

UNIVERSIDADE FEDERAL DO MARANHÃO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

**RAPHAEL GOMES SANTOS**

**OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS APLICADA AO  
PROBLEMA DE ALOCAÇÃO DE BERÇOS**

São Luís

2013

**RAPHAEL GOMES SANTOS**

**OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS APLICADA AO  
PROBLEMA DE ALOCAÇÃO DE BERÇOS**

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Alexandre Cesar Muniz de Oliveira

São Luís

2013

Santos, Raphael Gomes.

Otimização por colônia de formigas aplicada ao problema de alocação de berços/  
Raphael Gomes Santos. – São Luís, 2013.

48 f.

Impresso por computador (fotocópia).

Orientador: Alexandre César Muniz de Oliveira.

. Monografia (Graduação) – Universidade Federal do Maranhão, Curso de Ciência da Computação, 2013.

1. Alocação de berços. 2. Metaheurística. 3. Colônia de formigas. I. Título.

CDU 004.451.9

**RAPHAEL GOMES SANTOS**

**OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS APLICADA AO  
PROBLEMA DE ALOCAÇÃO DE BERÇOS**

Monografia apresentada ao curso de  
Ciência da Computação da Universidade  
Federal do Maranhão, como parte  
dos requisitos necessários para obtenção  
do grau de Bacharel em Ciência da  
Computação.

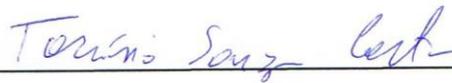
Aprovada em 11 de dezembro de 2013

**BANCA EXAMINADORA**



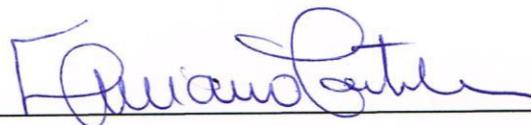
---

Prof. Dr. Alexandre César Muniz de Oliveira  
Departamento de Informática  
Universidade Federal do Maranhão



---

Prof. MSc. Tarcísio Souza Costa  
Departamento de Educação Profissional - Campus Pinheiro  
Instituto Federal de Educação, Ciência e Tecnologia do Maranhão



---

Prof. Dr. Luciano Reis Coutinho  
Departamento de Informática  
Universidade Federal do Maranhão

*Aos meus pais.*

## AGRADECIMENTOS

Agradeço a Deus por ter me dado a vida e todas as circunstâncias que nela vivi e me fizeram ser quem eu sou.

Agradeço aos meus pais pelo amor, cuidado e provisão de sempre, bem como pela educação que me deram.

Ao meu orientador Alexandre Cesar Muniz de Oliveira, por todas as oportunidades que me deu na direção de melhorar minha formação. E pela orientação durante o processo de pesquisa.

Ao corpo docente do curso de Ciência da Computação, pelo tempo dedicado ao ensino e pesquisa durante a graduação.

Aos colegas de graduação, Tiago, Busson, Paulo e Marcelo.

Ao pessoal do Laboratório de Aprendizado Computacional e Métodos de Otimização – LACMO, Valdecy, Tarcísio, Aécio e Igor.

E a todos que contribuíram para a minha formação.

*“Arriscar-se no sentido mais amplo é  
precisamente tomar consciência de si  
próprio.”*

*Soren Kierkegaard*

## RESUMO

O Problema da Alocação de Berços (PAB) em portos graneleiros consiste em determinar qual a melhor ordem que os navios devem atracar em cada berço de forma que o custo seja o mínimo possível, tal custo pode ser mensurado de várias maneiras, tais como tempo total de operação ou multa por sobrestada. O PAB pode ser visto como um problema de otimização combinatória da classe NP-difícil, tornando inviável a solução de grandes instâncias desse problema por métodos exatos.

Metaheurísticas são métodos de otimização de caráter geral que são eficientes e garantem boas soluções. A Otimização por Colônia de Formigas é uma metaheurística que tem se mostrado promissora e competitiva na literatura.

O objetivo geral deste trabalho é resolver o PAB com um método baseado no esquema geral da metaheurística Otimização por Colônia de Formigas.

