portanto, fornece o limite LB_1 naquele grafo. Defina o conjunto M_2 de arestas em G_2 como $M_2 = \{(i, j) \in E_2 :$ existe a aresta $(i, j) \in M_1\}$. O conjunto M_2 é um emparelhamento em G_2, por construção.

Distinguimos dois casos. Se M_1 é perfeito, M_2 também o é pois ambos grafos têm mesmo conjunto de vértices, e podemos verificar sobre seus custos:

$$w(M_2) = \sum_{(i,j)\in M_2} c_2(i, j) \geq \sum_{(i,j)\in M_2} (c'_i + c'_j + c_1(i, j)) = \sum_{(i,j)\in M_1} c_1(i, j) + \sum_{u\in V} c'_u = LB_1, \tag{8}$$

em que a primeira desigualdade vale por (7) e a segunda igualdade vale porque os emparelhamentos são perfeitos.

Se M_1 não é perfeito, existem pares de vértices (x, y) não cobertos por M_1. Por hipótese, M_1 tem máximo custo, então $(x, y) \notin E_1$, *i.e.* as correspondentes atribuições de menor custo são compatíveis. Portanto, em G_2 o custo correspondente é mínimo: $c_2(x, y) = c'_x + c'_y$. Estendemos M_2 para um emparelhamento perfeito em G_2 conectando, de forma arbitrária, pares de vértices ainda não cobertos por M_2. Seja C o conjunto de arestas incluídas dessa forma, com $\sum_{(x,y)\in C} c_2(x, y) = \sum_{(x,y)\in C} c'_x + c'_y$. Assim, de forma análoga ao caso anterior, verificamos:

$$\begin{aligned} w(M_2) &= \sum_{(i,j)\in M_2\setminus C} c_2(i, j) + \sum_{(x,y)\in C} c_2(x, y) \geq \sum_{(i,j)\in M_2\setminus C} (c'_i + c'_j + c_1(i, j)) + \sum_{(x,y)\in C} c_2(x, y) \\ &= \sum_{(i,j)\in M_1} c_1(i, j) + \sum_{u\in V} c'_u = LB_1, \end{aligned} \tag{9}$$

em que as últimas igualdades valem porque $M_2 = M_1 \cup C$ cobre todos os vértices.

Portanto, o emparelhamento M_2 em G_2 construído em ambos casos (8) e (9) tem custo pelo menos LB_1, e define um limite inferior para LB_2, que é definido como o máximo custo de um emparelhamento em G_2. Sendo essa uma instância genérica, a hipótese de que $LB_1 > LB_2$ para alguma instância é absurda, e vale sempre que $LB_2 \geq LB_1$. □

3.2. Pré-processamento para redução de variáveis

Os resultados anteriores, por fornecerem um limite inferior para o valor ótimo z de uma instância do problema (1), podem ser estendidos a um método de pré-processamento. A proposta consiste em reduzir (ou, de forma equivalente, fixar em valor nulo) o número de variáveis do modelo resultante da enumeração, preservando a exatidão e otimalidade do resultado.

Vale destacar que, tratando-se de uma fase anterior à solução do modelo, tal proposta pode ser integrada a qualquer método baseado na enumeração de variáveis da formulação GSPP e

solução do modelo resultante por um algoritmo de programaçao linear inteira. Essa estratégia já foi adotada por [Iris et al., 2015], a partir de um limite inferior implicado apenas pela fixação de uma atribuição ou de um par de atribuições a dois navios (comparamos esse método com o que propomos nesse trabalho na seção 4).

O resultado a seguir assume disponível um limite superior qualquer para z. Adotamos nas avaliações computacionais do presente trabalho o valor da solução obtida pela heurística de *warm start* descrita por [Iris et al., 2015].

