UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS AGRÁRIAS
DEPARTAMENTO DE COMPUTAÇÃO

VINICIUS MIRANDA LOPES SCHULZ

ADAPTIVE LARGE NEIGHBORHOOD SEARCH PARA RESOLUÇÃO DO PROBLEMA DE ESTIVAGEM DE UNIDADES DE CELULOSE

ALEGRE - ES
2013

VINICIUS MIRANDA LOPES SCHULZ

ADAPTIVE LARGE NEIGHBORHOOD SEARCH PARA RESOLUÇÃO DO PROBLEMA DE ESTIVAGEM DE UNIDADES DE CELULOSE

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Agrárias da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.
Orientador: Prof. Dr. Geraldo Regis Mauri.

ALEGRE - ES
2013

VINICIUS MIRANDA LOPES SCHULZ

ADAPTIVE LARGE NEIGHBORHOOD SEARCH PARA RESOLUÇÃO DO PROBLEMA DE ESTIVAGEM DE UNIDADES DE CELULOSE

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Agrárias da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 03 de setembro de 2013.

COMISSÃO EXAMIDADORA

Prof. Dr. Geraldo Regis Mauri
Universidade Federal do Espírito Santo
Orientador

Prof. M.Sc. Antonio Almeida de Barros Junior
Universidade Federal do Espírito Santo

Prof. M.Sc. Edmar Hell Kampke
Universidade Federal do Espírito Santo

À minha mãe e meus avós que sempre estiveram ao meu lado e fizeram de mim o homem que sou.

AGRADECIMENTOS

Primeiramente a Deus, pois sei que sem Ele eu não teria trilhado este caminho até aqui.

À minha mãe Rosangela e meus avós Jayme e Augusta, a quem devo tudo nesta vida e tenho imensa gratidão por tudo que eles representam em minha vida.

À Bárbara, a razão da minha vida e a pessoa que eu quero estar ao lado até o fim dos meus dias.

Às minhas tias Thalyta, por ser minha segunda mãe, Rosana, minha conselheira para todos os momentos, Rose, pelo imenso amor e carinho que sempre teve por mim e tios Jés, Jaime e Robson, por sempre estarem ao meu lado e fazerem parte da minha história.

Aos meus amigos Rômulo, Pablo e André que sempre estiveram ao meu lado durante todo o curso.

Aos meus amigos Ruan, Capote, Potin, Melocoton, Japa e Dikson, que estiveram presentes nos momentos mais felizes que passei aqui em Alegre e quem quero que estejam ao meu lado pro resto da vida.

Aos meus amigos de república; Paulo Vitor, Vinícius Paris e Laédio, por terem me acolhido como irmão.

Ao meu orientador Geraldo, por ter acreditado em meu potencial no começo do curso e por ter me dado o privilégio de ser seu orientado.

Aos meus professores da graduação, que me passaram seus conhecimentos sem nenhuma reserva e me fizeram crer que sou capaz de conquistar o mundo.

Aos meus amigos de turma.

Aos meus primos, primas, amigos e amigas de todos os lugares que já passei.

E a todos que, de alguma forma, participaram da minha vida.

RESUMO

O processo de estivagem de unidades de celulose consiste em organizar estes itens em porões de navios de modo a maximizar a quantidade de unidades estivadas. Devido aos vários tipos de navios e cargas, o processo de carregamento dos porões torna-se muito difícil, podendo levar até dias para ser concluído. Neste trabalho, é proposta a utilização da meta-heurística *Adaptive Large Neighborhood Search* (ALNS), que permite o uso de vários métodos de destruição e reconstrução de soluções dentro de um mesmo processo de busca de modo a maximizar a quantidade de carga estivada.

Palavras-chave: Problema de Estivagem de Unidades de Celulose; *Adaptive Large Neighborhood Search*; Meta-heurísticas.

ADAPTIVE LARGE NEIGHBORHOOD SEARCH TO SOLVE THE WOODPULP STOWAGE PROBLEM

ABSTRACT

The woodpulp stowage process consists of organizing woodpulp units in ship's holds maximizing the number of stowed units. Due to the various types of ships and cargo, the charging process becomes very difficult and can take up to days to be completed. In this work, we propose the use of the Adaptive Large Neighborhood Search (ALNS) meta-heuristic to solve the woodpulp stowage problem. ALNS allows to use several methods for destruction and reconstruction of solutions within the same searching process trying to maximize the amount of stowed cargo.

Keywords: Woodpulp Stowage Problem; Adaptive Large Neighborhood Search; Metaheuristics.

LISTA DE FIGURAS

Figura 1 - Estivagem de unidades de celulose em porões de navios.....13

Figura 2 - Possíveis orientações de uma unidade de celulose.....14

Figura 3 - Vértices de um espaço de busca. 19

Figura 4 - Solução viável. 19

Figura 5 - Solução inviável. 20

Figura 6 - Representação de uma solução. 20

Figura 7 - Heurística de Remoção de Shaw. 24

Figura 8 - Heurística de Remoção do Pior. 25

Figura 9 - Heurística de Remoção Aleatória.....25

Figura 10 - Heurística de Inserção Gulosa. 26

Figura 11 - Pseudocódigo da ALNS implementada.....28

LISTA DE TABELAS

Tabela 1 - Instâncias utilizadas. 29

Tabela 2 - Resultados obtidos pela ALNS. 30

Tabela 3 - Comparação do desempenho da ALNS com outros métodos encontrados na literatura. 31

LISTA DE ABREVIATURAS E SIGLAS

3D-BPP - *Three-Dimensional Bin Packing Problem*

ALNS - *Adaptive Large Neighborhood Search*

HIG - Heurística de Inserção Gulosa

HRA - Heurística de Remoção Aleatória

HRP - Heurística de Remoção do Pior

HRS - Heurística de Remoção de Shaw

LNS - *Large Neighborhood Search*

PCPP - Problema de Carregamento de Paletes do Produtor

PEUC - Problema de Estivagem de Unidades de Celulose

SA - *Simulated Annealing*

SUMÁRIO

1 INTRODUÇÃO 12

1.1 O PROBLEMA E SUA IMPORTÂNCIA 13

1.2 OBJETIVOS 16

1.2.1 Objetivo geral 16

1.2.2 Objetivos específicos 16

2 REVISÃO DE LITERATURA 17

3 METODOLOGIA 19

3.1 REPRESENTAÇÃO DE UMA SOLUÇÃO 19

3.2 ADAPTIVE LARGE NEIGHBORHOOD SEARCH 20

3.3 HEURÍSTICAS DE DESTRUIÇÃO (H-) E RECONSTRUÇÃO (H+) 23

3.3.1 Heurística de Remoção de Shaw (HRS) baseada na quantidade de sobreposições: 23

3.3.2 Heurística de Remoção do Pior (HRP): 24

3.3.3 Heurística de Remoção Aleatória (HRA): 25

3.3.4 Heurística de Inserção Gulosa (HIG): 25

3.3.5 Meta-heurística *Adaptive Large Neighborhood Search* (ALNS): 26

4 RESULTADOS 29

5 CONCLUSÕES 33

REFERÊNCIAS BIBLIOGRÁFICAS 34

1 INTRODUÇÃO

O transporte marítimo é fundamental para o crescimento econômico mundial, sendo essencial para impulsionar a economia e a competição entre as empresas. Atualmente, cerca de 80% do comércio mundial é feito por via marítima e, até 2018, o número de navios deverá aumentar cerca de 29% (CCE, 2009).