**Palavras-chave:** Alocação de Berços. Metaheurística. Colônia de Formigas.

## **ABSTRACT**

The Berth Allocation Problem (BAP) in bulk ports is to determine the best order that ships must dock at each berth, so that the cost is minimized, this cost can be measured in several ways, such as total time of operation or demurrage. The BAP can be viewed as a combinatorial optimization problem of NP-hard class, making unfeasible the solution of large instances of this problem by exact methods.

Metaheuristics are optimization methods of general character that are efficient and guarantee good solutions. Ant Colony Optimization is a metaheuristic that has shown promise and competitive in the literature.

The general objective of this work is to solve the BAP with a method based on the general scheme of metaheuristic Ant Colony Optimization.

**Keywords:** Berth allocation. Metaheuristic, Ant Colony.

## LISTA DE FIGURAS

Figura 1: Experimento da ponte dupla. ....	18
Figura 2: Comportamento das formigas, depois de um tempo no experimento da ponte dupla. ....	19
Figura 3: Convergência das formigas para o caminho mais curto. ....	19
Figura 4: Pseudocódigo da metaheurística ACO. ....	24
Figura 5: Cenário típico de um porto graneleiro. Fonte: (BARROS, 2009, pp. 13). ....	30
Figura 6: PAB modelado como um problema de transporte. Fonte: (BARROS, 2009, pp. 33). ....	32
Figura 7: Procedimento que deriva $x_{ij}$ de $u_{il}$ . ....	38
Figura 8: Produto cartesiano $N \times L$ . ....	38
Figura 9: Transformação de um percurso no grafo para uma matriz de navios $\times$ berços. Os vértices marcados são os vértices visitados. ....	39

## LISTA DE TABELAS

Tabela 1: Mostra a configuração dos parâmetros do algoritmo .....	41
Tabela 2: Resultados obtidos com a implementação do primeiro tipo de informação heurística proposto .....	43
Tabela 3: Resultados obtidos com a implementação do segundo tipo de informação heurística proposto .....	44

## SUMÁRIO

1	INTRODUÇÃO .....	14
2.	OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS .....	17
2.1.	Inspiração Biológica .....	17
2.2.	A Metaheurística ACO .....	20
2.2.1.	Representação do Problema .....	20
2.2.2.	Comportamento das formigas .....	21
2.2.3.	A Metaheurística .....	23
2.2.4.	Aplicação ao problema .....	24
2.3.	Variações da ACO .....	25
2.3.1.	Elitist Ant System (EAS) .....	25
2.3.2.	Rank-Based Ant System ( $AS_{rank}$ ) .....	26
2.3.3.	Ant Colony System (ACS) .....	27
3.	PROBLEMA DE ALOCAÇÃO DE BERÇOS .....	29
3.1.	Modelagem Matemática .....	30
3.1.1.	Dados de entrada .....	32
3.1.2.	Função Objetivo .....	33
3.1.3.	Restrições .....	33
4.	PROPOSTA .....	35
4.1.	Modelagem computacional do PAB .....	35
4.1.1.	Decisões de projeto .....	35
4.2.	ACO aplicada aplicado ao PAB .....	36
4.2.1.	Função Objetivo .....	36
4.2.2.	Representação .....	37
4.2.3.	Restrições .....	39
4.2.4.	Informação Heurística .....	40
4.2.5.	Definição dos parâmetros .....	41
5.	RESULTADOS .....	42
5.1.	Ambiente de testes .....	42
5.2.	Instancias .....	42
5.3.	Descrição dos resultados .....	42
6.	CONCLUSÃO .....	45
6.1.	Contribuições .....	45

6.2. Trabalhos Futuros .....	46
7. REFERÊNCIAS BIBLIOGRÁFICAS .....	47

# 1 INTRODUÇÃO

Nos últimos anos, importantes contribuições têm sido dadas no campo das metaheurísticas de busca. Isto porque, em razão da grande quantidade de aplicações científicas e industriais, problemas de otimização têm ganhado importância fundamental. No entanto, encontrar soluções ótimas para esses problemas pode ser uma tarefa difícil para diversos casos. Visto que, diversos problemas de otimização têm a característica de ter suas complexidades acrescidas consideravelmente em função das suas características. Por exemplo, há problemas de otimização nos quais uma instância grande o suficiente implica em que o tempo necessário para encontrar a solução se torna absurdamente grande, tornando a busca por soluções impraticável. Assim, as metaheurísticas de busca surgem como uma boa alternativa, uma vez que estas são métodos de busca de complexidade satisfatória, que encontram soluções boas para problemas de otimização.