Seja uma atribuição qualquer $y_k \in \Omega_k$, que assumiremos temporariamente ser fixada na solução. Definimos o grafo completo $G_{2,k}(V \backslash \{k\}, E_{2,k})$. Os custos das arestas $c_{2,k}$ correspondem às atribuições compatíveis para dois navios que também são compatíveis com y_k. Isto é, $c_{2,k}(i,j) = \min\{c(y_i) + c(y_j) : y_i \in \Omega_i, y_j \in \Omega_j, y_i \text{ e } y_j \text{ são compatíveis entre si e com } y_k\}$. Avaliando o acréscimo no limite inferior LB_2 (Teorema 2) implicado por fixar a atribuição y_k na solução, concluímos que essa atribuição não pode fazer parte de uma solução ótima se o novo limite inferior excede um limite superior conhecido. Formalizamos o resultado como a seguir.

Proposição 1. *Seja $LB_{2,k}$ o limite inferior do Teorema (2) determinado sobre $G_{2,k}$. Dado qualquer limite superior UB para z, se $c(y_k) + LB_{2,k} > UB$, então a atribuição $y_k \in \Omega_k$ não é parte de qualquer solução ótima do problema, e a coluna/variável correspondente pode ser removida do modelo.*

Assim, dispõe-se de um algoritmo iterativo para a redução de variáveis desnecessárias no modelo, preservando toda solução ótima do problema. Basta, para cada atribuição viável da formulação GSPP, avaliar o novo limite inferior conforme a proposição anterior.

Destaca-se que um método análogo pode ser derivado do limite inferior LB_1 no Teorema 1. Entretanto, o resultado do Teorema 3 implica que ele não seria melhor, ou seja, que não removeria uma coluna que o método baseado no Teorema 2 não remova.

3.3. Matheurística de classificação de colunas

Apresentamos a seguir um algoritmo que pertence à classe de matheurísticas ou metaheurísticas baseadas em modelos, que contempla hibridações de heurísticas e métodos exatos de programação matemática [Ribeiro e Maniezzo, 2015; Maniezzo et al., 2010]. Em particular, utilizamos a informação da relaxação combinatória para obter um modelo reduzido, que resolvemos com um *solver* de programação linear inteira. Destaca-se que existem na literatura de matheurísticas outras estratégias que envolvem a resolução de um modelo reduzido: *e.g.* pela eliminação heurística de variáveis [Stefanello et al., 2015; Fanjul-Peyro e Ruiz, 2011], ou para otimizar partes componentes de uma solução anterior [Lalla-Ruiz e Voß, 2016].

Uma vez que toda solução da formulação é constituída de apenas $|V| \ll \Omega$ colunas, é de se considerar que, dispondo-se de um critério ideal para classificação das colunas, poderia-se obter soluções de qualidade consistentemente alta utilizando apenas uma fração das colunas melhor classificadas. Assim, descarta-se o certificado de otimalidade, visando obter uma solução de alta qualidade em tempo reduzido.

O núcleo do método proposto é o Algoritmo 1, que classifica e seleciona um subconjunto Ω_F das colunas do modelo de GSPP de acordo com o limite combinatório $LB_{2,k}$ apresentado na Proposição 1 (representado por Δ, computado no laço da linha 3). Otimizando sobre o conjunto Ω_F de colunas selecionadas (um subconjunto do poliedro $\mathcal{P}_{\text{gspp}}$), obtemos um limite superior $\bar{z} \geq z$ para o problema original (1). Conforme indicamos em avaliações computacionais a seguir, este limite pode coincidir com o ótimo de instâncias padrão do BACAP mesmo com uma fração relativamente pequena de colunas.

Os parâmetros do algoritmo são como a seguir.

σ : a porcentagem das melhores colunas que se deseja incluir no modelo final;

μ : o mínimo de colunas de cada barco, quando disponíveis, que se deseja garantir no modelo final.

Enquanto σ controla a seleção de colunas dentre aquelas implicando melhores limites duais (laço da linha 8), o parâmetro μ garante que cada navio disponha de mais opções viáveis (seleção no laço da linha 13). Pode-se conceber variações do algoritmo, *e.g.* selecionando um número exato de colunas, assim como o estudo estatístico dos parâmetros; ambas tarefas podem ser assunto de trabalhos futuros.