Além deste crescimento do setor marítimo, segundo dados do Relatório Estatístico 2010/2011 da Associação Brasileira de Celulose e Papel – BRACELPA (http://www.bracelpa.org.br), o Brasil teve um aumento de 47,3%, em US$ milhões FOB (*Free On Board* (ICC, 2010) – expressão inglesa que quer dizer que o exportador é responsável pela mercadoria até ela estar dentro do navio para transporte e no porto indicado pelo comprador), nas exportações de celulose entre os anos de 2009 e 2010 e um aumento de 48,8% nas importações deste insumo neste mesmo período.

Esse crescimento do comércio internacional de mercadorias incentivou a procura de serviços de transporte marítimo e, com isso, devido aos mais variados tipos de navios e cargas, posicionar as cargas nos porões, muitas vezes, torna-se uma tarefa difícil de ser resolvida, podendo levar vários dias para ser concluída.

Por isso, otimizar o planejamento da estivagem, das cargas é de suma importância para agilizar o processo de carregamento, reduzindo o tempo desta operação e os custos de logística associados. De acordo com Handabaka (1994), esse plano deve avaliar o tipo e a natureza das cargas, o navio onde serão estivadas, dentre outros fatores, os quais não serão tratados neste trabalho.

Essa necessidade em otimizar o processo de estivagem motivou o surgimento de um problema conhecido na literatura como Problema de Estivagem de Unidades de Celulose (PEUC), que tem por objetivo maximizar a quantidade de unidades de celulose nos porões de navios.

Para resolver o PEUC, este trabalho propõe a aplicação da meta-heurística *Adaptive Large Neighborhood Search* (ALNS), proposta por Ropke e Pisinger (2006), para construção de soluções para o problema.

Então, para validação da abordagem proposta, foi utilizado um conjunto de instâncias baseadas em dados reais obtidos em portos brasileiros para o PEUC.

1.1 O PROBLEMA E SUA IMPORTÂNCIA

De acordo com o trabalho de Ribeiro e Lorena (2005), o Problema de Estivagem de Unidades de Celulose (PEUC) é uma importante tarefa no processo de exportação e pode gerar muitos custos para o porto, caso o navio atrase por causa do carregamento. Por outro lado, uma boa estivagem pode reduzir o tempo de carregamento e os custos logísticos associados.

Deste modo, este trabalho trata especificamente o PEUC e, segundo Ribeiro e Lorena (2005), as unidades de celulose são compostas por fardos, que por sua vez são constituídos por placas de celulose amarradas por arame. Essas unidades a serem estivadas apresentam dimensões variadas e são posicionadas nos porões dos navios.

A partir dessas informações, nota-se que o PEUC resume-se basicamente em maximizar a quantidade de unidades de celulose estivadas por meio do planejamento prévio da localização e orientação de cada unidade e, com isso, obter um melhor aproveitamento do espaço disponível no porão do navio, reduzir o tempo de embarque e maximizar os lucros econômicos da empresa produtora de celulose.

A Figura 1 apresenta algumas fotos de unidades de celulose sendo estivadas em porões de navios.

Figura 1 - Estivagem de unidades de celulose em porões de navios.
Fonte - Ribeiro (2007).

O PEUC pode ser visto como um caso particular de outro problema conhecido como *Three-Dimensional Bin Packing Problem* (3D-BPP). O 3D-BPP consiste em empacotar todos os *n* itens no menor número possível de *bins* (caixas), onde cada item é caracterizado pela largura *w*, altura *h* e comprimento *d* e todos os *bins* com iguais dimensões: largura *W*, altura *H* e comprimento *D* (SILVA et al., 2001).

Devido às restrições impostas ao transporte, é possível tratar o PEUC como sendo bidimensional, uma vez que somente a primeira camada é definida e as outras são repetições desta até que seja atingida uma altura satisfatória de camadas (RIBEIRO; LORENA, 2005). Pela tipologia proposta por Dyckhoff (1990), o PEUC pode ser classificado como o Problema de Carregamento de Paletes do Produtor (PCPP), que consiste em arranjar o maior número possível de caixas iguais sobre um palete.

Com base nestas informações, o PEUC é tratado neste trabalho como um caso particular do problema de corte bidimensional, proposto por Beasley (1985), que consiste em cortar um número de peças retangulares de um retângulo maior, de modo a maximizar o valor das peças cortadas. Analogamente, dado que o porão de um navio possui comprimento *L* e largura *W*, este seria o retângulo maior, segundo Beasley (1985), e as unidades de celulose com dimensões *l* e *w* representariam as peças possíveis obtidas a partir do seccionamento dos espaços do porão do navio, com a ressalva de que estas unidades de celulose possuem apenas duas orientações, horizontal e vertical, ilustradas na Figura 2, de tal modo que $L \geq W$ e, que *l* e *w* representam o comprimento e a largura das unidades de celulose, respectivamente, tal que $l \geq w$ e $l \leq W$.

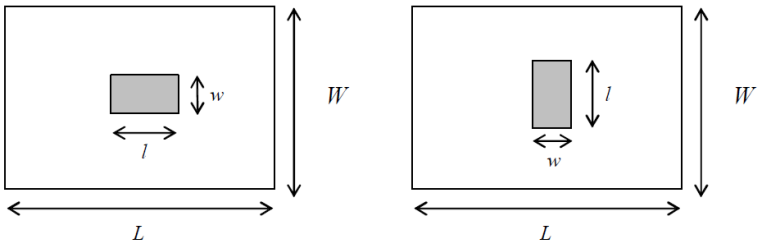

Figura 2 - Possíveis orientações de uma unidade de celulose.

Fonte - Ribeiro e Lorena (2005).

Conjuntos *X* e *Y* podem ser utilizados para representar as coordenadas (*p*,*q*) do canto inferior esquerdo das unidades de celulose, medidas a partir do canto inferior

esquerdo do porão do navio. Esses conjuntos podem ser descritos da seguinte maneira:

$$X = \left\{ p \in Z^{+} \middle| p = \sum_{i=1}^{2} l_i b_i, 0 \leq p \leq L - w, b_i \geq 0 \; e \; inteiro, i = 1,2 \right\} \quad (1)$$

$$Y = \left\{ q \in Z^{+} \middle| q = \sum_{i=1}^{2} w_i b_i, 0 \leq q \leq W - w, b_i \geq 0 \; e \; inteiro, i = 1,2 \right\} \quad (2)$$

Seja *a* uma função que descreve as restrições de sobreposição de unidades no porão do navio, que pode ser obtida com antecedência para cada vértice (*p*,*q*) em relação a cada posição (*r*,*s*), para cada orientação *i*. Sendo $p \in X | p \leq L - l_i,\ q \in Y | q \leq W - w_i,\ r \in X,\ s \in Y,$ e $i = 1,2$. Desse modo, essa função pode ser expressa como:

$$a_{ipqrs} = \begin{cases} 1, \text{se } 0 \leq p \leq r \leq p + l_i - 1 \leq L - 1 \text{ e } 0 \leq q \leq s \leq q + w_i - 1 \leq W - 1 \\ 0, \text{ caso contrário} \end{cases} \quad (3)$$

Seja $x_{ipq} \in \{0,1\}$ uma variável binária de decisão para todo $p \in X | p \leq L - l_i, q \in Y | q \leq W - w_i, e\ i = 1,2$. Se $x_{ipq} = 1$, então uma unidade de celulose é colocada nas coordenadas (*p*,*q*) do porão do navio com orientação *i*, e $x_{ipq} = 0$ caso contrário.

Dessa forma, o PEUC pode ser formulado como proposto por Beasley (1985), no qual o objetivo é maximizar a quantidade de vértices com coordenadas *p* e *q*.

$$v(PEUC) = Max\left(\sum_{i=1}^{2} \sum_{\{p \in X | p \leq L - l_i\}} \sum_{\{q \in Y | q \leq W - w_i\}} x_{ipq} \right) \quad (4)$$

Sujeito a:

$$\sum_{i=1}^{2} \sum_{\{p \in X | p \leq L - l_i\}} \sum_{\{q \in Y | q \leq W - w_i\}} a_{ipqrs} x_{ipq} \leq 1, \forall r \in X \; e \; s \in Y \quad (5)$$

$$x_{ipq} \in \{0,1\} \forall i = 1 \ldots 2, p \in X \; tal \; que \; p \leq L - l_i, e \; q \in Y \; tal \; que \; q \leq W - w_i \quad (6)$$

A restrição definida pela Equação (5) garante a inexistência de sobreposições das unidades de celulose e a Equação (6) indica que as variáveis de decisão sejam

binárias. As Equações (4)-(6) produzem diversas restrições e variáveis para as instâncias práticas do PEUC, inviabilizando a solução do problema de forma exata, mesmo com pacotes comerciais de otimização.

Do ponto de vista científico, este trabalho aplica um método recente e que ainda não foi explorado para resolução do PEUC, para com isso apresentar comparações de desempenho com técnicas já apresentadas na literatura para esse problema.

1.2 OBJETIVOS

Nesta Seção, são apresentados os objetivos gerais e específicos deste trabalho.

1.2.1 Objetivo geral

Este trabalho tem por objetivo aplicar a meta-heurística *Adaptive Large Neighborhood Search* (ALNS) para a resolução do Problema de Estivagem de Unidades de Celulose (PEUC), ou seja, é apresentada uma nova alternativa para resolução do problema exposto. Os objetivos específicos são apresentados a seguir.

1.2.2 Objetivos específicos

a. Desenvolver as adaptações necessárias para modelagem do PEUC e aplicação da ALNS;

b. Aplicar o método proposto às principais instâncias encontradas na literatura.

c. Avaliar o desempenho do método por meio de experimentos computacionais, comparando com outros trabalhos existentes na literatura.

O restante deste trabalho está organizado da seguinte forma. O Capítulo 2 apresenta uma revisão bibliográfica sobre o PEUC e a ALNS. Uma descrição detalhada da metodologia utilizada é apresentada no Capítulo 3, e os resultados são expostos no Capítulo 4. Por fim, no Capítulo 5 são apresentadas as conclusões deste trabalho.

2 REVISÃO DE LITERATURA

Considerando que o Problema de Estivagem de Unidades de Celulose (PEUC) pode ser analisado como um Problema de Carregamento de Paletes do Produtor (PCPP), e que este possui muitas aplicações práticas, os trabalhos iniciais sobre o PCPP surgiram no final da década de 70 e início da década de 80, quando Steudel (1979) e Smith e De Cani (1980) utilizaram métodos construtivos que dividem o palete em blocos.

Dowsland (1987) e Alvarez-Valdes et al. (2005) propuseram métodos exatos que usam basicamente uma estrutura de árvore para resolver o PCPP, porém essas abordagens são computacionalmente intratáveis dentro de um limite aceitável de tempo em situações práticas.

Young-Gun e Maing-Kyu (2001) usaram métodos recursivos e Scheithauer e Terno (1996) utilizaram métodos baseados em estruturas do tipo G4. Já em 2005, Pureza e Morabito (2006) aplicaram a Busca Tabu para o PCPP.

Segundo Sirtoli et al. (2006), o PEUC difere do PCPP devido à sua magnitude. Enquanto o PCPP apresenta instâncias na literatura com até 150 caixas, o PEUC apresenta problemas reais variando de 150 a 550 unidades de celulose.

Para o PEUC, Ribeiro e Lorena (2005) aplicaram uma relaxação lagrangiana com *clusters,* que a partir da modelagem do PEUC, monta um grafo de conflitos e então uma heurística de particionamento é aplicada no grafo, dividindo-o em *clusters* que podem ser resolvidos independentemente. Em seguida, esses *clusters* são reagrupados para atualização dos multiplicadores lagrangianos, repetindo o processo até que o *gap* entre os limitantes superiores e inferiores sejam fechados.

Sirtoli et al. (2006) também trataram o PEUC, mas os autores utilizaram as heurísticas de blocos propostas por Steudel (1979) e Smith e De Cani (1980), que trabalham com subdivisões das dimensões do porão em função das dimensões da unidade. As possíveis partições são formadas por combinações de unidades, com comprimento l e largura w, na dimensão L ou W, que não excedem as dimensões do porão.

Neste trabalho foi utilizada a meta-heurística *Adaptive Large Neighborhood Search* (ALNS), proposta por Ropke e Pisinger (2006), que é uma variação da meta-heurística *Large Neighborhood Search* (LNS) proposta por Shaw (1997). Essa meta-heurística permite a utilização de múltiplos métodos de destruição e reconstrução dentro de um mesmo processo de busca. Alguns dos trabalhos mais recentes encontrados na literatura que usam essa abordagem para a resolução de problemas de otimização são: Ribeiro e Laporte (2012), que utilizaram a ALNS para a resolução do Problema de Roteamento de Veículos Cumulativo e Capacitado; e Ribeiro et al. (2012), que aplicaram esse método para resolução do Problema de Roteamento de Sondas de Manutenção.

3 METODOLOGIA

A metodologia utilizada neste trabalho baseia-se na aplicação da meta-heurística *Adaptive Large Neighborhood Search* (ALNS) ao Problema de Estivagem de Unidades de Celulose (PEUC).

3.1 REPRESENTAÇÃO DE UMA SOLUÇÃO

Considerando um problema hipotético de dimensões L = 5, W = 4, l = 3 e w = 2 e as Equações (1)-(2) para a geração do conjunto X e Y de coordenadas p e q dos vértices, a Figura 3 ilustra os 7 possíveis vértices onde podem ser alocadas as unidades de celulose.

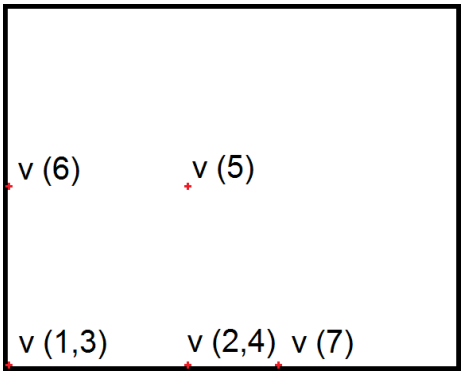

Figura 3 - Vértices de um espaço de busca.