## 1.1 Motivação

A importância dos portos cresce com os avanços tecnológicos na construção de grandes navios. A maior capacidade de navegação resulta em um número maior de operações de carregamento e descarregamento realizados por um navio. Além disso, o transporte marítimo é invariavelmente mais intensificado entre as nações devido à globalização e ao crescimento econômico mundial. O porto é uma porta de entrada, uma abertura para a constituição de negócios, criando um amplo campo de oportunidades industriais e comerciais (SILVA, 1999 apud SILVA e COELHO; SILVA e COELHO, 2007). Devido a estes fatores a demanda pelos serviços portuários aumentou consideravelmente.

Conforme Silva (1999, apud SILVA e COELHO, 2007) os portos devem adotar uma nova função que é a de organizar e gerenciar fluxos contínuos, buscando otimizar a eficácia interna, visto como instrumento capaz de fazer circular fluxos de mercadorias cada vez maiores em um tempo cada vez menor, e com custos sempre mais baixos e competitivos. Tendo em vista esses desafios, surge a necessidade de aproveitar ao máximo os recursos disponíveis nos portos. Usar os recursos de forma inteligente é fundamental porque, apesar das reformas e aplicações serem mecanismos adotados corriqueiramente pelas administrações portuárias, com o

objetivo de oferecer mais recursos e agilidade às operações com navios, é fácil concluir que os portos dificilmente serão projetados para disporem de recursos acima da demanda exigida.

No Maranhão, existe o complexo portuário marítimo de São Luís, composto por três portos: Porto do Itaqui, Terminal Marítimo de Ponta da Madeira (TMPM) e o Porto da Alumar. A Empresa Maranhense de Administração Portuária (EMAP) é responsável pela administração do Porto do Itaqui, tendo como principal movimentação a importação de derivados de petróleo. A mineradora Companhia Vale do Rio Doce (Vale) opera no TMPM, a exportação de minério de ferro e pelotas, derivados do minério de ferro, é o seu principal negócio. O Porto da Alumar está associado ao Consórcio de Alumínio do Maranhão (Alumar), que produz alumínio e alumina. (EMAP, 2009 apud BARROS; ALUMAR, 2009 apud BARROS; VALE, 2008 apud BARROS; BARROS, 2010).

O complexo portuário maranhense se destaca pela movimentação de granéis sólidos. A principal diferença de granéis sólidos para contêineres ou cargas gerais soltas, do ponto de vista das operações portuárias, está na estocagem e no transporte. Em terminais portuários com essas características o tempo de atracação está vinculado, principalmente, à capacidade dos carregadores/descarregados de navios. Outra característica do complexo portuário de São Luís é a presença acentuada do fenômeno conhecido como *avanço das marés*. Existe uma grande amplitude dos níveis de maré de São Luís, limitando, desta forma, a navegação dos navios, nas operações de atracação ou desatracação, a determinados períodos do dia.

Diante deste cenário, o problema de alocação de berços (PAB) em terminais portuários surge como um dos problemas fundamentais no planejamento das operações portuárias. O problema é determinar onde e quando os navios devem ser atracados, minimizando o custo total provocado. Berços são os espaços reservados no cais dos portos, onde os navios podem operar em segurança. Estes em geral são heterogêneos, ou seja, possuem infraestruturas variadas, o que acarreta em capacidades de vazão diferentes.

O PAB pertence à classe de problemas que podem ser classificados como NP-difíceis equivalentes ao problema de escalonamento com restrições de recursos

(BLAZEWICZ, 1996). Portanto, torna-se interessante propor a este problema algoritmos aproximativos na tentativa de obtenção de soluções boas em tempo hábil. Dentro dessa classe de algoritmos a Otimização por Colônia de Formigas (Ant Colony Optimization - ACO), como uma metaheurística promissora, pois seu desempenho na aplicação na maioria dos problemas clássicos tem sido favorável, comparando com outras técnicas como, por exemplo, os Algoritmos Genéticos.