Algoritmo 1: classificação de colunas

Entrada : conjunto inicial de colunas Ω, número de navios $|V|$
Saída : conjunto Ω_F de colunas classificadas para o modelo final
Parâmetros: porcentagem mínima das melhores colunas σ, mínimo de colunas por barco μ

1 $\Omega_F \leftarrow \varnothing$
2 $\Delta(k) \leftarrow 0$ para todo $y_k \in \Omega$
3 **para cada** $y_k \in \Omega$ **faça**
4 Seja $G_{2,k}$ o grafo de atribuições compatíveis definido para a Proposição 1
5 Seja M um emparelhamento de custo máximo em $G_{2,k}$
`// limite inferior implicado pela coluna, vide Prop.1`
6 $\Delta(k) \leftarrow c(y_k) + \sum_{e \in M} c_2(e)$
7 Seja L a lista de colunas de Ω em ordem crescente de Δ
8 **enquanto** $|\Omega_F|/|\Omega| < \sigma$ **faça**
9 Seja d o menor valor de Δ de uma coluna em L
10 **para cada** *coluna* $y_k \in L$ *com* $\Delta(k) = d$ **faça**
11 $\Omega_F \leftarrow \Omega_F \cup \{y_k\}$
12 $L \leftarrow L \backslash \{y_k\}$
13 **para cada** *navio* $i = 1, \ldots, |V|$ **faça**
14 **enquanto** $|\{y \in \Omega_F : y$ *é uma atribuição de* $i\}| < \mu$ *e* L *contém alguma coluna de* i **faça**
15 Seja $y_i \in L$ uma coluna de i de menor Δ
16 $\Omega_F \leftarrow \Omega_F \cup \{y_i\}$
17 $L \leftarrow L \backslash \{y_i\}$
18 **retorna** Ω_F

Acrescenta-se que, abrindo-se mão do certificado de otimalidade, não é necessário fechar o *gap* de dualidade do modelo resultante para encerrar o algoritmo. Tal estratégia pode ser interessante para contemplar limites no tempo de execução em uma aplicação do BACAP.

4. Resultados computacionais

Para avaliar a eficiência dos algoritmos propostos, consideramos o mesmo conjunto de instâncias utilizado nos trabalhos de [Meisel e Bierwirth, 2009] e [Iris et al., 2015], com 10 instâncias de cada dimensão: pequenas (20 navios), médias (30 navios) e grandes (40 navios).

Os algoritmos foram implementados em linguagem C++, utilizando o *solver* de programação linear inteira GUROBI versão 6.5. Para a determinação de emparelhamentos de custo máximo, utilizamos a implementação de algoritmo de *blossom shrinking* de Edmonds [Edmonds, 1965] disponível na biblioteca de código livre LEMON [Dezső et al., 2011], com complexidade de tempo no pior caso em $O(mn \log(n))$, em que n denota o número de vértices e m o número de arestas do grafo. O tempo de execução do *solver* nos experimentos envolvendo a matheurística foi limitado a 1800 segundos. Para os experimentos utilizando as técnicas propostas por [Iris et al., 2015] não foi imposto limite algum de tempo. Vale destacar que os resultados que apresentamos sobre o trabalho de [Iris et al., 2015] foram avaliados com uma implementação própria e os números envolvem alguma variação sutil em relação ao trabalho original (por questão de precisão numérica, concluiu-se). Todos os experimentos foram executados em uma máquina com processador Intel Core i7 4790K (4,00 GHz) e 16GB de RAM.

A tabela 1 mostra a comparação entre o desempenho das técnicas de redução de variáveis propostas nesse trabalho com as presentes em [Iris et al., 2015]. O número de colunas resultan-

tes após cada fase de redução é apresentado. $|\Omega|$ indica a quantidade original de colunas após a enumeração. $|\Omega_1|$ é o valor de colunas resultantes após dois pré-processamentos propostos por [Iris et al., 2015], que se baseiam em testes simples de exclusão de colunas a partir de comparações com o limite superior: seja o próprio custo de cada coluna, sejam implicações de fixá-la e assumir as melhores atribuições a outros navios (ignorando condições de viabilidade).

Tabela 1: Resultados dos pré-processamentos e redução de variáveis

Instância			Pré-processamento e Redução de variáveis				
			Iris et al. [2015]			Red. Combin.	
$\|V\|$	ID	$\|\Omega\|$	$\|\Omega_1\|$	$\|\Omega_2\|$	$R_I(\%)$	$\|\Omega_3\|$	$R_C(\%)$
20	1	316278	102657	60114	80,99	54582	**82,74**
	2	338438	19202	1349	99,60	1345	99,60
	3	252618	94083	77457	69,34	60941	**75,88**
	4	390624	106970	74730	80,87	73199	**81,26**
	5	275238	33857	23689	91,39	18852	**93,15**
	6	305485	28133	11352	96,28	11318	**96,30**
	7	330639	58616	24388	92,62	23963	**92,75**
	8	359703	61200	41789	88,38	40448	**88,76**
	9	286683	91561	65926	77,00	64592	**77,47**
	10	359320	132769	112444	68,71	83589	**76,74**
				$\overline{R}_{I_{20}}$	84,52	$\overline{R}_{C_{20}}$	**86,47**
30	11	496340	250999	199771	59,75	192806	**61,15**
	12	523233	100586	63672	87,83	47430	**90,94**
	13	483010	122717	93705	80,60	74715	**84,53**
	14	467661	164949	109924	76,49	109154	**76,66**
	15	477733	283098	244890	48,74	228111	**52,25**
	16	532026	182560	71015	86,65	69745	**86,89**
	17	467085	143231	94442	79,78	84927	**81,82**
	18	508500	252145	220062	56,72	212453	**58,22**
	19	506300	312026	270994	46,48	264999	**47,66**
	20	487945	238218	158871	67,44	156776	**67,87**
				$\overline{R}_{I_{30}}$	69,05	$\overline{R}_{C_{30}}$	**70,80**
40	21	768104	614745	571876	25,55	569520	**25,85**
	22	611023	403976	290449	52,47	264991	**56,63**
	23	720952	616879	577758	19,86	572515	**20,59**
	24	657415	634437	591415	10,04	584133	**11,15**
	25	535300	358648	300801	43,81	279464	**47,79**
	26	675819	596037	555547	17,80	545320	**19,31**
	27	615092	474742	435041	29,27	405074	**34,14**
	28	698783	631227	612466	12,35	590405	**15,51**
	29	673090	501858	415807	38,22	411307	**38,89**
	30	678072	523970	457119	32,59	450821	**33,51**
				$\overline{R}_{I_{40}}$	28,20	$\overline{R}_{C_{40}}$	**30,34**

$R_I = \dfrac{\Omega - \Omega_2}{\Omega}$; $R_C = \dfrac{\Omega - \Omega_3}{\Omega}$; $\overline{R}_{I_{|V|}}$ corresponde à média aritmética de R_I das instâncias com $|V|$ navios; $\overline{R}_{C_{|V|}}$ corresponde à média aritmética de R_C das instâncias com $|V|$ navios.

O valor $|\Omega_2|$ consiste no total de colunas após a aplicação das duas técnicas de redução de variáveis também proposta por [Iris et al., 2015], iniciando do conjunto em Ω_1. Finalmente, $|\Omega_3|$ representa o número final de colunas após a aplicação sucessiva: das duas técnicas de processamento utilizadas em Ω_1, a primeira técnica de redução de variáveis (mais simples) de [Iris et al., 2015] e, por fim, a técnica combinatória de redução de variáveis proposta nesse trabalho.

Na tabela também são apresentadas as porcentagens finais, R_I e R_C, da redução de variáveis considerando $|\Omega_2|$ e $|\Omega_3|$, respectivamente. Em média, a diminuição é maior nas instâncias pequenas e médias quando comparadas às grandes, chegando a reduzir o número de colunas em 99,60% na instância 2. Verifica-se que a redução é ligeiramente maior após a aplicação da técnica que utiliza a relaxação combinatória, vide resultados em negrito, empatando somente na instância 2. Outro fato a destacar é que a técnica de redução combinatória, em comparação com as técnicas pro-

postas por [Iris et al., 2015], foi mais expressiva, em média, nas instâncias grandes (aumento de 2,14 pontos percentuais) do que nas pequenas (aumento de 1,95 pontos percentuais) ou médias (aumento de 1,75 pontos percentuais).

Para avaliar o algoritmo de matheurística, combinações de parâmetros (σ, μ) foram experimentadas: $\sigma \in \{10, 20, 25, 33\}$ e $\mu \in \{1000, 2000\}$. Os melhores resultados foram alcançados com a configuração $(10, 2000)$ e são reportados na tabela 2. Para cada instância, apresentamos o valor z da solução ótima, a porcentagem de colunas resultantes após a filtragem da matheurística (Ω_F), o custo ($\bar{z}$) e o GAP_{OPT} total (em relação ao valor ótimo z) da solução encontrada pela matheurística. Sobre tempo computacional, mostramos o tempo de execução parcial considerando apenas a aplicação das técnicas de redução de variáveis da nossa proposta (T_C) e de [Iris et al., 2015] (T_I), bem como o total adicionando o tempo de execução para resolver o modelo com o *solver*, T_{C_F} e T_{I_F}. Por fim, mostramos a eficiência percentual do tempo de execução (T_E) da nossa proposta em relação à [Iris et al., 2015].

Tabela 2: Resultados sobre qualidade das soluções e tempo de execução

Instância			Resultados com matheurística			Tempos computacionais				Eficiência
						Matheurística		Iris et al. [2015]		
$\|V\|$	ID	z	$\Omega_F(\%)$	$\bar{z}$	$\text{Gap}_{\text{OPT}}(\%)$	T_C (s)	T_{C_F} (s)	T_I (s)	T_{I_F} (s)	$T_E(\%)$
20	1	89,00	60,84	89,00	0,00	32,44	**56,65**	14,84	64,28	88,13
	2	56,20	100,00	56,20	0,00	1,01	1,09	0,09	**0,16**	681,25
	3	85,70	59,09	85,70	0,00	9,63	**73,19**	0,79	114,73	63,79
	4	81,80	47,86	81,80	0,00	8,58	**39,24**	2,22	60,22	65,16
	5	59,20	99,13	59,20	0,00	2,62	7,99	0,16	**7,59**	105,27
	6	59,20	100,00	59,20	0,00	1,53	6,73	0,16	**5,09**	132,22
	7	75,20	94,32	75,20	0,00	6,42	21,33	2,56	**19,92**	107,08
	8	61,40	73,20	61,40	0,00	5,39	22,10	0,95	**21,41**	103,22
	9	79,00	52,22	79,00	0,00	16,37	**45,45**	3,77	54,32	83,67
	10	101,00	45,56	101,00	0,00	26,05	**76,27**	3,13	214,43	35,57
									$\overline{T}_{E_{20}}$	101,34
30	11	143,20	31,43	143,20	0,00	60,66	**186,16**	7,64	421,91	44,12
	12	92,00	87,55	92,00	0,00	13,59	43,93	0,85	**43,67**	100,60
	13	110,00	66,45	110,00	0,00	24,53	**64,00**	3,01	85,90	74,51
	14	107,40	50,85	107,40	0,00	28,50	**111,48**	6,73	164,24	67,88
	15	168,40	26,53	168,40	0,00	70,46	**236,08**	10,64	665,77	35,46
	16	121,60	72,09	121,60	0,00	57,76	**144,80**	22,03	153,33	94,44
	17	109,40	61,39	109,40	0,00	28,97	**73,98**	4,30	85,56	86,47
	18	135,00	28,24	135,00	0,00	56,29	**253,90**	10,72	651,13	38,99
	19	176,20	23,18	176,20	0,00	86,51	**252,40**	27,41	737,74	34,21
	20	139,80	38,27	139,80	0,00	70,84	**204,72**	28,60	346,50	59,08
									$\overline{T}_{E_{30}}$	59,01
40	21	246,80	16,20	252,60	2,30	235,29	**2035,52**	20,90	9237,00	22,04
	22	178,40	30,14	178,40	0,00	171,43	**449,33**	45,42	694,91	64,66
	23	266,30	15,39	266,30	0,00	322,13	**1771,94**	131,24	14516,88	12,21
	24	307,00	15,39	318,20	3,52	1788,03	**3588,11**	538,36	11071,74	32,41
	25	164,60	28,63	164,60	0,00	105,97	**457,75**	8,21	1049,86	43,60
	26	258,20	16,31	261,90	1,41	274,46	**2074,55**	89,55	4337,52	47,83
	27	205,40	19,92	205,40	0,00	171,92	**454,28**	17,81	1520,09	29,89
	28	294,20	16,04	301,30	2,36	518,37	**2318,46**	66,65	42988,26	5,39
	29	227,60	19,96	229,70	0,91	680,40	**2480,52**	567,79	2563,04	96,78
	30	210,10	18,57	210,10	0,00	223,22	**1470,65**	46,59	4354,89	33,77
									$\overline{T}_{E_{40}}$	30,14