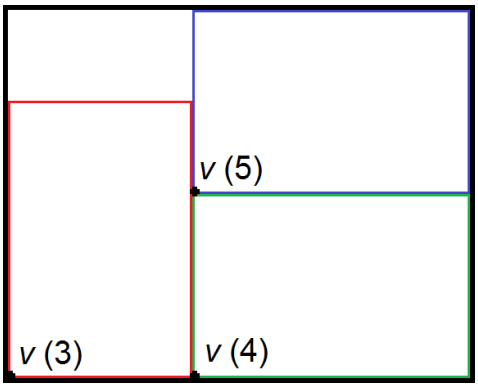

Figura 4 - Solução viável.

Para a geração de uma solução viável, são posicionadas unidades de celulose a partir do canto inferior esquerdo das mesmas, nos vértices definidos anteriormente,

de modo de não haja sobreposições entre as unidades estivadas. A Figura 4 apresenta uma solução viável, onde as unidades de celulose estão posicionadas nos vértices 3, 4 e 5.

A Figura 5 apresenta uma solução inviável, que é identificada por meio de sobreposições de arestas das unidades de celulose posicionadas nos vértices 2, 3 e 6 do porão do navio.

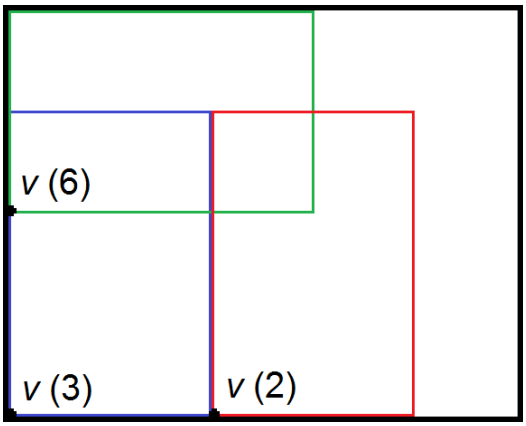

Figura 5 - Solução inviável.

Computacionalmente, uma solução para o PEUC é representada como um vetor binário *vet,* de dimensão *N*, na qual cada posição vet_k, para k = 1,...,*N*, representa um vértice de coordenadas *p* e *q* do canto inferior esquerdo de uma unidade de celulose. Os valores 1 (um) e 0 (zero) do vetor binário *vet* significam que o vértice *k* está ou não na solução. A Figura 6 representa a solução viável apresentada na Figura 4. Nessa solução, apenas os vértices 3, 4 e 5 estariam na solução, ou seja, apenas as 3 unidades de celulose cujo canto inferior esquerdo é representado pelos vértices 3, 4 e 5 seriam colocadas no porão do navio.

1	2	3	4	5	6	7
0	0	1	1	1	0	0

Figura 6 - Representação de uma solução.

3.2 ADAPTIVE LARGE NEIGHBORHOOD SEARCH

A *Adaptive Large Neighborhood Search* (ALNS) foi proposta por Ropke e Pisinger (2006) e permite que múltiplos métodos de destruição e reconstrução sejam utilizados dentro de um mesmo processo de busca.

Para cada heurística de destruição/reconstrução é atribuído um peso que controla a frequência que cada uma é utilizada durante a busca, sendo esses pesos ajustados dinamicamente à medida que a busca prossegue.

A cada iteração, o algoritmo destrói parte da solução corrente *s* e a recria de uma maneira diferente, gerando uma nova solução *s*'. Essa nova solução é aceita de acordo com um critério definido que, neste trabalho, será o mesmo critério utilizado pela meta-heurística *Simulated Annealing* (SA), proposta por Kirkpatrick (1983), na qual, se a solução *s*' for melhor do que a *s*, a busca continua a partir *s*', caso contrário, a busca continua de *s*' dada uma probabilidade.

Na ALNS, os procedimentos de destruição e reconstrução são selecionados de acordo com um mecanismo probabilístico. A cada iteração, a probabilidade de selecionar um dado procedimento depende do seu desempenho nas iterações passadas.

Para a resolução do PEUC, a ALNS foi implementada considerando as características descritas a seguir, e as heurística de destruição e reconstruções são apresentadas na Seção 3.3.

Vizinhança Ampla: dada uma solução *s*, a cada iteração, *q* vértices são removidos da solução utilizando uma das heurísticas de destruição e, utilizando uma das heurísticas de reconstrução, cria-se uma nova solução *s'* a partir dos *q* vértices removidos de *s*.

Busca Adaptativa: a escolha das heurísticas de destruição e reconstrução é feita por meio do mecanismo da roleta, no qual, a cada heurística é atribuído um peso que depende do seu comportamento passado. Mais precisamente, sendo ω_i a medida de quão bem a heurística *i* se saiu nas iterações passadas e, considerando *h* heurísticas com pesos ω_i, a heurística *j* é selecionada com a probabilidade $\omega_j / \sum_{i=1}^{h} \omega_i$. A heurística de reconstrução é independente da heurística de destruição.

Ajuste dos Pesos: a busca é dividida em um número de φ iterações consecutivas, chamadas de segmentos. No primeiro segmento todas as heurísticas têm o mesmo peso, isto é, $\omega_i = 1$ para $i = 1,...,h$. Depois de φ iterações, os pesos, usados para

selecionar as heurísticas de destruição e reconstrução, são atualizados de acordo com as pontuações obtidas durante o segmento. A pontuação da heurística é incrementada pelos parâmetros σ_1, σ_2 ou σ_3 quando é identificada uma nova solução. Para o PEUC, se um par de heurísticas de destruição-reconstrução encontrar uma nova melhor solução, suas pontuações serão incrementadas em σ_1, se encontrarem uma solução melhor do que a corrente, suas pontuações serão incrementadas em σ_2, e se encontrarem uma solução que não melhore a solução corrente ou a melhor solução encontrada, suas pontuações serão incrementadas por σ_3. Quando um segmento termina, novos pesos são calculados usando as pontuações obtidas, e todas as pontuações são atualizadas para zero para o próximo segmento. Considerando π_i e o_{ij} sendo, respectivamente, a pontuação da heurística *i* e o número de vezes que a heurística *i* foi escolhida no último segmento *j*. Então,

$$\omega_{i,j+1} = \begin{cases} \omega_{ij} & \text{se} \quad o_{ij} = 0 \\ (1-\eta)\omega_{ij} + \eta\pi_i/o_{ij} & \text{se} \quad o_{ij} \neq 0, \end{cases} \tag{7}$$

$\eta \in [0,1]$ é o parâmetro chamado fator de reação, que controla a velocidade com que o algoritmo de ajuste dos peso reage às mudanças na eficácia das heurísticas.