## **1.2 Objetivos**

Este trabalho tem como objetivo apresentar uma aplicação de um algoritmo baseado na metaheurística Otimização por Colônia de Formigas ao Problema de Alocação de Berços. Especificamente, pretende-se:

- Propor uma modelagem computacional do PAB, e uma nova estratégia de construção de soluções da ACO, tornando possível a aplicação do ACO ao PAB.
- Implementar a aplicação da metaheurística Otimização por Colônia de Formigas ao Problema de Alocação de Berços.
- Avaliar o desempenho desta aplicação;

## **1.3 Organização do Trabalho**

Este trabalho está organizado em capítulos, sendo este o primeiro e os demais são resumidos da seguinte maneira. O Capítulo 2 apresenta os principais conceitos referentes à metaheurística ACO. O Capítulo 3 descreve o problema de alocação de berços. Neste capítulo será mostrada a modelagem matemática do PAB. No Capítulo 4 será mostrada de forma detalhada a proposta deste trabalho. Aqui será ilustrada a modelagem computacional do PAB necessária para aplicação do ACO a ele, bem como a descrição do algoritmo ACO proposto, para que seja aplicável ao PAB. O capítulo 5 apresenta resultados obtidos na aplicação do ACO ao PAB, e os analisa. O capítulo 6 apresenta as conclusões deste trabalho.

## 2. OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS

No início da década de 1990, Marco Dorigo, em sua tese de doutorado, introduziu a otimização por colônia de formigas (Ant Colony System - ACO), como uma nova metaheurística para resolver problemas difíceis de otimização discreta. ACO é uma metaheurística em que uma colônia de formigas artificiais coopera entre si na busca de soluções para esse tipo de problema.

### 2.1. Inspiração Biológica

Pesquisas sobre o comportamento das formigas têm mostrado que a maior parte da comunicação entre os indivíduos de uma colônia se dá através de substâncias químicas produzidas pelas formigas. Estas substâncias são chamadas de feromônios. Algumas espécies de formigas, tais como *Laisus niger* ou a *Iridomrmex humilis* (Goss, Aron, Deneubourg e Pasteels, 1989), usam trilhas de feromônio para marcar caminho no chão, por exemplo, caminhos entre as fontes de alimento e o ninho. Essa trilha de feromônio estimula formigas a seguirem caminhos feitos por outras formigas. Ou seja rastro de substancias químicas influenciam as formigas a seguirem as outras. Esse comportamento propicia uma organização coletiva onde a cooperação entre os indivíduos melhora o desempenho das tarefas. Os algoritmos baseados na metaheurística ACO tentam reproduzir de forma artificial esse comportamento.

Esse comportamento de algumas espécies de formigas tem sido investigado em experimentos controlados por vários pesquisadores. Um experimento em particular, chama a atenção. Deneubourg e colegas (Deneuborg, Aron, Goss e Pasteels, 1990; Goss et al., 1989) apud (DORIGO, 2004) usaram uma ponte dupla para ligar um ninho de formigas da espécie argentina *I. humilis* e uma fonte de alimento. Eles fizeram os experimentos variando a razão entre o tamanho dos braços da ponte. No primeiro experimento os dois braços tinham o mesmo tamanho, ou seja a razão entre seus tamanhos é igual 1, como mostra a Figura 1.

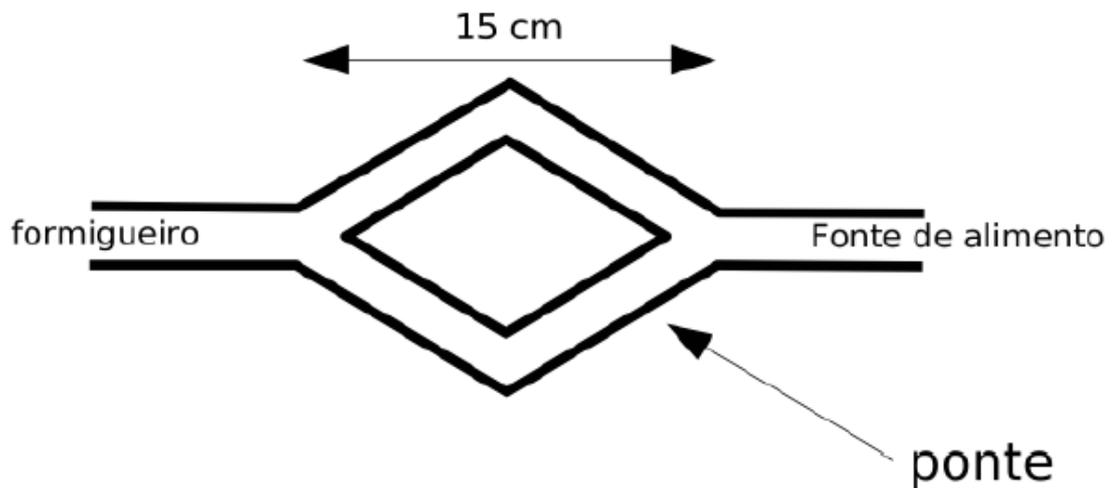


Figura 1: Experimento da ponte dupla. Retirado e adaptado de:  
 <[http://www.ufpi.br/subsiteFiles/rbritto/arquivos/files/Aula10\\_Colonia%20de%20Formigas.pdf](http://www.ufpi.br/subsiteFiles/rbritto/arquivos/files/Aula10_Colonia%20de%20Formigas.pdf)>

Quando o experimento começa as formigas são deixadas livres para se moverem entre o ninho e a fonte de alimento e durante o experimento a porcentagem de formigas que escolhem um ou outro braço da ponte é observada em relação ao tempo. O que se observou foi que embora na fase inicial sejam feitas escolhas aleatórias entre os dois braços, depois de algum tempo a tendência é que todas formigas usem o mesmo braço (Figura 2). Isso se dá pelo fato de que devido a oscilações aleatórias de probabilidade, visto que os dois braços têm o mesmo tamanho, eventualmente mais formigas passam deixando feromônio em um dos braços. Isso faz com que em um dos braços da ponte se tenha mais feromônio que o outro, o que estimula mais formigas a passarem por ali. Esse processo mostra um tipo de comportamento de auto-organização das formigas, onde se cria um padrão coletivo a partir de comportamentos individuais (Camazine, Deneubourg, Franks, Sneyd, Theraulaz, & Bonabeau, 2001; Haken, 1983; Nicolis & Prigogine, 1977) apud (DORIGO, 2004). Além disso, isto é um exemplo de estigmergia, que um tipo de comunicação indireta mediada pelo ambiente, ou pela modificação do ambiente.

No segundo experimento, um dos braços da ponte foi aumentado tornando a razão entre os tamanhos dos braços igual a 2 (Goss et al., 1989) apud (DORIGO, 2004). Neste caso, na maior parte dos ensaios as formigas convergiram para o caminho mais curto, como mostra a Figura 3. Isso ocorre devido ao fato de que, neste experimento, as formigas que optaram pelo braço menor voltam para o ninho mais rápido que as outras, e assim, com o passar do tempo há um grande acúmulo de feromônio nesse caminho menor, o que atrai outras formigas para esse caminho.

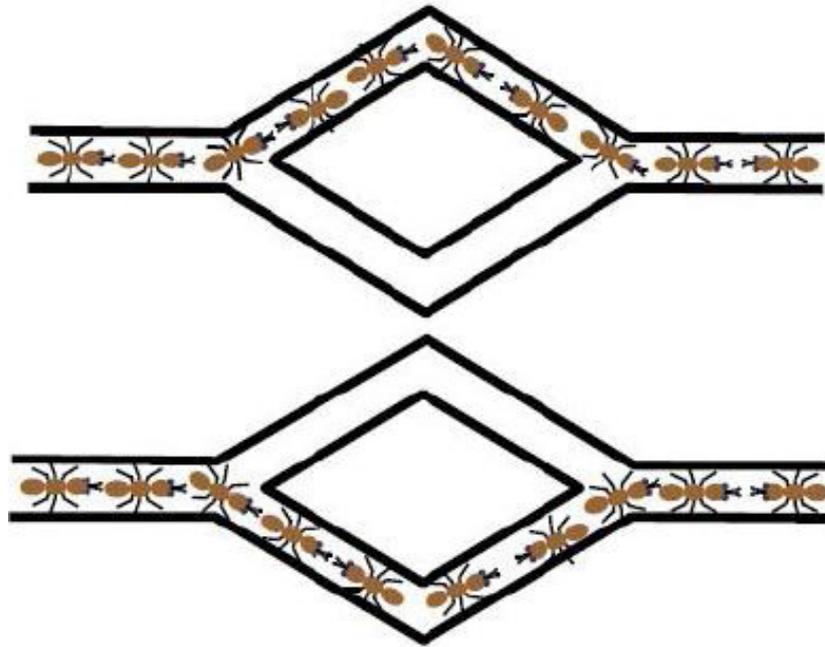


Figura 2: Comportamento das formigas, depois de um tempo no experimento da ponte dupla. Retirada e adaptada de: <[http://www.ufpi.br/subsiteFiles/rbritto/arquivos/files/Aula10\\_Colonia%20de%20Formigas.pdf](http://www.ufpi.br/subsiteFiles/rbritto/arquivos/files/Aula10_Colonia%20de%20Formigas.pdf)>

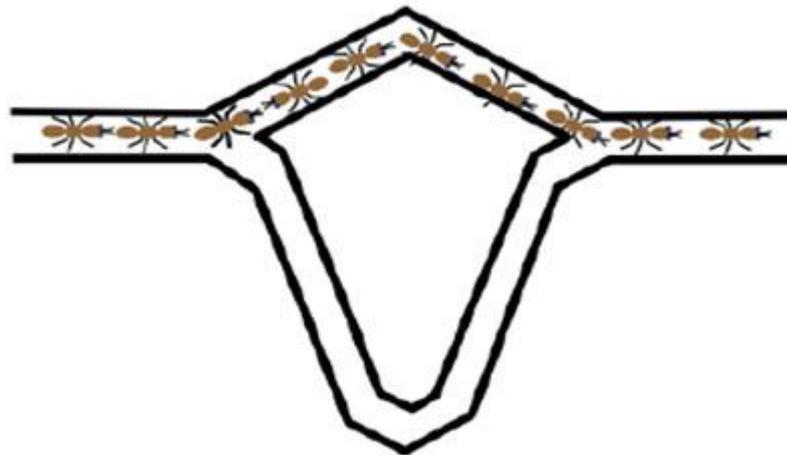


Figura 3: Convergência das formigas para o caminho mais curto. Retirada e adaptada de: <[http://www.ufpi.br/subsiteFiles/rbritto/arquivos/files/Aula10\\_Colonia%20de%20Formigas.pdf](http://www.ufpi.br/subsiteFiles/rbritto/arquivos/files/Aula10_Colonia%20de%20Formigas.pdf)>

Diante disso a metaheurística ACO tenta copiar o comportamento de auto-organização e comunicação indireta mediada pelo ambiente, para que formigas artificiais cooperem e construam soluções boas para problemas de otimização discreta e semelhantes.

## 2.2. A Metaheurística ACO

A otimização por colônia de formigas é uma metaheurística em que agentes simples, chamados de formigas, cooperam na busca de soluções para problemas de otimização discreta, utilizando uma comunicação indireta mediada pelo ambiente. A seguir temos a descrição formal da classe de problemas para os quais a ACO pode ser aplicado.

### 2.2.1. Representação do Problema

Uma formiga artificial na ACO é um procedimento construtivo estocástico que de forma incremental constrói uma solução pela adição de um componente a uma solução parcial em construção (DORIGO, 2004). Portanto, a ACO pode ser definida para qualquer problema de otimização combinatória para o qual uma heurística construtiva possa ser definida.

Apesar disso, é preciso considerar o problema de como criar para um determinado problema, uma representação que possa ser usada pelas formigas artificiais para construir soluções. Por isso, Dorigo propõe uma caracterização formal da representação que as formigas artificiais usam e da política que elas implementam (Dorigo, 2004). A seguir tem-se descrição dessa caracterização.

Considerando o problema de minimização  $(S, f, \Omega)$ , onde  $S$  é o conjunto de soluções candidatas,  $f$  é a função objetivo que atribui um valor de custo  $f(s, t)$  para cada solução candidata  $s \in S$ , e  $\Omega(t)$  é o conjunto de restrições. O parametro  $t$  indica que a função objetivo e as restrições podem ser dependentes do tempo, como no caso da aplicação da ACO para problemas dinâmicos. O objetivo é achar uma solução  $s^*$  viável (não viola as restrições) globalmente ótima, isto é, a solução viável com custo mínimo (ou ganho máximo) para o problema. O problema de otimização combinatória  $(S, f, \Omega)$  é mapeado em um problema com as seguintes características:

- Um conjunto finito  $C = \{c_1, c_2, \dots, c_N\}$  de componentes é dado, onde  $N$  é o numero de componentes.
- Os estados de um problema são definidos em termos de sequências finitas  $x = \langle c_1, c_2, \dots, c_h \rangle$  de elementos de  $C$ . O conjunto de todos os possíveis estados é denotado por  $X$ .

- O conjunto finito de restrições  $\Omega$  define o conjunto de estados viáveis  $X^*$ .
- O conjunto não vazio  $S^*$  de soluções ótimas, que  $S^* \subseteq X^*$  e  $S^* \subseteq S$ .
- Um custo  $f(s, t)$  é associado a cada solução candidata  $s \in S$ .

Dada esta formulação, formigas artificiais constroem soluções pela realização de caminhos randomizados em um grafo completamente conectado  $G_C = (C, L)$  onde os nós ( $C$ ) são os *componentes* e as arestas ( $L$ ) são as conexões entre os componentes. O grafo  $G_C$  é chamado de grafo de construção.

As restrições do problema são implementadas na política de construção de soluções das formigas artificiais. Isto pode ser feito de duas formas diferentes. Primeiro, na forma de restrição absoluta, eliminando certas ligações entre nós que determinam soluções inviáveis. Ou podem ser implementadas a partir da penalização de soluções inviáveis (DORIGO, 2004; TALBI, 2009).

### 2.2.2. Comportamento das formigas

Formigas artificiais são procedimentos construtivos estocásticos que constroem soluções pelo movimento em um grafo de construção  $G_C = (C, L)$ , como descrito anteriormente. Os componentes  $c_i \in C$  e as conexões  $l_{ij} \in L$  têm associado uma trilha de feromônio  $\tau$  ( $\tau_i$  se está associado com o componente  $i$  e  $\tau_{ij}$  se está associado com a conexão entre o componente  $i$  e o componente  $j$ ) e um valor de informação heurística  $\eta$  ( $\eta_i$  e  $\eta_{ij}$ , respectivamente). A trilha de feromônio codifica uma memória em longo prazo do processo de busca das formigas. Essa trilha de feromônio é atualizada durante a execução do algoritmo para refletir a experiência das formigas na busca de soluções boas. Já a informação heurística representa um conhecimento a priori que se tem do problema, ou da instância do problema.

Mais precisamente, cada formiga  $k$  da colônia tem as seguintes propriedades:

- Ela explora um grafo de construção  $G_C = (C, L)$  para buscar soluções ótimas  $s^* \in S^*$ .
- Ela tem uma memória  $M^k$  que pode ser usada para armazenar informações sobre o caminho parcial que ela seguiu. A memória pode ser usada para:

1. Construir soluções viáveis, ou seja soluções que não infringem as restrições do conjunto  $\Omega$ .
  2. Calcular e inferir valores heurísticos  $\eta$ .
  3. Avaliar a solução encontrada;
  4. refazer o caminho inverso para depositar feromônio;
- Ela tem um estado inicial  $x_i^k$  definido como uma sequência de comprimento unitário (componente simples ou sequência vazia) e uma ou mais condições de parada  $e_k$ ;
  - