

Mariella Berger

*Aplicação da Busca Tabu no Gerenciamento de
Pilhas em Pátios de Armazenamento de Placas
de Aço*

Vitória - ES, Brasil

28 de setembro de 2007

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Mariella Berger

*Aplicação da Busca Tabu no Gerenciamento de
Pilhas em Pátios de Armazenamento de Placas
de Aço*

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre em Informática na área de Otimização.

Orientadores:

Arlindo Gomes de Alvarenga
Hannu Tapio Ahonen

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Vitória - ES, Brasil

28 de setembro de 2007

Dissertação apresentada ao Programa de Pós-Graduação em Informática sob o título “*Aplicação da Busca Tabu no Gerenciamento de Pilhas em Pátios de Armazenamento de Placas de Aço*”, defendida por Mariella Berger e aprovada em 28 de setembro de 2007, em Vitória, Estado do Espírito Santo, pela banca examinadora constituída pelos doutores:

Prof. Dr. Arlindo Gomes de Alvarenga
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Hannu Tapio Ahonen
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Sérgio Antônio Andrade de
Freitas
Universidade Federal do Espírito Santo

Dr. Fernando José Martinelli Loureiro
ArcelorMittal Tubarão

Resumo

Nesta dissertação é apresentada uma ferramenta de suporte a tomada de decisão para alocar, em pátios de estocagem de uma usina siderúrgica, placas de aço que serão transformadas em bobinas. O objetivo desta ferramenta é otimizar a utilização dos equipamentos responsáveis por alocar e despachar as placas no pátio, denominados pontes rolantes. Para resolver tal problema, foi utilizada a Meta-Heurística Busca Tabu.

Palavras-chaves: alocação, busca tabu, despacho, movimentações internas, pátios, placas, pontes rolantes.

Abstract

In this dissertation is presented a decision support tool for loading, in a storage yard of siderurgical company, slabs that will be transformed in coils. The objective of this tool is to optimize the utilization of equipments, identified by overhead cranes, responsables for load and unload plates in yard. To solve this problem, a Meta-Heuristic Tabu Search was used.

Keywords: load, unload, tabu search, yards, slabs, overhead cranes, relocation movements.

Dedicatória

Dedico este trabalho aos meus grandes amores: Ciel, Junior e Mamãe.

Agradecimentos

Agradeço, inicialmente, à Mamãe e ao Júnior, pelo amor incondicional.

Ao Ciel, meu grande presente na Universidade, minha maior conquista na vida. Obrigada por tudo, sempre!

Aos professores Arlindo e Hannu, pela orientação e dedicação.

Ao Tio Raul, meu primeiro desorientador que tanto amo e que está me matando de saudades.

Ao Sérgio, que está sempre me ensinando, na universidade e na vida.

Aos amigos Hilário, Márcio e Alexandre. Certamente a graduação foi bem mais fácil por tê-los ao meu lado.

Ao amigo Daniel, especial, que está sempre ao meu lado.

Aos meus alunos queridos, que me fazem aprender sempre mais a cada dia.

Aos meus familiares, em especial, à Maria Luiza, que me ensina, diariamente, os verdadeiros valores da vida.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

Sumário

Lista de Figuras

Lista de Tabelas

1	Introdução	p. 12
1.1	Objetivo	p. 14
1.2	Estrutura da dissertação	p. 15
2	Descrição do Problema	p. 16
2.1	Contextualização na Área Siderúrgica	p. 17
2.2	Estudo de Caso: Pátio de Estocagem de Placas de Aço utilizadas na fabricação de Bobinas	p. 19
3	Solução Proposta	p. 24
3.1	A Busca Tabu	p. 25
3.2	A Busca Tabu aplicada ao Problema	p. 26
3.2.1	Construção da Solução Inicial	p. 26
3.2.2	Escolha dos Movimentos	p. 28
3.2.3	Restrições Tabu	p. 29
3.2.4	CrITÉrios de AspiraÇão	p. 29
3.2.5	Obtenção do Custo da Solução	p. 29
3.2.6	A Heurística Busca Tabu aplicada ao Problema de Alocação de Placas de Aço	p. 32
4	Testes Computacionais	p. 38

4.1 Testes	p. 39
5 Conclusões e trabalhos futuros	p. 43
Referências	p. 45
Anexo A – Algoritmo <i>Branch and Bound</i>	p. 47
A.1 Exploração do Espaço de Soluções	p. 47
A.2 <i>Lower Bounds</i>	p. 48
A.3 Solução viável inicial (<i>Upper Bound</i> Inicial)	p. 48
A.4 O algoritmo <i>Branch and Bound</i> aplicado ao Problema de Alocação de Placas de Aço	p. 48
A.4.1 Cálculo do <i>Lower Bound</i>	p. 49
A.4.2 Testes	p. 52
Anexo B – <i>Layout</i> dos Arquivos de Entrada	p. 56
Anexo C – <i>Layout</i> dos Arquivos de Saída	p. 58
Anexo D – Dados obtidos nos Testes	p. 59

Lista de Figuras

1	Etapas da Fabricação do Aço	p. 18
2	Alto Forno	p. 19
3	Laminação de uma placa de aço	p. 20
4	Placa sendo transformada em Bobina	p. 21
5	Vista superior de um pátio	p. 21
6	Vista lateral de um pátio	p. 22
7	Ponte rolante se preparando para retirar uma placa de uma esteira . . .	p. 22
8	Caixão de laminação	p. 23
9	Vista lateral de um pátio com empilhamento de placas em 7 camadas. As placas em azul são as que necessitam ser movimentadas para que a em cor verde seja despachada	p. 23
10	Simulação da geração de opções de armazenamento de uma placa . . .	p. 26
11	Opções de alocação	p. 27
12	Visão lateral de um pátio, para mostrar que ao despachar a placa em verde não será necessário a realização de movimentações internas extras	p. 30
13	Visão lateral de um pátio, para mostrar que ao despachar a placa em verde será necessário a realização de 2 movimentações internas extras .	p. 30
14	Visão lateral de um pátio, para mostrar que ao despachar a placa em verde será necessário a realização de 6 movimentações internas extras. Este corresponde ao pior caso	p. 31
15	Configuração inicial de um pátio fictício	p. 32
16	Alocação das Placas 9, 10 e 11 no pátio fictício	p. 32
17	Alocação da Placa 12 no pátio fictício	p. 33

18	Alocação da Placa 13 no pátio fictício	p.33
19	Alocação da Placa 14 no pátio fictício	p.34
20	Esquema da ferramenta de geração de Plano de Alocação do Pátio . . .	p.37
21	Gráfico correspondente ao Tempo Tabu 10	p.39
22	Gráfico correspondente ao Tempo Tabu 20	p.40
23	Gráfico correspondente ao Tempo Tabu 30	p.41
24	Gráfico correspondente ao Custo do Algoritmo Aleatório	p.42
25	Estado inicial de um pátio	p.49
26	Gráfico correspondente aos tempos obtidos	p.54
27	Gráfico correspondente aos custos obtidos	p.55

Lista de Tabelas

1	Custo em três situações	p. 30
2	Custo da alocação em situação fictícia	p. 34
3	Custo do despacho em situação fictícia	p. 35
4	Parâmetros de entrada relacionados ao algoritmo BT	p. 35
5	Custos Máximo, Mínimo, Médio e Desvio Padrão	p. 40
6	Custos Máximo, Mínimo, Médio e Desvio Padrão de todos os algoritmos	p. 41
7	Custo da alocação da placa C em cada pilha do pátio	p. 50
8	Parametrizações dos testes realizados	p. 53
9	Tempos obtidos	p. 53
10	Resultados obtidos	p. 53
11	Parâmetros de entrada relacionados ao algoritmo BT	p. 56
12	Parâmetros de entrada relacionados às placas	p. 57
13	Parâmetros de entrada relacionados ao algoritmo BT	p. 58
14	Resultados obtidos com tempo tabu 10	p. 59
15	Resultados obtidos com tempo tabu 20	p. 60
16	Resultados obtidos com tempo tabu 30	p. 61
17	Resultados obtidos com o algoritmo Aleatório	p. 62

1 Introdução

*“Um passo à frente e você não está
mais no mesmo lugar!”*

Chico Science

Devido a globalização e competitividade, que são características marcantes do mercado atual, as indústrias cada vez mais se deparam com forças competitivas que as ameaçam, e que, portanto precisam ser enfrentadas. Dentre essas forças podem ser citadas: rivalidade de concorrentes dentro de seu ramo de atividades e poder de barganha tanto dos clientes quanto dos fornecedores. Uma empresa pode sobreviver e ter sucesso a longo prazo se ela desenvolver eficazmente estratégias para enfrentar estas forças competitivas, como exemplo: liderança em custo, diferenciação, inovação, crescimento e alianças [OBrien 2001].

Apesar do interesse das empresas em desenvolver todas estas estratégias, a liderança em custo recebe especial destaque por ser almejado e valorizado por seus clientes. Existem várias formas de se tratar a redução de custos dos produtos, dentre elas: diminuir o lucro da empresa ou baixar o custo da produção e, conseqüentemente, o custo de venda ao cliente. Como diminuir o lucro não é interessante para a organização, o ideal é procurar alternativas para se baixar o custo do processo de produção.

São várias as formas de baixar o custo de uma produção. Uma delas é otimizar a utilização dos espaços de armazenamento dos produtos da organização, pois todo espaço mantido tem um custo de operacionalização associado. Dentre os parâmetros de custo podem ser citados: os recursos humanos, equipamentos necessários, consumo de energia elétrica, manuseio dos produtos armazenados e o custo da manutenção da área.

Todos os parâmetros citados possuem características bastante peculiares e, portanto, podem ser estudados separadamente afim de definir uma melhor estratégia para abordá-los. Nesse sentido, destaca-se o parâmetro de manuseio dos produtos que chegam e saem do local de armazenamento, pois ele pode ter efeito sobre os demais. Por exemplo, ao agilizar o despacho de materiais, menos são onerados os recursos necessários no processo, sejam eles humanos ou de equipamentos e, neste último caso, menor será o consumo de energia elétrica envolvido [Amorim 2006].

Tratar estes parâmetros de custo é extremamente complexo e, portanto, tratar desta situação de maneira genérica, além de complexo, seria ineficiente, pois cada organização possui suas próprias características no que se refere ao processo de estocagem. Desta forma, será adotado um estudo de caso, o que permitirá uma visão prática da situação.

Como estudo de caso foi escolhido o pátio de estocagem de placas de aço utilizadas na fabricação de bobinas de uma grande empresa Siderúrgica. Esta empresa é uma das maiores produtoras de aço do mundo. Com base nesta informação é fácil observar que há um fluxo intenso de materiais nos pátios e a utilização de ferramentas que auxiliem na redução dos custos envolvidos nestas áreas podem ser de grande ajuda.

O pátio de estocagem de placas de aço utilizadas na fabricação de bobinas é uma área onde as placas de aço ficam armazenadas antes de serem enviadas para o Laminador de Tiras a Quente (LTQ) e serem transformadas em bobinas. Nesta área, as placas podem ser empilhadas em aproximadamente 50 pilhas que podem ter até sete camadas de placas cada. Este empilhamento é o principal responsável pelo aumento da complexidade no processo de despacho. Maiores detalhes serão apresentados nos capítulos posteriores.

Existem várias maneiras de se posicionar as placas em um pátio. Além disso, o fluxo de entrada e saída de materiais é constante. A questão é: qual a melhor maneira de se fazer isso. O grau de complexidade deste problema é muito alto e quando são propostas soluções sem o auxílio de uma ferramenta informatizada, alocações ineficientes podem ser adotadas com frequência.

1.1 Objetivo

Neste trabalho será desenvolvida uma ferramenta de suporte a tomada de decisão para alocar, em pátios de estocagem de uma grande empresa siderúrgica, placas de aço que serão enviadas para o Laminador de Tiras a Quente (LTQ) para serem transformadas em bobinas. O objetivo desta ferramenta é otimizar a utilização dos equipamentos responsáveis pelas movimentações das placas neste pátio, denominados pontes rolantes, de forma a diminuir o número de movimentos que estes realizam no armazenamento e despacho das placas no pátio e conseqüentemente, reduzir os custos com manutenção de equipamentos, mão de obra especializada e energia elétrica.

Ao buscar referências bibliográficas para este trabalho, embora não tenham sido encontradas referências que abordem este problema no contexto de pátios de placas de aço, foram encontrados diversos trabalhos relacionados ao problema de alocação de *containers* em pátios, como exemplo podemos citar: Preston e Kozan [Preston e Kozan 2001], que propõem minimizar o custo do despacho de *containers* escolhendo as melhores posições para armazená-los no pátio através da utilização de algoritmos genéticos; Kim e Kim [Kim e Kim 1999] que sugerem uma metodologia para reduzir o número de realocações de *containers* através de uma alocação mais eficiente do espaço disponível com a utilização de Relaxação Lagrangeana; Preston e Kozan [Preston e Kozan 1999] que buscam minimizar o tempo de transferência de *containers* entre os navios e os pátios, visando a melhor estratégia de armazenagem, utilizando algoritmos genéticos.

Dada a complexidade e a natureza combinatorial do problema, observa-se que a utili-

zação de algoritmos que obtenha uma solução ótima demanda um tempo muito elevado de processamento. Desta forma, uma abordagem não exata para o tratamento do problema, baseada na meta-heurística Busca Tabu, foi conduzida.

1.2 Estrutura da dissertação

O texto está organizado em cinco capítulos, começando pela presente introdução.

O capítulo 2 apresenta a descrição detalhada do problema ao qual este trabalho se propõe a resolver, familiarizando o leitor no contexto de produção, armazenamento e despacho de placas de aço utilizadas na produção de bobinas.

O capítulo 3 descreve a solução adotada para resolver o problema, com foco no estudo de caso. O algoritmo utilizado bem como sua adaptação ao problema também serão discutidos.

O capítulo 4 apresenta os testes numéricos realizados a partir da ferramenta implementada para o estudo de caso, a fim de validar os apontamentos e decisões identificadas ao longo deste trabalho.

O capítulo 5 apresenta as conclusões e os indicativos de projetos futuros.

2 Descrição do Problema

*“Obstáculos são essas coisas
amedrontadoras que você vê quando
tira seus olhos do objetivo.”*

Hannah More

Este capítulo tem por objetivo apresentar uma descrição detalhada do problema para o qual esta dissertação propõe uma solução. Inicialmente é apresentada uma contextualização na área siderúrgica e, em seguida, um detalhamento do problema, com foco no estudo de caso.

2.1 Contextualização na Área Siderúrgica

A metalurgia é o conjunto de procedimentos e técnicas que o homem adquiriu com o decorrer do tempo que lhe permitiu extrair, fabricar, fundir e tratar metais e suas ligas.

Os primeiros metais a serem descobertos foram os metais nobres, que são encontrados em sua forma bruta na natureza. Estes metais passaram a ser trabalhados quando se descobriu que o calor podia amolecê-los. Acredita-se que, por volta de 2500 a.C., surgiram as primeiras ligas metálicas, com a adição de estanho ao cobre, gerando o bronze: um metal que tinha propriedades superiores às do cobre [Costa e Silva 2006]. O bronze era utilizado em ferramentas que começaram a suplantiar, lentamente, as de pedra e metais não fundidos. Este período histórico é conhecido como idade do bronze.

A revolução industrial representou a transição da sociedade agrária e têxtil para a sociedade industrial, que se baseava no carbono, como combustível, e no ferro, como matéria-prima fundamental para a fabricação das máquinas. Conhecido desde os tempos pré-históricos, o metal dá nome à idade do ferro, que sucedeu a idade do bronze.

A fronteira entre o ferro e o aço foi definida com a invenção de fornos que permitiam não só corrigir as impurezas do ferro, como adicionar-lhes propriedades como resistência ao desgaste, ao impacto, à corrosão, dentre outras. Por causa dessas propriedades e do seu baixo custo o aço passou a representar cerca de 90% de todos os metais consumidos pela civilização industrial [Instituto Brasileiro de Siderurgia 2007].

A siderurgia é o ramo da metalurgia que se dedica à fabricação e tratamento do aço. Basicamente, o aço é produzido a partir de minério de ferro, carvão, cal e coque. A fabricação do aço pode ser dividida em quatro etapas, apresentadas na Figura 1, e estas são descritas a seguir:

- **Preparação da carga:** Antes de serem levados ao alto forno, o minério e o carvão são previamente preparados para melhoria do rendimento e economia do processo. Grande parte do minério de ferro é aglomerada utilizando-se cal e coque.
- **Redução:** as matérias-primas, agora preparadas, são carregadas no alto forno. O ar aquecido é soprado pela parte inferior do alto forno, como mostra a Figura 2. O coque, em contato com o oxigênio, produz calor que funde a carga metálica e dá início ao processo de redução do minério de ferro em um metal líquido: o ferro-gusa.
- **Refino:** aciarias são utilizadas para transformar o gusa em aço líquido. Nessa etapa, parte do carbono contido no gusa é removida juntamente com impurezas. A maior

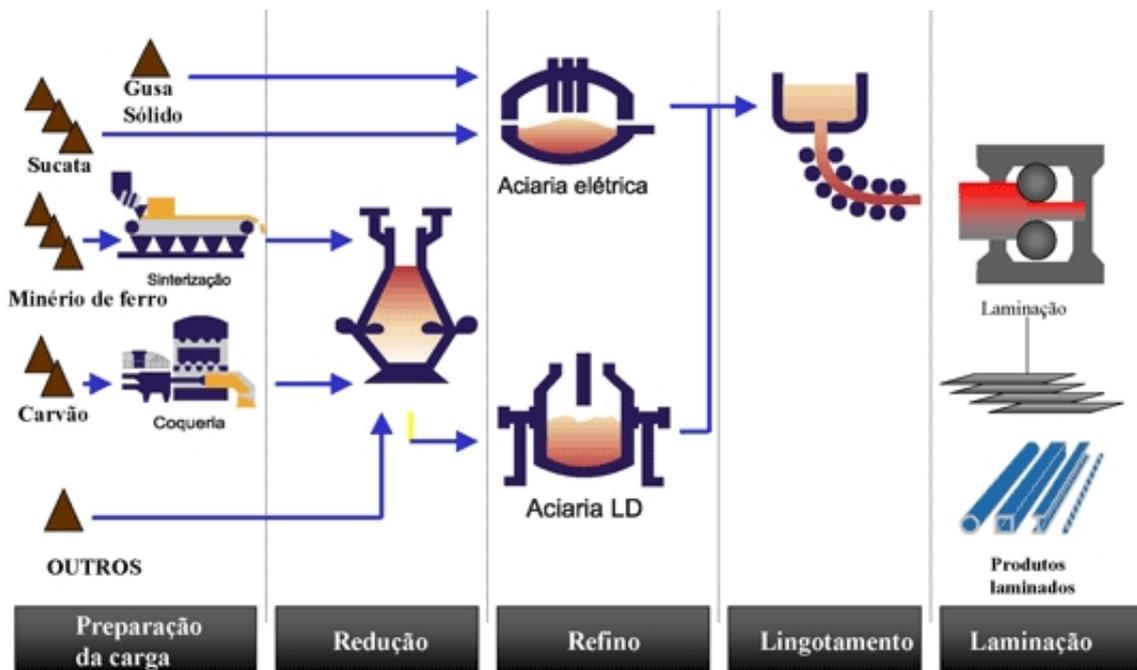


Figura 1: Etapas da Fabricação do Aço

parte do aço líquido é solidificada em equipamentos de lingotamento contínuo para produzir semi-acabados, como exemplo placas, lingotes e blocos.

- Laminação: os semi-acabados, lingotes e blocos são processados por equipamentos chamados laminadores e transformados em produtos siderúrgicos utilizados pela indústria de transformação, como chapas grossas e finas, bobinas, vergalhões, arames, perfilados e barras.

O foco deste trabalho está voltado para o pátio de estocagem de placas de aço utilizadas na produção de bobinas, um produto mais refinado, porém ainda considerado um produto semi-acabado, resultante do processo de laminação de uma placa. Na Figura 3 pode ser visualizada uma simplificação deste processo, onde a placa, ao passar pelos cilindros de laminação, tem sua espessura reduzida, sendo transformada em chapa de aço. Essa chapa é encaminhada para a bobinadeira, que é responsável por enrolá-la, gerando assim uma bobina. Um dos processos realizados na produção de uma bobina é ilustrado na Figura 4.

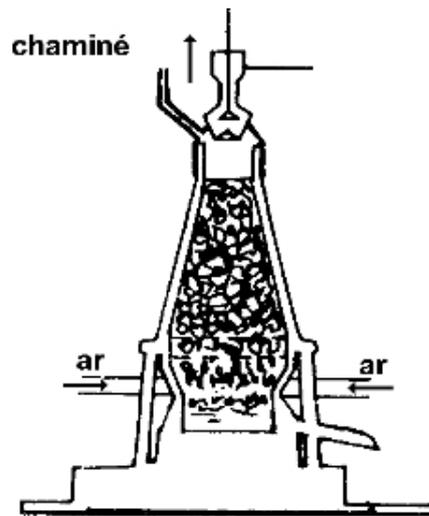


Figura 2: Alto Forno

2.2 Estudo de Caso: Pátio de Estocagem de Placas de Aço utilizadas na fabricação de Bobinas

Como estudo de caso foi escolhido o pátio de estocagem de placas de aço utilizadas na fabricação de bobinas de uma grande siderúrgica.

Após a fabricação das placas de aço, estas podem seguir dois fluxos distintos:

- Pátio de condicionamento de placas;
- Pátio de entrada do Laminador de Tiras a Quente (LTQ).

O pátio de condicionamento de placas é a área onde as placas produzidas são inspecionadas e, quando necessário, ajustadas para atender aos requisitos dos clientes, caso seja uma placa para venda, ou às restrições do LTQ. Depois de condicionadas, as placas a serem laminadas são encaminhadas para o pátio de entrada do LTQ para, no momento certo, serem laminadas. Este pátio é o foco desta dissertação.

O pátio de entrada do LTQ é dividido em alas e colunas. A Figura 5 representa a vista superior de um pátio que, neste caso, tem suas alas enumeradas de 1 e 2 e suas colunas de 1 a 50. Além disso, as placas podem ser empilhadas em até 7 camadas, como pode ser visualizado na Figura 6, que representa a vista lateral do pátio. Logo, o endereço de uma placa no pátio é composto pelas coordenadas x, y e z, que correspondem, respectivamente, a ala, coluna e camada.

As placas são alocadas no pátio em forma de pilhas através de pontes rolantes, equi-

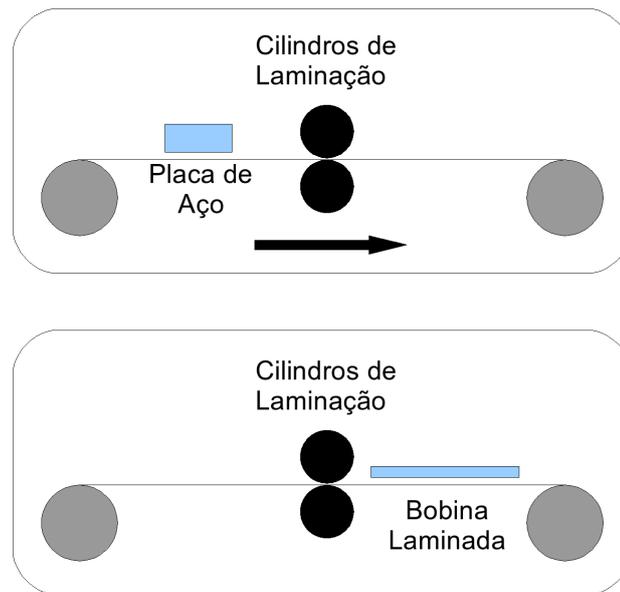


Figura 3: Laminação de uma placa de aço

pamentos que se movem sobre trilhos a uma determinada altura do pátio e que possuem garras capazes de transportar até 3 placas simultaneamente. A Figura 7 ilustra uma ponte rolante se preparando para retirar uma placa da esteira de entrada das placas.

A presença de restrições introduz ao problema uma complexidade adicional, por dividir o espaço de candidatos à solução em duas classes: soluções factíveis, que atendem a todas as restrições simultaneamente, e soluções infactíveis, que violam uma ou mais restrições.

Como restrição do pátio podemos citar placas de baixo peso ou baixa resistência não devem estar sob placas pesadas, pois isso poderia danificar a placa da camada inferior.

O LTQ também possui suas restrições: as placas a serem laminadas devem ser seqüenciadas de forma que os cilindros de laminação sofram o menor desgaste possível, ou ainda, que este desgaste seja previsível. Os cilindros de laminação sofrem desgaste devido ao esforço de compressão, ao atrito e à temperatura. As placas deixam marcas nos cilindros, causadas principalmente pelas bordas, que são regiões mais duras e críticas de laminar devido ao resfriamento rápido. Portanto, os cilindros devem ser trocados de tempos em tempos para que as marcas causadas pelo desgaste não passem para as bobinas, garantindo assim a qualidade superficial das mesmas, conseqüentemente, um maior valor do produto.

A seqüência de placas de aço que será laminada e transformada em bobinas entre duas trocas consecutivas de cilindros de laminação é chamada por alguns autores de caixão de



Figura 4: Placa sendo transformada em Bobina

		COLUNAS												
		1	2	3	4	5	6	7	8	9	10	...	50	
ALAS	1													
	2													

Figura 5: Vista superior de um pátio

laminação ou *product block*. Este nome se deve à semelhança do típico perfil de larguras que a seqüência de placas que compõem um caixão de laminação deve ter com o de um caixão, como observado na Figura 8. Este perfil visa o melhor aproveitamento do cilindro que, como dito anteriormente, fica marcado pelas bordas das placas a serem laminadas.

Quando uma placa chega ao pátio, ela deve ser imediatamente alocada em uma determinada posição (ala, coluna e camada). Essa posição deve ser decidida levando-se em consideração, além das restrições anteriormente citadas, o fato de que no movimento de alocação e despacho deve-se realizar o menor número de movimentações internas possível. Como exemplo, no caso de se despachar a placa destacada na cor verde (ala 1, camada 3) da Figura 9, é necessário movimentar para outra posição as quatro placas de cor azul localizadas acima dela. Estas movimentações de materiais de uma posição para outra dentro do mesmo pátio são denominadas movimentações internas.

A determinação das posições das placas no pátio de placas de entrada do LTQ não

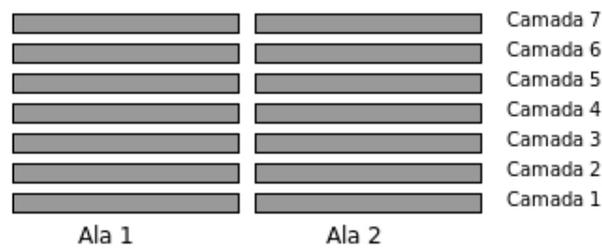


Figura 6: Vista lateral de um pátio

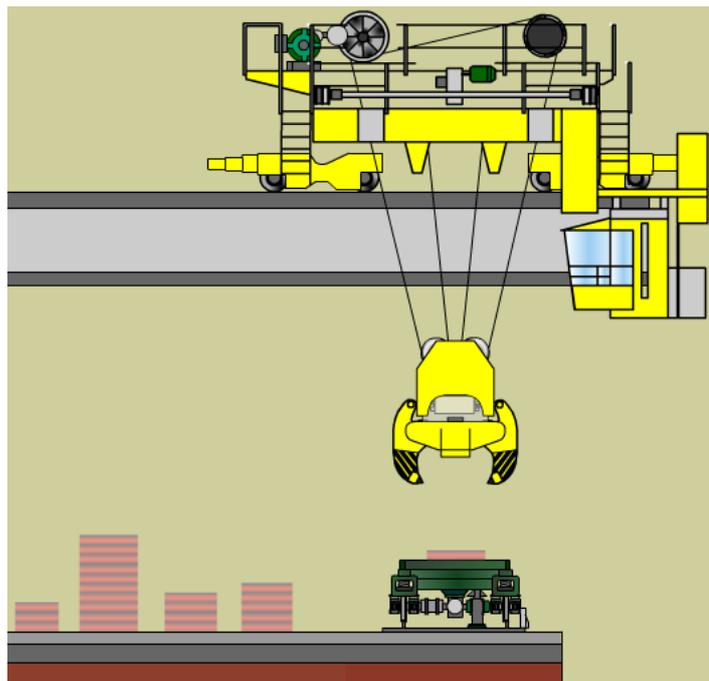


Figura 7: Ponte rolante se preparando para retirar uma placa de uma esteira

é feita atualmente visando uma diminuição de movimentos internos das pontes. Apenas é levado em consideração a ordem dos pedidos dos clientes e as restrições exigidas pelo equipamento de laminação.

Caso a diminuição da movimentação interna fosse levada em consideração, reduzindo-se tal número, seriam reduzidos também:

1. o risco de acidentes provocados pela queda de placas das garras das pontes rolantes;
2. o consumo de energia elétrica utilizada nos equipamentos do pátio;
3. o custo de manutenção de equipamentos;
4. e o tempo de alocação e despacho.

O problema é que não se pode garantir qual a ordem de chegada e saída das placas

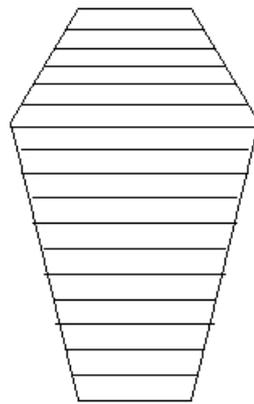


Figura 8: Caixão de laminação

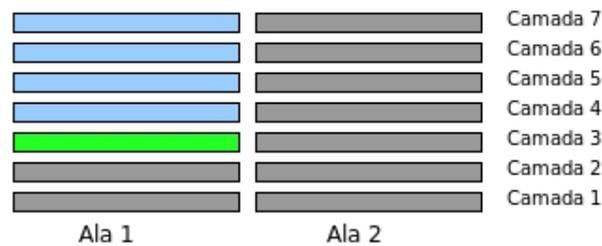


Figura 9: Vista lateral de um pátio com empilhamento de placas em 7 camadas. As placas em azul são as que necessitam ser movimentadas para que a em cor verde seja despachada no pátio. Mesmo havendo um plano de entrada de placas e de despacho para laminação, estes não necessariamente são respeitados. Estes planos têm a finalidade de prover às áreas internas da usina, informações que possibilitem o atendimento às demandas de produção [Amorim 2006].

Com o intuito de simplificar e apresentar um estudo com maior clareza de idéias serão consideradas apenas as previsões de chegada e saída das placas no pátio, a configuração atual do mesmo e a restrição de peso das placas para definir a melhor posição de alocação.

3 Solução Proposta

*“Se você tem uma maçã
e eu tenho outra maçã e as trocarmos
então cada um de nós continuará a ter uma maçã.*

*Mas, se você tem uma idéia
e eu tenho outra idéia e as trocarmos,
então cada um de nós terá duas idéias.”*

George Bernard Shaw

Neste capítulo é apresentada a estratégia adotada para se desenvolver uma ferramenta de suporte à tomada de decisão para auxiliar na alocação, em pátios de estocagem, de placas que serão transformadas em bobinas.

Inicialmente é apresentado o algoritmo utilizado na resolução do problema, bem como uma discussão da estratégia adotada para se obter um plano de alocação das placas de aço no pátio de estocagem. Posteriormente, é apresentada a ferramenta desenvolvida.

3.1 A Busca Tabu

A Busca Tabu (BT), segundo Glover e Laguna [Glover e Laguna 1997], é uma meta-estratégia para guiar heurísticas conhecidas a superar o problema da otimalidade local. De forma geral, é uma heurística de busca conhecida por geralmente superar o problema da convergência local em problemas de otimização.

A palavra tabu vem do Tongan, uma linguagem da Polinésia, e sugere algo proibido, que não pode ser tocado por ser sagrado. A heurística básica tem este nome por aplicar restrições para inibir determinadas escolhas durante o processo de busca de soluções. Desta forma, a BT conduz a busca para áreas ainda não analisadas do espaço de busca, tendendo a evitar a convergência da solução para um máximo (ou mínimo) local.

As restrições tabu são geralmente controladas por uma lista que memoriza os últimos movimentos executados, denominada Lista Tabu. O tempo que um movimento deve permanecer nesta lista, em geral, está relacionado com o número de iterações do algoritmo e com o conjunto de movimentos possíveis a partir da solução candidata atual (solução que está sendo analisada).

Alguns procedimentos denominados critérios de aspiração são utilizados para decidir quando movimentos, mesmo classificados como tabu, poderão ser executados.

A implementação de um algoritmo BT básico envolve decidir como os movimentos (a geração de novas soluções) são realizados, a definição dos critérios de aspiração do problema e como é feito o gerenciamento da memória dos movimentos.

Ao aplicarmos a heurística BT para resolução de problemas, devemos considerar seus pontos positivos e negativos. Como pontos positivos podemos destacar que a BT permite aceitar soluções sem melhoria para fugir do ótimo local, é aplicável em problemas com espaço de soluções contínuas ou não e é aplicável para problemas grandes e difíceis. Como pontos negativos, a BT necessita da definição de muitos parâmetros para sua utilização, pode exigir muitas iterações e pode não encontrar o ótimo global [Pinedo 1995].

Apesar de os pontos negativos sobre BT, existem várias aplicações nas mais diversas áreas conforme ilustrado por Glover [Glover e Laguna 1997]. Com o objetivo de ilustrar algumas aplicações da BT, podemos citar: Schepke et al [Schepke et al. 2004], que abordam o problema de roteamento de veículos, classificado como sendo de otimização combinatória, e consiste em arranjar o conjunto ótimo de rotas para determinada frota de veículos, de maneira a servir um dado conjunto de clientes; Gambardella et al

[Gambardella et al. 2001], que apresentam uma solução para a alocação de recursos e programação das operações de carregamento e descarregamento de *containers* em terminais portuários; Ryan et al [Ryan, Bailey e Moore 1998], que propõem uma simulação para resolver o problema de roteamento de veículos aéreos não tripulados; Oliveira [Oliveira 2006], que apresenta uma solução para a alocação de produtos granéis em terminais portuários de importação.

3.2 A Busca Tabu aplicada ao Problema

A seguir será descrito como foi aplicada a heurística BT ao problema descrito anteriormente.

3.2.1 Construção da Solução Inicial

Considerando S o conjunto de todas as alocações possíveis, uma solução inicial $S_0 \in S$, é gerada aleatoriamente da seguinte forma: para cada período de planejamento, realiza-se o despacho das placas a serem laminadas e, em seguida, identificam-se as placas que estão previstas para chegarem ao pátio. A partir daí, determina-se quais são as possíveis opções de endereços para armazenamento das placas e faz-se a escolha aleatória do local de armazenamento a partir das possibilidades identificadas. Este procedimento é executado até o final do período de planejamento.

A Figura 10 ilustra um exemplo do levantamento das possíveis opções de armazenamento de uma placa. Supondo que o pátio é constituído de 2 alas e 1 coluna e que todas as placas possuem as mesmas características, caso seja necessário alocar uma nova placa neste pátio, existem 2 possibilidades: pilha 1 ala 1 ou pilha 2 ala 2.

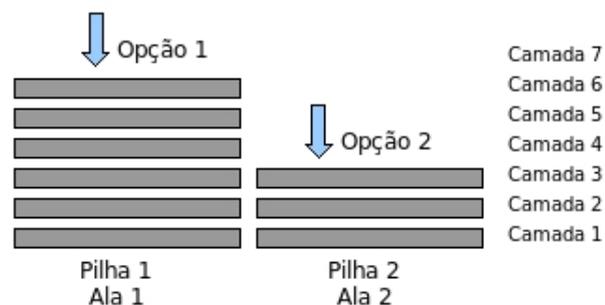


Figura 10: Simulação da geração de opções de armazenamento de uma placa

Desta forma, têm-se então as opções de armazenamento, identificando qual é o ende-

reço da placa no pátio e a quantidade de opções que foram encontradas.

Como dito anteriormente, o endereço das placas no pátio é dado pelas coordenadas x , y e z (ala, camada e coluna). Percebe-se que as coordenadas x e y caracterizam uma pilha no pátio, logo, estas foram agregadas formando uma única estrutura: a estrutura pilha. É válido salientar, ainda, que uma placa sempre poderá ser alocada em uma pilha que possua menos do que 7 placas, devido a possibilidade de, mesmo sendo mais pesada que as demais, ocupar a camada mais inferior da pilha. Caso isso aconteça, os endereços de todas as placas que trocaram de camadas deverão ser atualizados.

Para a construção da solução inicial, além do estado atual do pátio, teremos também um vetor de opções escolhidas. Este vetor armazena quais foram as opções escolhidas, dentre as possíveis, para o armazenamento de cada placa dentro do período proposto. A Figura 11 ilustra um exemplo.

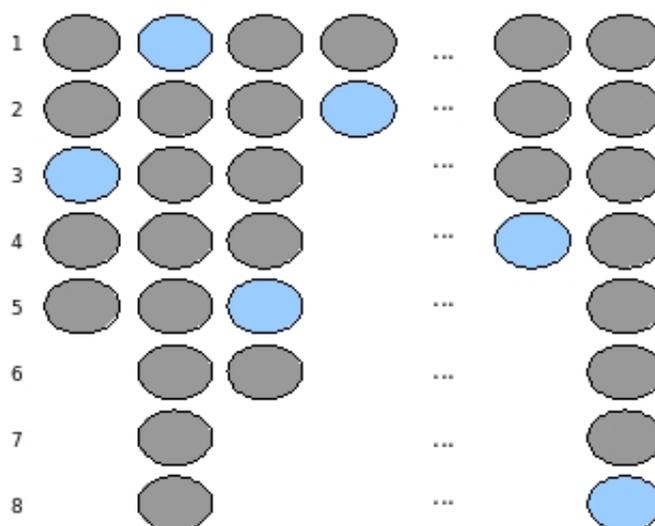


Figura 11: Opções de alocação

Cada coluna da Figura 11 representa as opções de armazenamento para cada placa. Como exemplo, a placa 1 possui 5 opções, a placa 2 possui 8 opções e assim sucessivamente. De acordo com a ilustração, ao atingirmos a solução inicial, teremos o seguinte vetor de escolhas: $M = [3, 1, 5, 2, \dots, 4, 8]$

De posse da solução inicial, novas soluções serão geradas conforme discutido a seguir.

3.2.2 Escolha dos Movimentos

Na heurística BT, a geração de novas soluções objetiva principalmente em analisar, a cada iteração do algoritmo, uma solução vizinha à solução corrente. Apesar de as soluções vizinhas serem geradas seguindo a idéia da geração da solução inicial, a escolha da opção do local de armazenamento não é feita aleatoriamente.

A estratégia de escolha de uma solução vizinha utiliza como base o vetor onde estão armazenadas as escolhas que foram realizadas para alocar as placas durante a geração da solução inicial. De posse deste vetor, realizam-se movimentos definidos através de incrementos e decrementos em cada posição do mesmo para a geração da solução vizinha. Uma possível representação para os movimentos é M_i^- e M_i^+ , e define, respectivamente, o decremento e o incremento do *iésimo* elemento do vetor.

Para ilustrar, suponha que existem posições livres em todas as pilhas do pátio. A partir do vetor de escolhas $M = [3, 1, 5, 2, \dots]$, para obtermos uma solução vizinha, decrementamos o valor do primeiro elemento do vetor, conforme abaixo:

$$M_1^- = [-1, 0, 0, 0, \dots] \Rightarrow M' = [2, 1, 5, 2, \dots]$$

M' é o novo vetor de escolhas gerado a partir de M pela operação decremento na posição 1 do vetor (M_1^-). Outra solução vizinha pode ser gerada a partir do incremento do primeiro elemento do vetor (M_1^+):

$$M_1^+ = [+1, 0, 0, 0, \dots] \Rightarrow M'' = [4, 1, 5, 2, \dots]$$

Esses movimentos são realizados para todas as posições do vetor M a fim de obtermos diversas soluções vizinhas para um mesmo período de tempo. É importante salientar que, para a geração das novas soluções, não é considerado que serão geradas apenas soluções melhores ou piores que a solução inicial e sim, novas soluções.

Para a geração de novas soluções do problema proposto, como movimento de incremento e decremento é realizado, na verdade, o incremento da pilha atual até que a pilha destino possua espaço livre, assegurando sempre a validade da solução gerada. Com o objetivo de simplificar a implementação, considerou-se que os endereços das pilhas estão dispostos em círculos.

Paralelo a geração das novas soluções, são armazenados também os movimentos efetuados sobre a solução atual (incremento de quantas pilhas) e sobre qual posição do vetor de soluções este movimento foi aplicado. Este armazenamento é realizado para a geração da lista tabu.

3.2.3 Restrições Tabu

Uma restrição tabu será criada toda vez que for aceita uma nova solução para o pátio dentre as soluções vizinhas que foram geradas. Quando uma solução é aceita, o inverso do movimento que deu origem a ela será colocado na lista tabu, armazenando também em qual posição do vetor de escolhas foi realizado tal movimento.

O número de iterações para um movimento permanecer classificado como tabu, ou seja, o tempo tabu, é parâmetro de entrada do algoritmo desenvolvido.

A cada iteração do algoritmo BT, contadores que marcam o tempo tabu de cada movimento pertencente a lista tabu serão decrementados de 1. Quando o contador de um determinado movimento chegar ao valor zero, este será retirado da lista tabu.

3.2.4 Critérios de Aspiração

São considerados critérios de aspiração aqueles que determinam quando uma restrição tabu pode ser quebrada e um movimento classificado como tabu pode ser realizado. Geralmente atingimos este critério quando o movimento considerado conduz a busca de uma solução de alta qualidade.

Para o problema em questão, será considerado como critério de aspiração, ou seja, será permitido realizar um movimento considerado tabu quando a solução gerada com o movimento tabu for melhor do que a melhor solução encontrada até o momento.

3.2.5 Obtenção do Custo da Solução

Durante os processos de alocação e despacho de uma placa, dependendo da posição que ela será alocada (no processo de alocação) ou de seu atual endereço no pátio (no processo de despacho), pode ser necessária a realização de várias movimentações extras de outras placas.

Como exemplo, para se despachar a placa em verde na Figura 12, não ocorrerão movimentações extras, pois não existem placas sobre ela. Já na Figura 13 será necessário retirar as 2 placas ilustradas em azul que estão sobre ela. E por fim, na Figura 14, que corresponde ao pior caso de despacho, serão necessárias 6 movimentações extras.

O custo de enviar uma determinada placa ao LTQ corresponde ao número de movimentações internas realizadas pelas pontes rolantes durante a alocação e durante o



Figura 12: Visão lateral de um pátio, para mostrar que ao despachar a placa em verde não será necessário a realização de movimentações internas extras



Figura 13: Visão lateral de um pátio, para mostrar que ao despachar a placa em verde será necessário a realização de 2 movimentações internas extras

despacho da mesma. Supondo não ter sido realizada nenhuma movimentação interna extra durante a alocação das placas no pátio, os custos nas três situações anteriores podem ser visualizados na Tabela 1, a seguir:

Situação	Custo
Figura 12	0
Figura 13	2
Figura 14	6

Tabela 1: Custo em três situações

Durante um determinado período, várias placas são alocadas e despachadas. Logo, o custo de um plano de alocação e despacho para um período pode ser descrito conforme a seguinte fórmula:

$$Custo_{Plano}(d) = \sum_{i=1}^{n(d)} (Custo_{Aloca}(i) + Custo_{Despacha}(i)),$$

onde:

- d = período de planejamento;
- i = índice da placa a ser enviada ao LTQ;
- $n(d)$ = quantidade de placas a serem enviadas ao LTQ no período d ;



Figura 14: Visão lateral de um pátio, para mostrar que ao despachar a placa em verde será necessário a realização de 6 movimentações internas extras. Este corresponde ao pior caso

- $Custo_{Aloca}(i)$ = custo da alocação da placa i no pátio de estocagem;
- $Custo_{Despacha}(i)$ = custo do despacho ao LTQ da placa i ;
- $Custo_{Plano}(d)$ = custo total do plano de alocação e despacho para o período d .

Supondo que no período d as placas que serão enviadas para o LTQ, ou seja, que serão alocadas no pátio de estocagem e enviadas para laminação, são as de cor verde das Figuras Figura 12, Figura 13 e Figura 14. Supondo ainda que não foram necessários movimentos internos para a alocação das mesmas, o custo do plano deste período será:

$$Custo_{Plano}(d) = (0 + 0) + (0 + 2) + (0 + 6) = 8.$$

Para reforçar a idéia de como são calculados os custos na alocação e despacho das placas no pátio, segue um novo exemplo: suponha que um determinado pátio possua 2 alas e 1 coluna (ou seja, apenas 2 pilhas), e que é possível armazenar até 7 placas em uma mesma pilha. A Figura 15 apresenta a configuração inicial deste pátio, com as placas que nele estão alocadas e seus respectivos pesos. Como plano de entrada considera-se que existam 6 placas que deverão ser alocadas (Placa 9, Placa 10, Placa 11, Placa 12, Placa 13 e Placa 14), todas pesando 20 toneladas. Para alocar as Placas 9, 10 e 11, basta inserí-las na Ala 2, como ilustrado na Figura 16, pois nenhuma placa desta pilha tem peso inferior às que devem ser alocadas. Entretanto, para alocar as placas 12, 13 e 14, que deverão ser inseridas na Ala 1 dado que esta é a única que possui espaço no pátio, será realizada sempre a retirada da Placa 1 do topo da pilha, a inserção da nova placa e o retorno da Placa 1 para o topo. Esta retirada de placas já alocadas no pátio para a inserção de novas placas em uma determinada pilha conta, no cálculo do custo, como movimento interno extra realizado durante a alocação das placas. A alocação das Placas 12, 13 e 14 podem ser visualizadas, respectivamente, nas Figuras Figura 17, Figura 18 e Figura 19. Os custos das alocações de todas as placas do plano de entrada no pátio podem ser visualizados na Tabela 2.



Figura 15: Configuração inicial de um pátio fictício



Figura 16: Alocação das Placas 9, 10 e 11 no pátio fictício

Supondo que como plano de despacho teremos a seguinte seqüência de placas a serem enfiadas: Placa 3, Placa 2, Placa 1, Placa 9, Placa 10, Placa 11, Placa 12, Placa 13 e Placa 14. Tomando como base a configuração do pátio após a alocação de todas as placas (ilustrado na Figura 19) e o plano de despacho proposto, os custos referentes aos despachos das placas podem ser visualizados na Tabela 3.

Finalmente, o custo total do plano (custo de alocação somado ao custo de despacho) do período proposto será:

$$Custo_{Plano}(d) = (0 + 0 + 0 + 1 + 1 + 1) + (5 + 4 + 0 + 2 + 1 + 0 + 2 + 1 + 0) = 18.$$

3.2.6 A Heurística Busca Tabu aplicada ao Problema de Alocação de Placas de Aço

Nesta seção será apresentada a ferramenta desenvolvida para solucionar o problema de alocação das placas de aço utilizadas na fabricação de bobinas: a BTAlocaçãoPlacas. Este nome foi escolhido como simplificação de Heurística **B**usca **T**abu aplicada ao problema de **A**locação de **P**lacas de **A**ço.

A Figura 20 esquematiza os parâmetros de entrada e a saída da ferramenta desenvolvida (BTAlocaçãoPlacas), e estes serão detalhados posteriormente.



Figura 17: Alocação da Placa 12 no pátio fictício



Figura 18: Alocação da Placa 13 no pátio fictício

Esta ferramenta possui como parâmetros de entrada os seguintes dados:

- *Layout* do pátio (número de alas, colunas e camadas);
- Período de planejamento;
- Plano de placas a serem armazenadas;
- Plano de placas a serem laminadas;
- Configuração atual do pátio (placas e seus respectivos endereços já alocadas no pátio);
- Informações das Placas (identificador da placa, tipo da placa, peso da placa);
- Parâmetros do algoritmo BTAlocaçãoPlacas.

Os parâmetros do algoritmo são apresentados na Tabela 4, a seguir:

As placas a serem despachadas para o LTQ podem ser dos seguintes tipos:

1. Placas já produzidas e com previsão de despacho do pátio;
2. Placas que ainda não foram produzidas, mas com previsão de despacho do pátio;



Figura 19: Alocação da Placa 14 no pátio fictício

Placa	Custo da Alocação
Placa 9	0
Placa 10	0
Placa 11	0
Placa 12	1
Placa 13	1
Placa 14	1

Tabela 2: Custo da alocação em situação fictícia

3. Placas que se encontram no pátio, sem previsão de despacho no período considerado.

As placas referentes aos itens 1 e 2 são geradas com base em um plano de produção (Plano de Produção de Placas) e de despacho de placas (Plano de Placas para Laminação) e nas placas que já foram produzidas e não se encontram no pátio de despacho. Porém, o módulo responsável por gerar tais informações (Módulo Gerador de Placas Planejadas ilustrado na Figura 20) não será tratado neste trabalho.

Como saída da ferramenta é gerado um plano de alocação das placas no pátio, para o período planejado, bem como o custo deste plano. Uma outra possível saída da ferramenta são mensagens de erro.

Nos anexos B e C podem ser visualizados os formatos dos arquivos de entrada e saída da ferramenta, respectivamente.

A notação utilizada para os parâmetros da `BTAlocaçãoPlacas` segue:

- `NumeroMáximoIterações`: quantidade máxima de iterações que o algoritmo realizará até encontrar a solução final;
- `TempoTabu`: quantidade máxima de iterações que um determinado movimento é considerado tabu;

Placa	Custo de Despacho
Placa 3	5
Placa 2	4
Placa 1	0
Placa 9	2
Placa 10	1
Placa 11	0
Placa 12	2
Placa 13	1
Placa 14	0

Tabela 3: Custo do despacho em situação fictícia

Parâmetro	Notação
Número Máximo de Iterações do Algoritmo	NumeroMáximoIterações
Número de iterações que um determinado movimento permanece considerado como tabu	TempoTabu

Tabela 4: Parâmetros de entrada relacionados ao algoritmo BT

- **VetorSoluções**: vetor que armazena todas as soluções geradas com seus respectivos custos e o inverso dos movimentos que as originaram;
- **ListaTabu**: armazena os movimentos considerados tabu;
- **SoluçãoAtual**: armazena a solução que está sendo analisada;
- **CustoAtual**: armazena o custo da solução que está sendo analisada (custo de SoluçãoAtual);
- **MelhorSolução**: armazena a solução de menor custo encontrada até o momento;
- **MenorCusto**: armazena o custo da melhor solução encontrada até o momento;
- **AceitaMovimento**: variável de controle;

O algoritmo segue:

Algorithm 1 Busca Tabu aplicada ao problema de alocação de placas de aço no pátio de entrada do LTQ

```
1: //Inicialização
2: Inicialize NumeroMáximoIterações.
3: Inicialize TempoTabu.
4: IteraçãoAtual = 0.
5: Gere a solução inicial aleatoriamente ( $S_0 \in S$ ) para o pátio e atribua SoluçãoAtual
   com o valor de  $S_0$ .
6: Atribua CustoAtual com o valor do custo de SoluçãoAtual.
7: Atribua MelhorSolução com o valor de SoluçãoAtual e MenorCusto com o custo da
   MelhorSolução.
8: while IteraçãoAtual < NumeroMáximoIterações do
9:   // Geração de Soluções Vizinhas
10:  Gere um conjunto de soluções a partir da solução corrente e as armazene em Ve-
    torSoluções.
11:  Ordene VetorSoluções em ordem crescente de seus custos.
12:  AceitaMovimento = 0.
13:  // Escolha de Nova Solução Corrente
14:  while AceitaMovimento  $\neq$  1 do
15:    Pegue a primeira solução no VetorSoluções e verifique se o movimento é tabu.
16:    if movimento é tabu then
17:      if o movimento tabu proporciona a solução com o melhor custo encontrado até
        o momento then
18:        Atualiza os valores de MelhorSolução e MenorCusto com o valor desta solu-
          ção e seu custo, respectivamente.
19:        Atualiza os valores de SoluçãoAtual e CustoAtual.
20:        Atualiza a ListaTabu.
21:        AceitaMovimento = 1.
22:      else
23:        Busca a próxima solução no vetorSoluções.
24:      end if
25:    else
26:      Atualiza os valores de SoluçãoAtual e CustoAtual.
27:      Atualiza a ListaTabu.
28:      Inclui novo movimento na ListaTabu.
29:      AceitaMovimento = 1.
30:    end if
31:  end while
32: end while
```

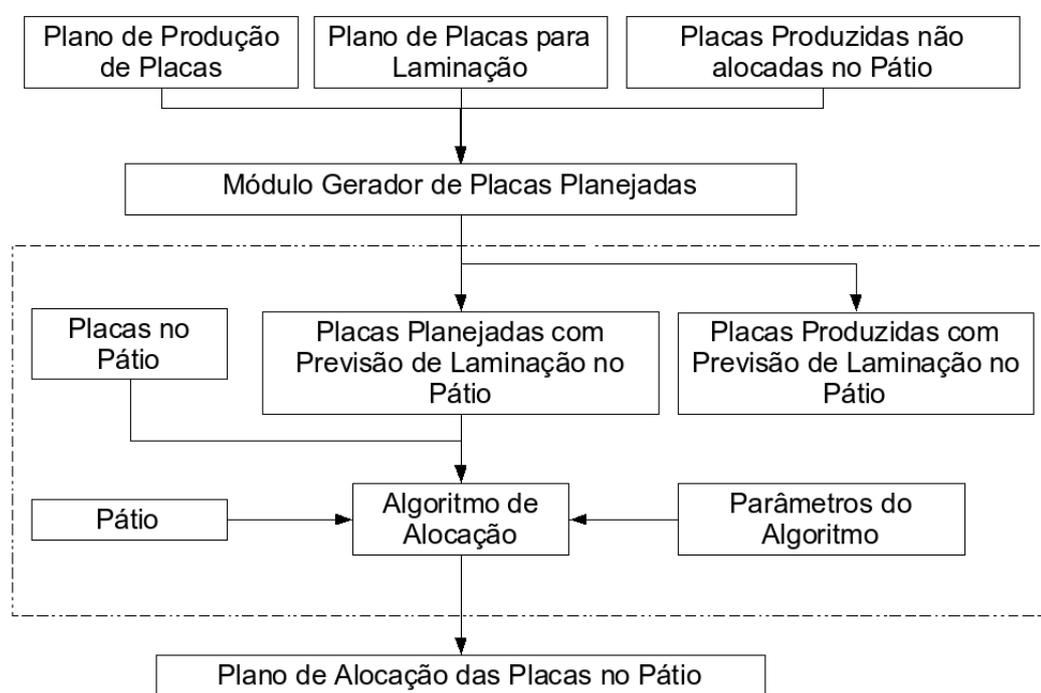


Figura 20: Esquema da ferramenta de geração de Plano de Alocação do Pátio

4 *Testes Computacionais*

*“Acredite em Alá, mas amarre
o seu camelo.”*

Provérbio Árabe

Este capítulo tem por objetivo demonstrar alguns resultados gerados pela ferramenta proposta, obtidos através de diferentes parametrizações aplicadas a uma mesma amostra de testes. Além disso, foram desenvolvidos três algoritmos para efeito de comparação com a ferramenta desenvolvida: o algoritmo aleatório, o algoritmo guloso e o algoritmo exato. O primeiro algoritmo realiza alocações aleatórias no pátio. Já o segundo, aloca a placa sempre na pilha que possui menos placas até o momento. E, por fim, o último encontra a solução de menor custo para o problema.

A ferramenta utilizada foi implementada na linguagem de programação Python, no ambiente Eric Python IDE, versão 3.9.

Todos os testes foram realizados utilizando uma amostra de testes com as seguintes características:

1. Pátio de dimensões 2 X 25 X 7, ou seja, 2 alas, 25 colunas e 7 camadas.
2. Um total de 80 placas a serem alocadas no pátio.
3. Um total de 80 placas a serem despachadas do pátio.

Essa massa de teste foi submetida à ferramenta desenvolvida, utilizando um computador pessoal com 1 Gb de memória RAM e Processador Athlon XP 3Ghz.

4.1 Testes

Durante a realização de testes numéricos, percebeu-se que, mesmo alterando o número de iterações máximas do algoritmo para valores superiores a 100, os resultados chegavam a um resultado muito próximo ao alcançado quando se utilizavam 100 iterações. Portanto, foi fixado o valor de 100 iterações como sendo o número máximo de iterações que o algoritmo realizará nesta bateria de testes.

O parâmetro que foi alterado em cada conjunto de testes foi o número de iterações que um determinado movimento permanece considerado como tabu. Para cada variação do parâmetro, o algoritmo foi executado 30 vezes, obtendo-se um valor máximo, mínimo, médio e um desvio padrão do custo da solução.

Seguem, nas Figuras Figura 21, Figura 22 e Figura 23 o resultado da seqüência de testes realizados com seus respectivos parâmetros de execução. Os valores que geraram estes gráficos estão listados no Anexo D.

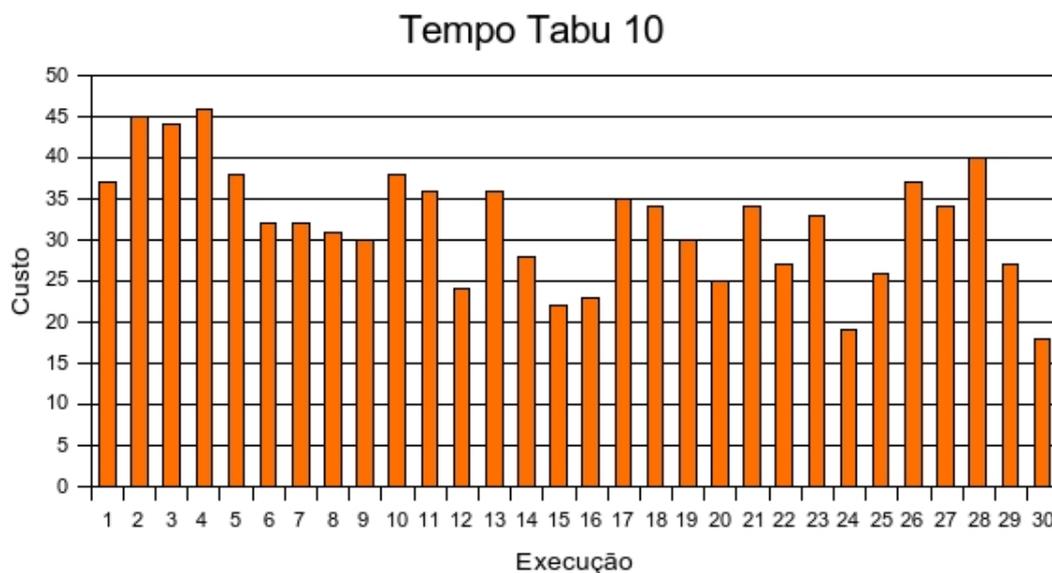


Figura 21: Gráfico correspondente ao Tempo Tabu 10

A partir destes resultados, foram calculadas as médias das soluções obtidas e o desvio padrão de cada cenário, conforme apresentado na Tabela 5.

Para efeito de comparação dos resultados obtidos pelo algoritmo BTAlocaçãoPlacas, foram desenvolvidos 3 outros algoritmos, que serão descritos posteriormente: o aleatório, o guloso e o exato.

O algoritmo aleatório realiza as alocações no pátio de forma aleatória, levando em

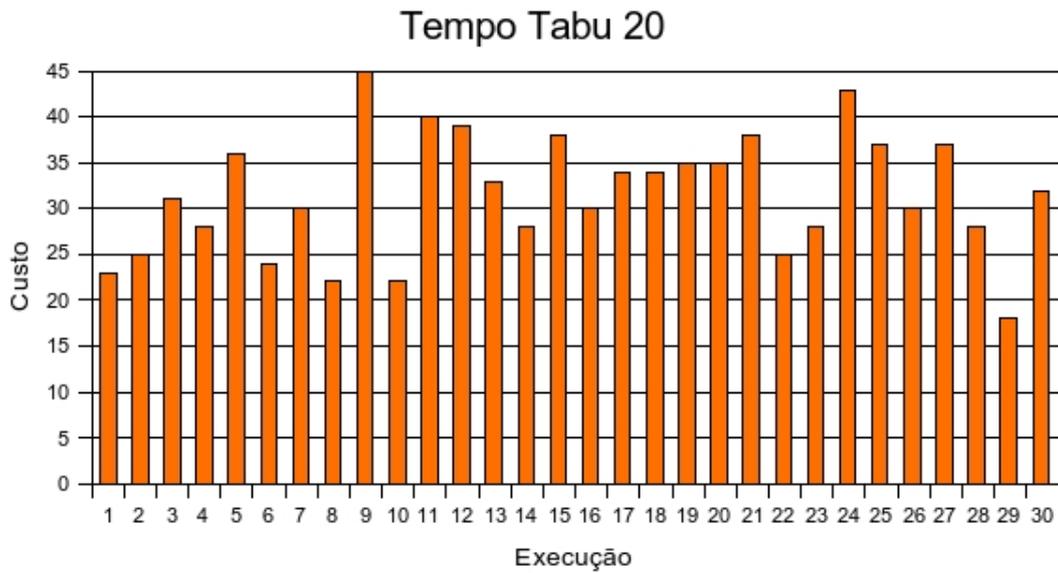


Figura 22: Gráfico correspondente ao Tempo Tabu 20

Tempo Tabu	Mínimo	Máximo	Médio	Desvio Padrão
10	18	46	32,03	7,24
20	18	45	31,6	6,64
30	17	42	29,83	7,64

Tabela 5: Custos Máximo, Mínimo, Médio e Desvio Padrão

consideração apenas o fato de ser possível ou não alocar a placa em uma determinada pilha. Foram realizadas 90 execuções deste algoritmo, e os resultados podem ser analisados na Figura 24. Os valores que geraram este gráfico estão listados no Anexo D.

Já o algoritmo guloso realiza as alocações no pátio objetivando alocar as placas nas pilhas que possuem menor quantidade de placas empilhadas até o momento. Para a massa de teste considerada, este algoritmo apresentou como resultado um custo de 76 movimentos internos durante a alocação e despacho das placas.

Finalmente, o algoritmo exato implementado é baseado na técnica *Branch and Bound*, e uma descrição deste bem como de sua implementação é apresentada no Anexo A.

Na Tabela 6 são apresentados os resultados obtidos nas parametrizações testadas do algoritmo BTAlocaçãoPlacas com os resultados obtidos a partir dos algoritmos aleatório e guloso.

Como observado anteriormente, o algoritmo BTAlocaçãoPlacas, em qualquer de suas parametrizações utilizadas, apresenta melhores resultados que uma forma aleatória de alocação das placas no pátio ou ainda que uma alocação baseada em uma estratégia

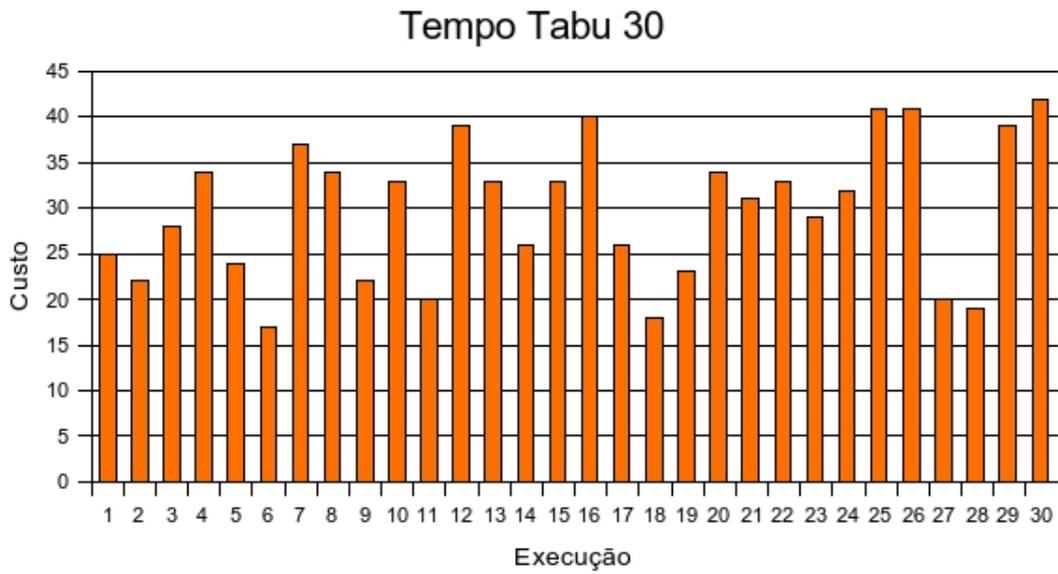


Figura 23: Gráfico correspondente ao Tempo Tabu 30

Tempo Tabu	Mínimo	Máximo	Médio	Desvio Padrão
10	18	46	32,03	7,24
20	18	45	31,6	6,64
30	17	42	29,83	7,64
aleatório	72	116	87,93	8,7
guloso	76	76	76	-

Tabela 6: Custos Máximo, Mínimo, Médio e Desvio Padrão de todos os algoritmos

gulosa. Pode-se observar, ainda, que o valor mínimo encontrado no algoritmo aleatório bem como o valor da solução gulosa são superiores ao valor máximo encontrado pelo algoritmo BTAlocaçãoPlacas.

Apesar do algoritmo exato obter a melhor solução, ou seja, a alocação de menor custo relativa ao problema, seu tempo de processamento é elevado e, às vezes, superior ao horizonte de planejamento da empresa, tornando-se uma proposta inviável para solução do problema. Os testes realizados entre os algoritmos BTAlocaçãoPlacas e exato são apresentados no Anexo A.

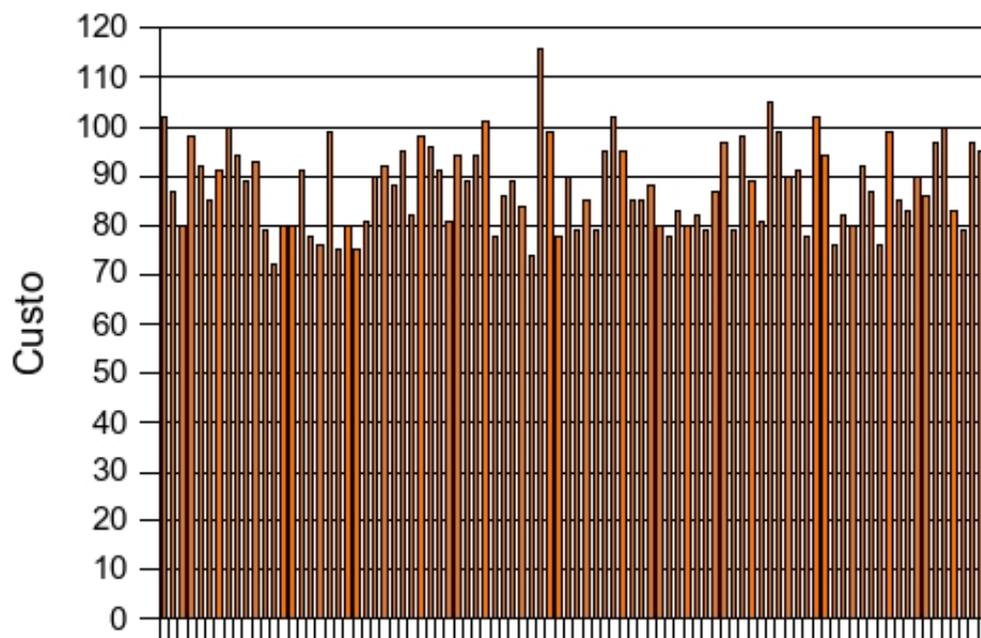


Figura 24: Gráfico correspondente ao Custo do Algoritmo Aleatório

5 *Conclusões e trabalhos futuros*

*“Não há nada de nobre
em sermos superiores ao próximo.
A verdadeira nobreza consiste em
sermos superiores ao que éramos antes.”*

Autor Desconhecido

No presente trabalho foi desenvolvida uma ferramenta para a geração de um plano de alocação de placas de aço em um pátio de despacho, minimizando o número de movimentações internas realizadas durante a alocação e o despacho das placas no mesmo. Os ganhos obtidos com a utilização desta ferramenta vão desde a redução do risco de acidentes até a redução dos custos de manutenção dos equipamentos utilizados neste processo. Uma importante contribuição, ainda, foi a modelagem do problema de alocação de placas de aço em pátios de estocagem.

Quanto à Busca Tabu, esta vem sendo utilizada com sucesso na resolução de vários problemas de otimização, como citado no capítulo 3. Neste trabalho foram aplicados os princípios básicos desta estratégia. O objetivo principal foi explorar as características desta heurística e demonstrar que ela pode gerar boas soluções quando bem aplicadas ao problema proposto.

Para o problema em questão foram realizadas comparações entre o BTAlocaçãoPlacas (a ferramenta desenvolvida), um algoritmo que faz a alocação das placas no pátio de forma aleatória e um algoritmo guloso, que objetiva alocar a placa na pilha que possui, até o momento, o menor número de placas. Neste contexto, o algoritmo proposto obteve melhores resultados para todas as parametrizações testadas.

Também foi desenvolvido, para efeito de comparação com a ferramenta, um algoritmo exato baseado na estratégia *Branch and Bound*. Este mostrou-se inviável na resolução do problema devido ao grande tempo de processamento necessário para a obtenção de soluções grandes. Entretanto, sua modelagem é um importante resultado deste trabalho.

É importante destacar, ainda, que outros algoritmos podem ser utilizados para a resolução do problema, e a implementação destas novas abordagens pode ser considerada como trabalho futuro.

Como outros trabalhos futuros e possibilidades de melhorias, podem-se citar a melhoria da interface da ferramenta com o usuário e a inclusão de novas restrições ao problema, como exemplo, o ajuste da ferramenta para considerar também as restrições referentes às resistências das placas de aço.

Por fim, conclui-se que o algoritmo implementado pôde contribuir com o mercado industrial no sentido de apresentar uma possibilidade de solução para um problema complexo.

Referências

- [Amorim 2006]AMORIM, W. P. *Sistema de Alocação em Pátios de Estocagem: Uma Aplicação no Armazenamento de Bobinas de Aço*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2006.
- [Costa e Silva 2006]COSTA, A. L. V.; SILVA, P. R. M. *Aços e Ligas Especiais*. São Paulo, Brasil: Edgard Blucher, 2006.
- [Eastman 1959]EASTMAN, W. A solution to the travelling salesman problem. *Econometrica*, n. 27, p. 282–296, 1959.
- [Gambardella et al. 2001]GAMBARDELLA, L. M. et al. An optimization methodology for intermodal terminal management. *Journal of Intelligent Manufacturing*, n. 12, p. 521–534, 2001.
- [Glover e Laguna 1997]GLOVER, F.; LAGUNA, F. *Tabu Search*. Norwell, MA, USA: Kluwer Academic Publishers, 1997. ISBN 079239965X.
- [Instituto Brasileiro de Siderurgia 2007]Instituto Brasileiro de Siderurgia. *IBS*. 2007. URL:<http://www.ibs.org.br>. Último acesso: 30/08/2007.
- [Kim e Kim 1999]KIM, K. H.; KIM, H. B. Segregating space allocation models for containers inventories in port container terminals. *Int. J. Production Economics*, n. 59, p. 415–423, 1999.
- [Land e Doig 1960]LAND, A.; DOIG, A. An automatic method for solving discrete programming problems. *Econometrica*, n. 28, p. 497–520, 1960.
- [OBrien 2001]OBRIEN, J. A. *Sistemas de Informação e as Decisões Gerenciais na Era da Internet*. [S.l.]: Saraiva, 2001.
- [Oliveira 2006]OLIVEIRA, D. C. *Aplicação da Busca Tabu na Alocação de Produtos Granelados em Terminais Portuários de Importação*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2006.
- [Papadimitriou e Steiglitz 1982]PAPADIMITRIOU, C. H.; STEIGLITZ, K. *Combinatorial optimization: algorithms and complexity*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1982.
- [Pinedo 1995]PINEDO, M. *Scheduling: theory, algorithms and systems*. Englewood Cliffs: Prentice-Hall, 1995.
- [Preston e Kozan 1999]PRESTON, P.; KOZAN, E. Genetic algorithms to schedule container transfers at multimodal terminals. *International Transactions in Operational Research*, 1999.

-
- [Preston e Kozan 2001]PRESTON, P.; KOZAN, E. An approach to determine storage locations of containers at seaport terminals. *Computers and Operations Research*, v. 28, n. 10, p. 983–995, 2001.
- [Ryan, Bailey e Moore 1998]RYAN, J. L.; BAILEY, T. G.; MOORE, J. T. Reactive tabu search in unmanned aerial reconnaissance simulations. *Proceeding of the 1998 Winter Simulation Conference*, 1998.
- [Schepke et al. 2004]SCHEPKE, C. et al. Avaliação de heurísticas de melhoramento e da metaheurística busca tabu para soluções de prv. *VII Fórum de Tecnologia e XIV Simpósio Regional de Informática*, 2004.

ANEXO A - Algoritmo Branch and Bound

O algoritmo *Branch and Bound* é um dos algoritmos mais utilizados na obtenção da solução ótima de problemas de otimização combinatória. Ele foi inicialmente desenvolvido e utilizado no contexto de problemas de programação inteira mista [Land e Doig 1960] e do problema *Travelling Salesman Problem* (TSP) [Eastman 1959].

A técnica *Branch and Bound* consiste basicamente em um método exato enumerativo que explora apenas as regiões do espaço de soluções onde existe a possibilidade de uma solução ótima ser encontrada.

A chave para esta técnica é utilizar um bom critério para decidir se uma região do espaço de soluções pode conter uma solução ótima. O que tal critério faz é resolver uma relaxação do problema, considerando o conjunto de decisões tomadas até o momento na construção da solução, obtendo um *lower bound* (no caso de um problema de minimização) que indica que as soluções daquela região não possuem valor inferior ao mesmo, e então comparar este valor com o valor da função objetivo da melhor solução encontrada até o momento da busca (*upper bound*). Caso o *lower bound* seja menor que o *upper bound*, então o algoritmo continua explorando aquela região em busca da solução ótima (*branch*), caso contrário o algoritmo não segue na exploração da região considerada (*bound*). Seguindo este *framework* é fundamental então trabalhar com boas técnicas de obtenção tanto de *lowers* quanto de *uppers bound*. Além disto é preciso decidir de que forma o espaço de soluções é explorado durante o algoritmo, ou seja, como é feito o particionamento do mesmo e em qual ordem as regiões serão exploradas [Papadimitriou e Steiglitz 1982].

A.1 Exploração do Espaço de Soluções

A execução do procedimento *branch* pode ser representada por uma árvore de busca, na qual o problema original corresponde ao nó raiz e os subproblemas de um problema aos nós filhos.

Para realizar a exploração do espaço de soluções, deve-se manter uma lista de nós não processados, correspondente aos subproblemas ainda não eliminados, ou àqueles cujos subproblemas não foram gerados ainda. Deve-se, ainda, armazenar o valor da melhor solução encontrada até o presente momento. Este valor servirá como limite superior para as novas soluções que serão analisadas.

A.2 Lower Bounds

Um *lower bound* de um dado nó da árvore de busca do *Branch and Bound* para um problema de minimização pode ser entendido como uma estimativa do menor valor possível que pode ser atingido por uma solução, dado que um conjunto de decisões sobre a resolução do problema já tenham sido tomadas. Então, o *lower bound* é sempre menor ou igual ao valor da melhor solução possível, dadas as decisões já tomadas. Esta estimativa é interessante porque permite ao algoritmo deixar de explorar certos ramos da árvore de busca, tendo a certeza de que não haverá perda de soluções melhores que as já encontradas.

A.3 Solução viável inicial (*Upper Bound* Inicial)

Como já discutido anteriormente, é muito importante para o desempenho do *branch and bound* trabalhar com um *upper bound* que tenha valor o mais próximo do valor da função objetivo da solução ótima. Um *upper bound*, para um problema de minimização, nada mais é que o valor da função objetivo de alguma solução viável conhecida. Portanto, para obtermos um bom *upper bound*, basta gerar uma solução viável de boa qualidade através de algum método heurístico.

A.4 O algoritmo *Branch and Bound* aplicado ao Problema de Alocação de Placas de Aço

Com o objetivo de obter um novo algoritmo para efeito de comparação com a ferramenta `BTAlocaçãoPlacas`, foi desenvolvido um algoritmo baseado no método *Branch and Bound*: o algoritmo exato.

Na implementação do algoritmo, a estratégia adotada para a exploração do espaço de soluções consistiu em percorrer a árvore de busca, de forma recursiva, utilizando busca em profundidade. Já o *upper bound* inicial foi alcançado utilizando o método guloso descrito

em seções anteriores.

A.4.1 Cálculo do *Lower Bound*

O cálculo do *lower bound* para o problema proposto é obtido através do plano de placas a serem armazenadas, do plano de placas a serem laminadas e da configuração atual do pátio.

Inicialmente, é criada uma lista de tarefas a serem realizadas a partir dos planos de armazenamento e despacho fornecidos para um determinado período. Para cada tarefa desta lista deverá ser realizada uma estimativa do menor valor possível que pode ser atingido na realização de tal tarefa, dado que um conjunto de decisões sobre a resolução do problema já tenham sido tomadas, ou seja, que um determinado número de alocações e despachos da lista de tarefas já tenham sido realizados.

Com o intuito de simplificar a idéia de como é calculado o *lower bound* das tarefas, este será apresentado em paralelo ao exemplo a seguir: suponha que um determinado pátio é constituído de apenas três pilhas: P1, P2 e P3, cada uma podendo conter até 7 placas empilhadas cada. Considere, ainda, a existência de 3 tipos de placas diferentes: A, B e C, sendo A a placa de menor peso e C a de maior peso. O estado inicial deste pátio é ilustrado na Figura 25.

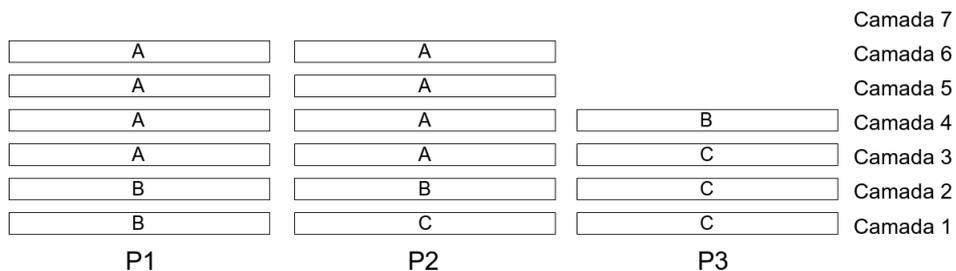


Figura 25: Estado inicial de um pátio

Assuma, como notação para a lista de tarefas, que X representa a ação de alocar uma placa de peso X no pátio e X' a ação de despachar uma placa de peso X do pátio. Uma possível seqüência de tarefas a serem realizadas no pátio em um determinado período segue:

$$Tarefas = [A, B, A, C, C', B']$$

Como discutido anteriormente, para o cálculo do *lower bound* é necessário definir qual seria o menor custo possível para realizar todas as tarefas da lista de tarefas dada a

configuração atual do pátio.

Inicialmente, serão computados quais seriam os menores custos de alocar as placas mais pesadas da lista de tarefas, ou seja, os custos das alocações das placas que possivelmente serão armazenadas nas camadas inferiores das pilhas.

Voltemos ao exemplo. As placas mais pesadas, neste caso, são as placas de peso C. Tomemos, como ponto de partida, a tarefa alocar a placa de peso C da lista de tarefas. O primeiro passo no cálculo do custo desta tarefa é obter a informação de qual seria a melhor pilha para realizar determinada alocação. O custo da alocação da placa em cada uma das pilhas do pátio é apresentado na Tabela 7. Como pode ser observado, a melhor opção de alocação da placa C é na pilha P3.

Pilha	Custo da Alocação
P1	6
P2	5
P3	1

Tabela 7: Custo da alocação da placa C em cada pilha do pátio

Após a decisão da pilha em que será alocada a placa, deve-se realizar um levantamento do número de posições livres existentes em outras pilhas do pátio. No melhor caso para alocação de uma determinada placa, todas as placas com menor peso que foram alocadas anteriormente, a partir do vetor de tarefas, foram alocadas em outras pilhas do pátio.

Um possível problema ocorre nos casos em que não existem posições livres suficientes para realizar todas as alocações das placas mais leves nas outras pilhas. Neste caso, placas de menor peso teriam sido alocadas na pilha em que será alocada também a placa em questão e, como possuem menor peso, seriam realizados movimentos internos extras para a alocação da placa mais pesada, aumentando o custo desta solução.

O cálculo do custo de realizar uma determinada tarefa é dado por:

$$Custo = custo_{(p,P)} + numPlacasMenorPeso_{(p,P)},$$

onde:

- $custo_{(p,P)}$ = custo de alocar ou desalocar a placa p na pilha P ;
- $numPlacasMenorPeso_{(p,P)}$ = número de placas de menor peso do que p que foram alocadas na pilha P .

O $numPlacasMenorPeso_{(p,P)}$ é dado por

$$numPlacasMenorPeso_{(p,P)} = numPlacasMenorPesoAlocadas_{(p)} - numEspacos(P),$$

onde:

- $numPlacasMenorPesoAlocadas_{(p)}$ = número de placas alocadas anteriormente que possuem menor peso do que p e que ainda não foram desalocadas;
- $numEspacos(P)$ = numero de espaços livres existentes nas pilhas diferentes de P no pátio.

Caso o $numPlacasMenorPeso_{(p,P)}$ for um número negativo, será utilizado o valor zero no cálculo do Custo.

Finalmente, voltemos ao exemplo. Para calcular o custo de alocar o C, deve-se levantar o número de placas já alocadas que possuem menor peso do que C na lista de tarefas. Neste caso, temos as alocações das placas A, B e A, respectivamente, primeira, segunda e terceira tarefa da lista de tarefas. Como só existem 2 espaços livres nas outras pilhas do pátio, uma destas placas certamente foi alocada na pilha P3. Portanto, o custo de alocar o C é dado por:

$$Custo = 1 + (3 - 2) = 2$$

Agora, após a obtenção do menor custo de alocar a placa C no pátio, deverão ser levantados, seguindo o mesmo princípio de cálculo, todos os custos das outras tarefas. Inicia-se pela obtenção do custo de alocação de outros Cs. Em seguida, devem ser levantados todos os custos de alocação dos Bs. Já os custos dos As serão sempre zero, pois estas placas, por serem as mais leves, estarão sempre nos topos das pilhas. Finalizando, mas ainda seguindo o mesmo princípio de cálculo, deverão ser calculados os custos dos despachos dos Cs e dos Bs. Após o cálculo de todos os custos de todas as tarefas da lista, estes deverão ser somados, formando assim o *lower bound*.

Portanto, o *lower bound* será dado, no exemplo, por:

$$lower\ bound = \text{custo de alocar todos os Cs} + \text{custo de alocar todos os Bs} + \text{custo de desalocar todos os Cs} + \text{custo de desalocar todos os Bs}$$

A notação utilizada para os parâmetros do algoritmo de cálculo do *lower bound* segue:

- totalLB = armazena o custo do *lower bound*;

- MAX_PLACAS_PILHA = número máximo de placas que podem ser empilhadas em uma mesma pilha;
- listaTarefas = armazena todas as tarefas a serem realizadas;
- menor_custo = armazena o menor custo dentre os custos de alocar ou desalocar uma placa em uma pilha;
- p_menor_custo = armazena o identificador da pilha do pátio em que a alocação ou despacho de uma placa proporcionou o menor custo dentre todos;
- custo_anteriores = número de placas de peso menor que a placa que está sendo alocada ou desalocada no momento, que foram alocadas na p_menor_custo;
- lb = custo do *lower bound* correspondente à tarefa que está sendo realizada.

O algoritmo de cálculo do *lower bound* segue:

Algorithm 2 Cálculo do Lower Bound

```

1: totalLB = 0
2: for all tarefa in listaTarefas: do
3:   menor_custo = MAX_PLACAS_PILHA + 1
4:   for all pilha in pátio: do
5:     if É viável realizar tal tarefa na pilha: then
6:       custo = custo de realizar tarefa na pilha
7:       if custo < menor_custo: then
8:         menor_custo = custo
9:         p_menor_custo = p
10:      end if
11:    end if
12:    calcule o número de placas que também serão alocadas na p_menor_custo e
    atribua este valor a custo_anteriores
13:    lb = menor_custo + custo_anteriores
14:    totalLB = totalLB + lb
15:  end for
16:  return totalLB
17: end for

```

A.4.2 Testes

Foram realizados vários testes comparativos entre os algoritmos BTAlocaçãoPlacas, guloso e exato. Foram realizadas, para cada algoritmo proposto, 3 execuções, e o melhor resultado obtido será apresentado.

No algoritmo BTAlocaçãoPlacas, foi fixado como 100 iterações o número máximo de iterações que o algoritmo realizará e 10 iterações do algoritmo, o tempo que um determinado movimento permanece considerado como tabu.

Todos os testes foram realizados utilizando-se amostras aleatórias. As parametrizações dos testes são apresentadas na Tabela 8. Os tempos de processamento de cada teste são apresentados na Tabela 9. E, por fim, os custos obtidos são apresentados na Tabela 10.

Teste	Entradas	Saídas	Placas no Pátio	Pilhas no Pátio
1	4	4	21	6
2	4	8	7	5
3	11	7	25	7
4	7	6	25	7
5	7	27	25	7
6	7	20	23	6
7	14	17	7	5

Tabela 8: Parametrizações dos testes realizados

Teste	Tempo Guloso	Tempo BTAlocaçãoPlacas (seg)	Tempo Exato (seg)
1	0,00	3,11	0,02
2	0,00	1,86	0,10
3	0,00	16,64	35,87
4	0,00	12,88	82,74
5	0,00	24,09	1070,52
6	0,00	22,96	3,40
7	0,00	37,55	3557,90

Tabela 9: Tempos obtidos

Teste	Custo Guloso	Custo BTAlocaçãoPlacas	Custo Exato
1	5	2	2
2	6	3	3
3	6	4	4
4	14	10	10
5	40	27	27
6	30	22	21
7	24	9	9

Tabela 10: Resultados obtidos

Seguem, nas Figuras Figura 26 e Figura 27 os gráficos dos tempos e dos custos, respectivamente, resultantes da seqüência de testes realizados.

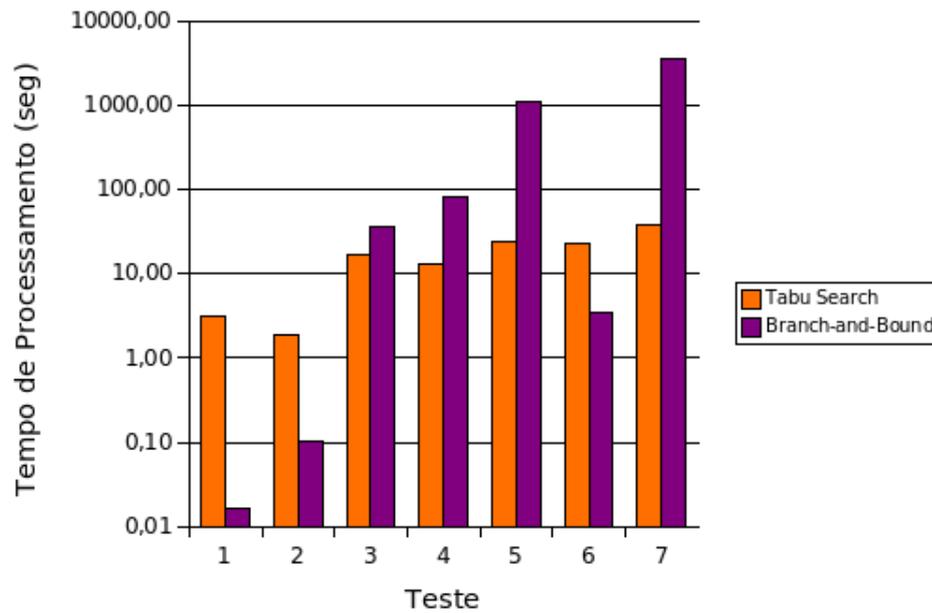


Figura 26: Gráfico correspondente aos tempos obtidos

Pode-se perceber, através dos resultados apresentados, que o algoritmo exato implementado através da metodologia *Branch and Bound* apresenta uma grande variação no tempo de processamento, variando desde 0,02 segundos até quase 1 hora de processamento, e obtendo, praticamente, o mesmo resultado do algoritmo BTALocaçãoPlacas. Apenas nos testes 1 e 6 o algoritmo exato encontrou a solução em tempo inferior ao algoritmo BTALocaçãoPlacas.

Os testes realizados foram todos testes com amostras pequenas, utilizados apenas como ferramenta de comparação entre os algoritmos. Testes com amostras maiores demandariam um tempo elevado de processamento do algoritmo exato, não sendo viáveis.

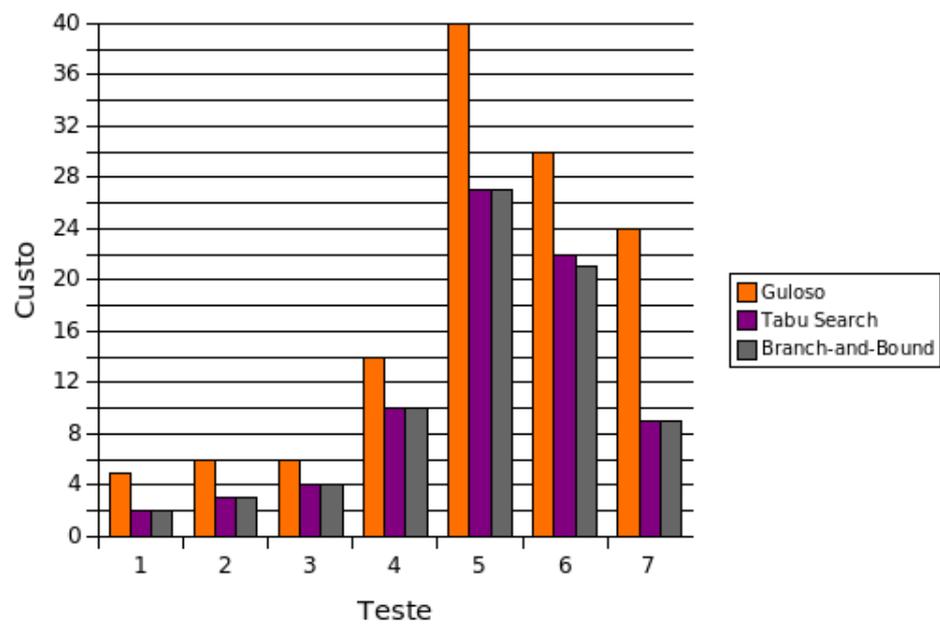


Figura 27: Gráfico correspondente aos custos obtidos

ANEXO B – Layout dos Arquivos de Entrada

- Arquivo: Parametrização do Algoritmo BTAlocaçãoPlacas

Nome: BTAlocaPlacas.txt

Conteúdo: contém as informações básicas das variáveis de parametrização do algoritmo.

Formato: apresentado na Tabela 11

Descrição	Formato
Número Máximo de Iterações do Algoritmo	NumeroMáximoIterações = <valor inteiro positivo>
Número de iterações que um determinado movimento permanece considerado como tabu	TempoTabu = <valor inteiro positivo>
Número de alas do pátio	alas = <valor inteiro positivo>
Número de colunas do pátio	colunas = <valor inteiro positivo>
Número de camadas do pátio	camadas = <valor inteiro positivo>
Plano de placas a serem armazenadas	plano_entrada = <vetor de strings>
Plano de placas a serem laminadas	plano_saida = <vetor de strings>
Configuração atual do pátio	placas_patio = <vetor de vetor strings>

Tabela 11: Parâmetros de entrada relacionados ao algoritmo BT

- Arquivo: Placas

Nome: Placas.txt

Conteúdo: contém as informações básicas sobre as placas a serem trabalhadas. Cada linha corresponde a uma única placa.

Formato: apresentado na Tabela 12

Descrição	Formato
Identificação da Placa	id_Placa = <valor inteiro positivo>
Tipo da Placa	tipo_Placa = <valor inteiro positivo>
Peso da Placa	peso_Placa = <valor inteiro positivo>

Tabela 12: Parâmetros de entrada relacionados às placas

ANEXO C – Layout dos Arquivos de Saída

- Arquivo: Plano de Alocação

Nome: planoAlocacao.txt

Conteúdo: contém um plano de alocação das placas no pátio para um determinado período de planejamento especificado em um arquivo de entrada.

Formato: apresentado na Tabela 13

Descrição	Formato
Plano de alocação das placas no pátio	plano_alocacao = <vetor de pilhas>
Custo do plano de alocação	custo_plano = <valor inteiro positivo>

Tabela 13: Parâmetros de entrada relacionados ao algoritmo BT

O plano de alocação das placas é apresentado como um vetor de pilhas. Cada pilha no pátio possui as informações ala e coluna.

Cada elemento do vetor de saída indica em qual pilha deve ser alocada a placa correspondente a mesma posição do vetor de entrada no pátio (plano de placas a serem armazenadas).

ANEXO D - Dados obtidos nos Testes

Número da Execução	Custo
1	37
2	45
3	44
4	46
5	38
6	32
7	32
8	31
9	30
10	38
11	36
12	24
13	36
14	28
15	22
16	23
17	35
18	34
19	30
20	25
21	34
22	27
23	33
24	19
25	26
26	37
27	34
28	40
29	27
30	18

Tabela 14: Resultados obtidos com tempo tabu 10

Número da Execução	Custo
1	23
2	25
3	31
4	28
5	36
6	24
7	30
8	22
9	45
10	22
11	40
12	39
13	33
14	28
15	38
16	30
17	34
18	34
19	35
20	35
21	38
22	25
23	28
24	43
25	37
26	30
27	37
28	28
29	18
30	32

Tabela 15: Resultados obtidos com tempo tabu 20

Número da Execução	Custo
1	25
2	22
3	28
4	34
5	24
6	17
7	37
8	34
9	22
10	33
11	20
12	39
13	33
14	26
15	33
16	40
17	26
18	18
19	23
20	34
21	31
22	33
23	29
24	32
25	41
26	41
27	20
28	19
29	39
30	42

Tabela 16: Resultados obtidos com tempo tabu 30

Número da Execução	Custo	Número da Execução	Custo	Número da Execução	Custo
1	102	31	91	61	87
2	87	32	81	62	97
3	80	33	94	63	79
4	98	34	89	64	98
5	92	35	94	65	89
6	85	36	101	66	81
7	91	37	78	67	105
8	100	38	86	68	99
9	94	39	89	69	90
10	89	40	84	70	91
11	93	41	74	71	78
12	79	42	116	72	102
13	72	43	99	73	94
14	80	44	78	74	76
15	80	45	90	75	82
16	91	46	79	76	80
17	78	47	85	77	92
18	76	48	79	78	87
19	99	49	95	79	76
20	75	50	102	80	99
21	80	51	95	81	85
22	75	52	85	82	83
23	81	53	85	83	90
24	90	54	88	84	86
25	92	55	80	85	97
26	88	56	78	86	100
27	95	57	83	87	83
28	82	58	80	88	79
29	98	59	82	89	97
30	96	60	79	90	95

Tabela 17: Resultados obtidos com o algoritmo Aleatório

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)