$\Omega_F = \dfrac{\Omega_M}{\Omega_3}$; Ω_M é o número final de variáveis após a filtragem realizada pela matheurística; $T_E = \dfrac{T_{C_F}}{T_{I_F}}$; $\overline{T}_{E_{|V|}}$ corresponde à média geométrica de T_E das instâncias com $|V|$ navios.

Verifica-se a partir dos resultados que a metodologia proposta alcança a solução ótima em 83% das instâncias, falhando nas mais difíceis e com um GAP inferior à 4% do valor ótimo. Com relação ao tempo de processamento, a eficiência da metodologia é inferior à da literatura em metade das instâncias com 20 navios e em apenas uma com 30 navios. No entanto, considera-se

que o impacto desse resultado não é significativo pois os tempos em questão não ultrapassam 44s, e as diferenças entre os métodos não ultrapassam 2s. Por exemplo, destaca-se a instância 2, que é resolvida instantaneamente pelo método da literatura, e gasta 1,09s com a matheurística, o que faz nossa proposta perder em eficiência média nas instâncias fáceis.

Por outro lado, ao observar os resultados para as instâncias médias e grandes, percebe-se que a matheurística obtém resultados significativamente superiores ao da literatura. A solução ótima é encontrada em todas as instâncias com 30 navios gastando em média 59% do tempo gasto pelo método da literatura. Já nas instâncias mais difíceis, com 40 navios, apesar da solução ótima não ser obtida em metade dos casos, encontra-se boas soluções utilizando apenas 30% do tempo gasto pelo método de [Iris et al., 2015].

Na instância 23, por exemplo, a solução ótima é obtida 8 vezes mais rapidamente. Já na instância 28, a matheurística utiliza menos de 6% do tempo do algoritmo da literatura para encontrar uma solução com um GAP menor que 3% do valor ótimo. Esses resultados sugerem que a classificação pelo limite combinatório da matheurística é um bom critério de seleção: o tamanho do modelo diminui drasticamente, o tempo de solução diminui de forma consistente em instâncias de dimensões mais desafiadoras, enquanto conserva-se soluções a 4% do ótimo.

5. Conclusões

Apresentamos nesse trabalho um conjunto de resultados sobre a solução de um problema de logística de portos navais, formulado em programação linear inteira generalizando o problema de particionamento de conjuntos. O problema consiste em obter uma alocação individual de cada navio programado para chegar em um terminal de contêineres, associando a eles um espaço físico, um período de tempo e um número de guindastes portuários para seu atendimento.

Partindo de uma relaxação combinatória, obtida com informações de emparelhamentos em grafos adequados ao problema, descreve-se técnicas de redução de variáveis e uma matheurística baseada na classificação de alocações com relação ao limite combinatório, e sucessiva solução de um modelo reduzido. Avaliações computacionais da metodologia proposta e comparações com métodos descritos como estado da arte indicam sua eficiência em instâncias padrão do problema.

Referências

Bierwirth, C. e Meisel, F. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202(3):615 – 627. ISSN 0377-2217. URL `http://www.sciencedirect.com/science/article/pii/S0377221709003579`.

Bierwirth, C. e Meisel, F. (2015). A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 244(3):675 – 689. ISSN 0377-2217. URL `http://www.sciencedirect.com/science/article/pii/S0377221714010480`.

Dezső, B., Jüttner, A., e Kovács, P. (2011). LEMON – an Open Source C++ Graph Template Library. *Electronic Notes in Theoretical Computer Science*, 264(5):23 – 45. ISSN 1571-0661.

Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467. URL `http://cms.math.ca/10.4153/CJM-1965-045-4`.

Fanjul-Peyro, L. e Ruiz, R. (2011). Size-reduction heuristics for the unrelated parallel machines scheduling problem. *Computers & Operations Research*, 38(1):301 – 309. ISSN 0305-0548. URL `http://www.sciencedirect.com/science/article/pii/S0305054810001188`. Project Management and Scheduling.

Hoffarth, L. e Voß, S. (1994). *Operations Research Proceedings 1993*, chapter Berth allocation in a container terminal — Development of a decision support system (in German), p. 89–95.

Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-642-78910-6. URL http://dx.doi.org/10.1007/978-3-642-78910-6_28.

Iris, Ç., Pacino, D., Ropke, S., e Larsen, A. (2015). Integrated berth allocation and quay crane assignment problem: Set partitioning models and computational results. *Transportation Research Part E: Logistics and Transportation Review*, 81:75 – 97. ISSN 1366-5545. URL http://www.sciencedirect.com/science/article/pii/S1366554515001301.

Lalla-Ruiz, E. e Voß, S. (2016). Popmusic as a matheuristic for the berth allocation problem. *Annals of Mathematics and Artificial Intelligence*, 76(1):173–189. ISSN 1573-7470. URL http://dx.doi.org/10.1007/s10472-014-9444-4.

Li, F., Sheu, J.-B., e Gao, Z.-Y. (2015). Solving the continuous berth allocation and specific quay crane assignment problems with quay crane coverage range. *Transportation Science*, 49(4):968–989. URL http://dx.doi.org/10.1287/trsc.2015.0619.

Maniezzo, V., Stützle, T., e Voß, S. (2010). *Matheuristics: hybridizing metaheuristics and mathematical programming*, volume 10 of *Annals of Information Systems*. Springer US. ISBN ISBN: 978-1-4419-1305-0.

Meisel, F. e Bierwirth, C. (2009). Heuristics for the integration of crane productivity in the berth allocation problem. *Transportation Research Part E: Logistics and Transportation Review*, 45 (1):196 – 209. ISSN 1366-5545. URL http://www.sciencedirect.com/science/article/pii/S1366554508000768.

Park, Y.-M. e Kim, H. K. (2003). A scheduling method for berth and quay cranes. *OR Spectrum*, 25(1):1–23. ISSN 1436-6304. URL http://dx.doi.org/10.1007/s00291-002-0109-z.

Ribeiro, C. C. e Maniezzo, V. (2015). Preface to the special issue on matheuristics: Model-based metaheuristics. *International Transactions in Operational Research*, 22(1):1–1. ISSN 1475-3995. URL http://dx.doi.org/10.1111/itor.12142.

Saadaoui, Y., Umang, N., e Frejinger, E. (2015). A column generation framework for berth scheduling at port terminals. Technical Report CIRRELT-2015-15, Université de Montréal. URL http://www.cirrelt.ca/DocumentsTravail/CIRRELT-2015-15.pdf.

Stahlbock, R. e Voß, S. (2008). Operations research at container terminals: A literature update. *OR Spectrum*, 30(1):1–52. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-36248971841&partnerID=40&md5=4588095e77c11800fb778c47e775e252.

Stefanello, F., de Araújo, O. C. B., e Müller, F. M. (2015). Matheuristics for the capacitated p-median problem. *International Transactions in Operational Research*, 22(1):149–167. ISSN 1475-3995. URL http://dx.doi.org/10.1111/itor.12103.

Türkoğulları, Y., Taşkın, Z. C., Aras, N., e Altınel, I. K. (2014). Optimal berth allocation and time-invariant quay crane assignment in container terminals. *European Journal of Operational Research*, 235(1):88 – 101. ISSN 0377-2217. URL http://www.sciencedirect.com/science/article/pii/S0377221713008382.

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. URL http://pubsonline.informs.org/doi/abs/10.1287/trsc.1120.0428.