Formulação Relaxada: neste trabalho, a formulação do PEUC (4)-(6) é tratada de uma forma relaxada, ou seja, a restrição (5) será "ignorada" pela ALNS. Porém, para garantir a viabilidade das soluções obtidas, a função objetivo é modificada, considerando o número de sobreposições (β) e uma penalidade (Ψ) para o número de sobreposições. Assim, a função objetivo utilizada pela ALNS, para uma solução *s* qualquer, é descrita a seguir:

$$v(s) = Max\left[\left(\sum_{i=1}^{2} \sum_{\left\{\substack{p\in X \\ p\leq L-l_i}\right\}} \sum_{\left\{\substack{q\in Y \\ q\leq W-w_i}\right\}} x_{ipq}\right) - \beta * \Psi\right] \tag{8}$$

Critérios de Aceitação e de Parada: o critério de aceitação utilizado na ALNS é similar ao utilizada na SA, isto é, dada uma solução corrente *s*', uma solução *s* vizinha é aceita se *v*(*s*) > *v*(*s*') e, caso contrário, com uma probabilidade $e^{-(v(s)-v(s'))/T}$, sendo *T* a temperatura corrente e *v*(*s*') e *v*(*s*) os custos das soluções

s' e *s* definidas pela Equação (8). A temperatura inicia em T_0 e é decrementada em $T = \alpha T$ a cada iteração, onde $0 < \alpha < 1$ é a taxa de resfriamento.

3.3 HEURÍSTICAS DE DESTRUIÇÃO (H-) E RECONSTRUÇÃO (H+)

Essa seção descreve três heurísticas de destruição (remoção de vértices), que removem γ vértices de uma solução *s* e retornam uma solução *s'* e o conjunto dos *D* vértices removidos, e uma heurística de reconstrução (inserção de vértices), que reconstrói a solução *s'* a partir do conjunto dos *D* vértices.

3.3.1 Heurística de Remoção de Shaw (HRS) baseada na quantidade de sobreposições:

De acordo com Ropke e Pisinger (2006) e Shaw (1997), a ideia geral da heurística de remoção de Shaw é remover vértices que possuem alguma similaridade, de modo que seja sensivelmente mais fácil redistribuir vértices vizinhos, gerando assim novas ou melhores soluções. O grau de similaridade entre dois vértices *m* e *n* é computado por uma medida de similaridade $R_{HRS}(m,n)$, na qual valores mais baixos correspondem a uma maior similaridade entre os vértices da solução. A HRS é baseada na diferença absoluta entre a contribuição de *m* e *n* para a função objetivo. Nesse caso é utilizada a diferença absoluta entre a quantidade de sobreposições de *m* e de *n,* o que impacta diretamente no valor da função objetivo, pois quanto maior a quantidade de sobreposições de um vértice, menor será seu valor.

Incialmente, a heurística cria duas listas vazias *D* e *L* e preenche *L* com os vértices de *s.* Um vértice *r* é removido aleatoriamente de *s*, inserido em *D* e removido de *L*. Logo após esses passos iniciais, o algoritmo inicia um *loop*, com critérios de parada $|D| < \gamma$ e $|L| \neq 0$, que, a cada iteração do algoritmo, um vértice aleatório *r* é selecionado de uma lista *D*, então a lista *L* é ordenada segundo o $R_{HRS}(m,n) = |a_m - a_n|$ e um vértice aleatório *m* é escolhido segundo a regra $\lfloor y^{\delta}|L| \rfloor$, onde *y* é um valor entre 0 e 1 e δ um parâmetro usado para evitar determinismo ao selecionar um vértice. Em seguida, o vértice L_m é adicionado à lista *D* e removido de *s* e de *L.*

Assim sendo, a similaridade é usada de acordo com Shaw (1997) para remover γ vértices. Retornando uma solução *s* e um conjunto *D* de vértices removidos, o

algoritmo seleciona um vértice *k* aleatoriamente de *D,* calcula sua similaridade com todos os vértices ainda não removidos, e então escolhe um novo vértice para ser inserido em *D.* Esse processo é repetido enquanto $|D| < \gamma$. Como pode ser percebido na Figura 7, o parâmetro $\delta \geq 1$ é usado para evitar determinismo ao selecionar um vértice.

```
HRS (s, γ, δ ≥ 1)
1.  CRIAR (listas L e D);
2.  L ← vértices de s;
3.  r ← um vértice aleatório de s;
4.  D ← {r};
5.  REMOVER (r de s e de L);
6.  ENQUANTO (|D| < γ e |L| ≠ 0) FAÇA
7.       r ← um vértice aleatório de D;
8.       ORDENAR (L de modo que R_HRS(r,L_m) < R_HRS(r,L_n));
9.       y ← número aleatório do conjunto {0,1};
10.      m ← ⌊y^δ |L|⌋;
11.      D ← D ∪ {L_m};
12.      REMOVER (L_m de s e de L);
13. FIM-ENQUANTO;
14. RETORNAR (s e D);
```

Figura 7 - Heurística de Remoção de Shaw.

3.3.2 Heurística de Remoção do Pior (HRP):

De acordo com Ribeiro e Laporte (2012), essa heurística remove vértices com um alto custo na solução *s*. Seja $Custo^{-}(m,s) = v(s) - v_{-m}(s)$ o custo associado com o vértice *m* na solução corrente *s,* onde $v_{-m}(s)$ representa o custo da solução sem o vértice *m* em *s.* A heurística primeiro ordena os vértices de acordo com o custo $Custo^{-}(m,s)$, escolhe um para ser removido, recalcula $Custo^{-}(m,s)$ para os vértices restantes e então o processo é repetido enquanto $|D| < \gamma$.

Incialmente, a heurística cria duas listas vazias *D* e *L* e preenche *L* com os vértices de *s.* Logo após estes passos iniciais, o algoritmo inicia um *loop*, com critério de parada $|D| < \gamma$, que, a cada iteração do algoritmo, a lista *L* é ordenada segundo o critério $Custo^{-}(L_m,s) < Custo^{-}(L_n,s)$) e um vértice aleatório *m* é escolhido segundo a regra $\lfloor y^{\rho}|L| \rfloor$, onde *y* é um valor entre 0 e 1 e ρ um parâmetro. Em seguida, o vértice L_m é adicionado na lista *D* e removido de *s* e de *L*. Essa heurística é apresentada na Figura 8. É importante destacar que a remoção também é controlada por um parâmetro, $\rho \geq 1$, que evita determinismo.

```
HRP (s, γ, ρ ≥ 1)
1.  CRIAR (listas L e D);
2.  L ← vértices de s;
3.  ENQUANTO (|D| < γ) FAÇA
4.        ORDENAR (L de modo que Custo⁻(L_m,s) < Custo⁻(L_n,s));
5.        y ← número aleatório do conjunto {0,1};
6.        m ← ⌊y^ρ|L|⌋;
7.        D ← D ∪ {L_m};
8.        REMOVER (L_m de s e L);
9.  FIM-ENQUANTO;
10. RETORNAR (s e D);
```

Figura 8 - Heurística de Remoção do Pior.

3.3.3 Heurística de Remoção Aleatória (HRA):

Essa simples heurística, detalhada na Figura 9, remove aleatoriamente γ vértices da solução corrente e, segundo Ribeiro e Laporte (2012), tende a gerar um pobre conjunto de soluções, pois não utiliza nenhuma estratégia para a remoção dos vértices, mas ajuda a diversificar a busca.

A HRA cria inicialmente uma lista vazia *D* em seguida inicia um *loop*, com critério de parada, $|D| < \gamma$, em que, a cada iteração do algoritmo, um vértice *m* é escolhido aleatoriamente de *s*, inserido na lista *D* e removido da solução *s* em seguida.

```
HRA (s,γ)
1.  CRIAR (lista D);
2.  ENQUANTO (|D| < γ) FAÇA
3.        m ← vértice aleatório de s;
4.        D ← D ∪ {m};
5.        REMOVER (m de s);
6.  FIM-ENQUANTO;
7.  RETORNAR (s e D);
```

Figura 9 - Heurística de Remoção Aleatória.

3.3.4 Heurística de Inserção Gulosa (HIG):

Diferentemente de outras heurísticas de inserção encontradas na literatura, essa heurística não tenta reinserir o conjunto de *D* de vértices, obtidos pelas heurísticas de remoção já descritas, na solução corrente *s*. Sua estratégia consiste em tentar inserir os vértices que ainda não possuem unidades de celulose e que têm sobreposições com as unidades de celulose dos vértices do conjunto *D*, mas que não pioram a solução corrente *s*. Essa heurística é apresentada na Figura 10.

```
HIG (s,D)
1.   s' ← s;
2.   FAÇA
3.        m ← ∞;
4.        melhorou ← FALSO;
5.        PARA i ← 0 até |D| FAÇA
6.             PARA j ← 0 até C[Di] FAÇA
7.                  SE vértice M[Di][j] de s' = 0 ENTÃO
8.                       FO ← v(s');
9.                       vértice M[Di][j] de s' ← 1;
10.                      SE v(s') > FO e C[M[Di][j]] < m ENTÃO
11.                           s ← s';
12.                           melhorou = VERDADEIRO;
13.                           m ← C[M[Di][j]];
14.                      SENÃO
15                            vértice M[Di][j] de s' ← 0;
16                       FIM-SE;
17                  FIM-SE;
18             FIM-PARA;
19        FIM-PARA;
20   ENQUANTO (melhorou);
21   DELETAR (D);
22   RETORNAR (s);
```

Figura 10 - Heurística de Inserção Gulosa.

A partir de um conjunto *D*, com a lista de vértices removidos da solução *s*, é criada uma solução auxiliar *s'*, que será usada durante o processo de busca para guardar a melhor solução obtida durante as tentativas de inserir os vértices conflitantes com os vértices do conjunto *D.* As variáveis *m* e *melhorou* têm função de guardar o número de conflitos do vértice candidato a entrar na solução *s* e sinalizar que houve uma melhora na solução corrente *s*, respectivamente. Já a variável $C[D_m]$, é um vetor que contém o número de conflitos que o vértice D_m possui, enquanto a variável $M[D_m][n]$ é uma matriz, na qual cada linha possui um lista com todos os vértices que têm conflito com o vértice D_m.

3.3.5 Meta-heurística *Adaptive Large Neighborhood Search* (ALNS):

Uma vez expostas as heurísticas de remoção e reconstrução, o ALNS será detalhado, assim como todos os seus parâmetros e significados. Deste modo, os parâmetros e variáveis necessários para implementação da ALNS são descritos a seguir:

- $ALNS_{max}$: número máximo de iterações do método;
- φ: tamanho dos segmentos;

- η: fator de reação;
- σ_1: pontuação se a melhor solução é encontrada;
- σ_2: pontuação se a solução corrente é superada;
- σ_3: pontuação se uma solução pior for aceita;
- ζ: percentual de vértices removidos;
- α: taxa de resfriamento;
- T_0: temperatura inicial;
- T_c: temperatura de congelamento;
- ω_{H+}: peso da heurística de reconstrução H^+;
- ω_{H-}: peso da heurística de destruição H^-;
- π_{H+}: pontuação da heurística de reconstrução H^+;
- π_{H-}: pontuação da heurística de destruição H^-;
- o_{H+}: número de vezes que a heurística de reconstrução $H+$ foi escolhida no último segmento;
- o_{H-}: número de vezes que a heurística de destruição H^- foi escolhida no último segmento;

A Figura 11 apresenta o pseudocódigo da ALNS implementada para resolução do PEUC.

```
ALNS (ALNS_max, φ, η, σ1, σ2, σ3, ζ, α, T0, Tc, Tmax s)
1.  s* ← s; i ← 0; T ← T0; Te ← 0; Texec ← 0;
2.  ωH+ ← 1; Para todo H+   ωH- ← 1; Para todo H-
3.  πH+ ← 0; Para todo H+   πH- ← 0; Para todo H-
4.  ENQUANTO (Texec < Tmax) FAÇA
5.       i ← 0;
6.       ENQUANTO (i < ALNSmax) FAÇA
7.            i ← i + 1;     j ← j + 1;
8.            γ ← 2 + número aleatório entre 0 e (⌊ζ*n° vértices⌋ - 1)
9.            SELECIONAR (uma heurística de destruição H-);
10.           APLICAR (H-(s'));
11.           SELECIONAR (uma heurística de reconstrução H+)
12.           APLICAR (H+(s'));
13.           oH+ ← oH+ + 1;     oH- ← oH- + 1;
14.           SE (v(s') > v(s))
15.                s ← s';
16.                SE (v(s) > v(s*))
17.                     s* ← s;   πH+ ← πH+ + σ1;   πH- ← πH- + σ1;
18.                SENÃO
19.                     πH+ ← πH+ + σ2;   πH- ← πH- + σ2;
20.                FIM-SE;
21.           SENÃO
22.                TOMAR (x ∈ [0,1])
23.                SE (x < e^(-(v(s)-v(s')/T))
24.                     s ← s';   πH+ ← πH+ + σ3;   πH- ← πH- + σ3;
25.                FIM-SE;
26.           FIM-SE;
27.           SE (j = φ)
28.                j ← 0;
29.                ATUALIZAR (ωH-,πH-,ωH+,πH+);
30.           FIM-SE;
31.           T ← α*T;
32.      FIM-ENQUANTO;
33.      Texec ← tempo de execução;
34. FIM-ENQUANTO;
35. RETORNAR (s*);
```

Figura 11 - Pseudocódigo da ALNS implementada.

A cada iteração da ALNS, são selecionadas uma das heurísticas de destruição H^- e uma de reconstrução H^+ já descritas, e estas são aplicadas à solução s*'*. Após a aplicação destas heurísticas à solução corrente, a meta-heurística atualiza os seus valores o_{H+} e o_{H-} e os seus pesos em σ_1, σ_2 ou σ_3, de acordo com o valor da função objetivo encontrado. Quando um segmento termina, ou seja, $j = \varphi$, os pesos ω_{H+} e ω_{H-} das heurísticas são atualizados a partir dos pesos ω_{H+} e ω_{H-}, obtidos por segmentos passados e de π_{H+}, π_{H-}, o_{H+} e o_{H-}, obtidos durante o seguimento corrente, e em seguida π_{H+}, π_{H-}, o_{H+} *e* o_{H-} são zerados para a execução do próximo seguimento.

4 RESULTADOS

A meta-heurística *Adaptive Large Neighborhood Search* (ALNS) apresentada no capítulo anterior foi implementada na linguagem C++ e os experimentos computacionais descritos a seguir foram realizados em um computador com processador Intel Core i3-2330M de 2,2 GHz, com 4 GB de memória RAM. Com o intuito de verificar a complexidade do PEUC, o *solver* comercial CPLEX v. 12.1 também foi utilizado para resolver o modelo descrito pelas Equações (4) a (6).

Foram utilizadas 15 instâncias distintas baseadas em dados reais obtidos junto a um porto brasileiro. Essas instâncias foram utilizadas em Ribeiro e Lorena (2005) e Sirtoli et al. (2006). A Tabela 1 apresenta uma descrição dessas instâncias, contendo o comprimento (*L*) e a largura (*W*) dos porões dos navios e o comprimento (*l*) e largura (*w*) das unidades de celulose. Os números de vértices e de arestas foram definidos com base nas coordenadas *p* e *q* definidas pelas Equações (1) e (2), ou seja, essas informações definem o "tamanho" de cada instância.

Tabela 1 - Instâncias utilizadas.

Instância	**Porão**		**Unidade**		**Problema**	
	L	***W***	***l***	***w***	**nº vértices**	**nº arestas**
I1	2296	1230	136	94	24.292	15.669.282
I2	2536	1312	144	84	22.870	9.585.638
I3	2252	1470	144	84	23.413	10.155.396
I4	1470	1458	144	84	12.787	4.938.096
I5	2296	1230	135	92	25.911	17.344.486
I6	1804	1230	137	95	14.548	7.102.460
I7	2466	1230	137	95	27.072	18.485.016
I8	1804	1750	137	95	29.603	21.477.868
I9	2426	1230	137	95	26.316	17.694.300
I10	2530	1320	137	95	32.948	25.110.744
I11	2560	1610	143	84	58.808	62.738.446
I12	2625	1600	137	95	51.655	50.501.882
I13	1838	1600	137	95	25.335	16.869.124
I14	2100	1600	144	84	24.418	10.801.598
I15	1600	1230	137	95	11.300	4.798.444
Média	**2174**	**1407**	**139**	**91**	**27.418**	**19.551.519**

A definição dos vértices e arestas foi feita por meio de um programa gerado para esse fim específico, implementado também em C++, com base nas dimensões do problema (*L*, *W*, *l* e *w*) e de acordo com as Equações (1), (2) e (3). Os parâmetros da ALNS foram definidos de forma empírica, por meio de diversos experimentos. Assim, os valores que apresentaram os melhores resultados foram: $ALNS_{max}$ = 100.000,

T_{max} = 1.000 seg., T_0 = 10.000, T_c = 0.01, α = 0.975, φ = 50, η = 0.05, σ_1 = 50, σ_2 = 10, $\sigma 3 = 5$, $\delta = 2$, $\rho = 2$ e ζ = 0.05. Já a penalização para as sobreposições foi Ψ = 30.

A ALNS foi executada 10 vezes para cada instância, sendo o tempo computacional médio, para a obtenção das melhores soluções, de 806,11 segundos por instância. Além disso, o desvio entre a média das soluções obtidas (**Solução Média**) em relação a melhor solução obtida pela ALNS (***v*(*s**)**) foi de 2,18%, o que indica uma boa robustez do método, ou seja, as soluções obtidas são sempre próximas. Porém, o desvio médio entre as melhores soluções obtidas pela ALNS em relação às melhores soluções conhecidas (Desvio*) foi de 17,37%, indicando que, apesar de obter sempre soluções viáveis, a ALNS apresentou soluções inferiores às já conhecidas.

Tabela 2 - Resultados obtidos pela ALNS.

Instância	**Solução *v*(*s**)**	**Solução Média**	**Tempo (seg.)**	**Desvio (%)**	**Desvio* (%)**
L01	184	180,00	969,87	2,17	17,81
L02	234	229,60	973,24	1,88	15,90
L03	227	225,80	826,37	0,53	16,68
L04	156	153,70	889,39	1,47	12,17
L05	184	181,40	678,57	1,41	19,73
L06	149	145,60	617,61	2,28	13,33
L07	193	186,40	590,87	3,42	19,31
L08	199	194,10	911,39	2,46	19,13
L09	189	184,80	915,66	2,22	18,59
L10	211	204,00	1009,09	3,32	19,69
L11	279	269,70	393,59	3,33	20,68
L12	260	249,00	896,63	4,23	21,94
L13	184	181,80	628,4	1,20	18,48
L14	235	231,20	995,01	1,62	16,53
L15	133	131,50	795,96	1,13	10,54
Média	201,13	196,57	806,11	2,18	17,37

A Tabela 3 apresenta uma comparação de desempenho entre a ALNS, o método LagClus (RIBEIRO; LORENA, 2005), algumas heurísticas de blocos (SIRTOLI et al., 2006) e o CPLEX. Além disso, também são apresentadas as soluções utilizadas “na prática” pelo porto. Os tempos (**T**) apresentados nessa tabela estão em segundos.

Comparando com os resultados obtidos pelo CPLEX, a ALNS mostrou-se mais eficiente para resolução do Problema de Estivagem de Unidade de Celulose (PEUC), pois esta encontrou soluções viáveis para todas as instâncias, enquanto que o CPLEX só conseguiu obter soluções para as duas menores instâncias (I4 e I15).

Tabela 3 - Comparação do desempenho da ALNS com outros métodos encontrados na literatura.

Inst.	Solução Porto	CPLEX		ALNS		Sirtoli et al. (2006)										Rib. e Lor. (2005)	
						Um Bloco		Dois Blocos		Três Blocos		Quatro		Cinco Blocos		LagClus	
		v(*s**)	T	*v*(*s**)	T	*v*(*s**)	T	*v*(*s**)	T	*v*(*s**)	T	*v*(*s**)	T	*v*(*s**)	T	*v*(*s**)	T
I1	213	-	150,85	184	969,87	216	0,00	219	0,00	219	0,00	219	0,01	219	0,04	219	4768
I2	269	-	68,13	234	973,24	270	0,00	270	0,00	273	0,00	273	0,01	273	0,02	273	24637
I3	265	-	72,01	227	826,37	260	0,00	267	0,00	270	0,00	271	0,01	271	0,03	271	9634
I4	170	150	7202,49	156	889,39	170	0,00	174	0,00	174	0,00	175	0,00	175	0,01	175	2889
I5	217	-	93,04	184	678,57	221	0,00	224	0,00	224	0,00	226	0,01	226	0,03	226	6268
I6	158	-	54.69	149	617,61	156	0,00	166	0,00	166	0,00	168	0,00	168	0,01	168	907
I7	222	-	3,88	193	590,87	216	0,00	230	0,00	230	0,00	230	0,01	231	0,02	231	7460
I8	234	-	31,17	199	911,39	234	0,00	236	0,00	239	0,00	240	0,01	240	0,03	240	16249
I9	204	-	15,16	189	915,66	204	0,00	220	0,00	226	0,00	227	0,01	227	0,03	227	7384
I10	246	-	0,00	211	1009,09	234	0,00	250	0,00	250	0,00	254	0,01	254	0,03	*	*
I11	335	-	-	279	393,59	330	0,00	338	0,00	338	0,00	340	0,02	340	0,05	*	*
I12	310	-	-	260	896,63	304	0,00	317	0,00	317	0,00	318	0,01	319	0,06	*	*
I13	208	-	25,85	184	628,40	209	0,00	219	0,00	219	0,00	223	0,01	223	0,03	*	*
I14	264	-	79,55	235	995,01	275	0,00	277	0,00	277	0,00	277	0,01	277	0,03	*	*
I15	144	134	7204,16	133	795,96	132	0,00	142	0,00	143	0,00	146	0,00	147	0,00	*	*
Média	230,60	142,00	1245,52	201,13	806,11	228,73	0,00	236,60	0,00	237,67	0,00	239,13	0,01	239,33	0,03	225,56	8910,67

- O Cplex não apresentou soluções. * O LagClus não apresentou soluções.

Comparando os resultados obtidos pela ALNS com os obtidos pelas heurísticas de blocos, a ALNS foi menos eficiente. Enquanto as heurísticas de um, dois, três, quatro e cinco blocos obtiveram soluções médias de 228,73, 236,60, 237,67, 239,13 e 239,33 unidades de celulose estivadas respectivamente, a ALNS obteve uma solução média de 201,13, e com um tempo computacional muito superior.

O LagClus também mostrou-se mais eficiente que a ALNS e obteve uma média de 225,56 unidades de celulose estivadas por instância, numa média de 8910,67 segundos. Porém, o LagClus não foi capaz de gerar soluções para as instâncias I11, I12, I13, I14 e I15.

Por fim, em relação às soluções obtidas no porto, a ALNS também não conseguiu bons resultados. Entretanto, vale destacar que o planejamento da estivagem utilizado pelo porto não teve seu tempo cronometrado, ficando assim inviável uma comparação “justa” com a ALNS.

5 CONCLUSÕES

Este trabalho abordou o Problema de Estivagem de Unidades de Celulose (PEUC) em porões de navios especializados nesse tipo de transporte. Foi visto que o PEUC pode ser reduzido ao caso bidimensional e foi implementado o método *Adaptive Large Neighborhood Search* (ALNS) para a resolução deste problema.

Ao comparar os resultados obtidos pelo método com os já existentes na literatura, verificou-se que as soluções geradas pela ALNS não foram eficientes. Uma possível explicação para estes resultados deve-se às dimensões das instâncias testadas, que possuem uma média de 27.418 vértices e 19.551.519 arestas. Assim, esse número elevado de variáveis influencia diretamente no processo de busca de soluções pela ALNS, ou seja, a meta-heurística não consegue “cobrir” todo o espaço de soluções em tempo hábil, usando a modelagem computacional utilizada neste trabalho.

Porém, mesmo que a ALNS não tenha gerado resultados tão eficientes quanto os apresentados por Ribeiro e Lorena (2005) e Sirtoli et al. (2006), todas as soluções obtidas são viáveis, e podem servir de base para outros estudos envolvendo a modelagem utilizada e o uso de outras técnicas para a resolução deste problema.

REFERÊNCIAS BIBLIOGRÁFICAS

ALVAREZ-VALDES, R.; PARREÑO, F.; TAMARIT, J. M. A branch-and cut algorithm for the pallet loading problem. *Computers & Operations Research*, v. 32, n. 11, p. 3007-3029, 2005.

BEASLEY, J. An exact two-dimensional non guillotine cutting tree search procedure. *Operations Research*, v. 33, n. 1. p. 49-64, 1985.

CCE - Comissão das Comunidades Europeias. *Objetivos estratégicos e recomendações para a política comunitária de transporte marítimo no horizonte de 2018*. Comunicação da Comissão ao Parlamento Europeu, ao Conselho, ao Comitê Econômico e Social Europeu e ao Comitê das regiões. 2009. Disponível em: http://eur-ex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2009:0008:FIN:PT:PDF. Acesso em: 18 de nov. 2012.

DOWSLAND, K. An exact algorithm for the pallet loading problem. *European Journal of Operational Research*, v. 31, n. 1. p. 78-84, 1987.

DYCKHOFF, H. A typology of cutting and packing problems. *European Journal of Operational Research*, v. 44, p. 145-159, 1990.

HANDABAKA, A. R. *Gestão logística da distribuição física internacional*. São Paulo: Maltese, 1994.

ICC. *ICC - International Chamber of Commerce*, 2010. 130 p.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *Science*, v. 220, n. 4598, p. 671-680, 1983.

PUREZA, V.; MORABITO, R. Some experiments with a simple tabu search approach for the manufacture's pallet loading problem. *Computers & Operations Research*, v. 33, n. 3, p. 804-819, 2006.

RIBEIRO, G. M. *Relaxação lagrangeana com divisão em clusters para alguns problemas de otimização modelados em grafos de conflitos*. Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2007.

RIBEIRO, G. M.; LAPORTE, G. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, v. 39, n. 3, p. 728-735, 2012.

RIBEIRO, G. M.; LAPORTE, G.; MAURI, G. R. A comparison of three metaheuristics for the workover rig routing problem. *European Journal of Operational Research*, v. 220, n. 1, p. 28-36, 2012.

RIBEIRO, G. M.; LORENA, L. A. N. Análise da estivagem de unidades de celulose usando relaxação lagrangeana. In: *Congresso de Pesquisa e Ensino em Transportes*, 19, 2005, Recife. Anais… ANPET, 2005.

ROPKE, S.; PISINGER, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, v. 40, n. 4, p. 455-472, 2006.

SCHEITHAUER, G.; TERNO, J. The g4-heuristic for the pallet loading problem. *Journal of the Operational Research Society*, v. 47, n. 4, p. 511-522, 1996.

SILVA, J. L. C.; SOMA, N. Y.; MACULAN, N. Uma heurística para o problema de empacotamento de bins tridimensionais. In: *Encontro Nacional dos Estudantes de Engenharia de Produção*, 21, 2001, Salvador. Anais... ENEGEP, 2001.

SIRTOLI, D. R.; RIBEIRO, G. M.; LORENA, L. A. N. Estivagem de unidades de celulose: uma análise prática com heurísticas de bloco. In: *Encontro Nacional dos Estudantes de Engenharia de Produção*, 26, 2006, Fortaleza. Anais… ENEGEP, 2006.

SHAW, P. *A new local search algorithm providing high quality solutions to vehicle routing problems*. University of Strathclyde - Glasgow, 1997. (Relatório Técnico). Disponível em: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.1273&rep=rep1&type=pdf. Acesso em: 9 de fev. 2012.

SMITH, A.; DE CANI, P. An algorithm to optimize the layout of boxes in pallets. *Journal of the Operational Research Society*, v. 31, p. 573–578, 1980.

STEUDEL, H. J. Generating pallet loading patterns: a special case of the two-dimensional cutting stock problem. *Management Science*, v. 25, p. 997, 1979.

YOUNG-GUN, G.; MAING-KYU, K. A fast algorithm for two dimensional pallet loading problems of large size. *European Journal of Operational Research*, v. 134, n. 1, p. 193-202, 2001.

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS AGRÁRIAS
DEPARTAMENTO DE COMPUTAÇÃO

VINICIUS MIRANDA LOPES SCHULZ

ADAPTIVE LARGE NEIGHBORHOOD SEARCH PARA RESOLUÇÃO DO PROBLEMA DE ESTIVAGEM DE UNIDADES DE CELULOSE

ALEGRE - ES
2013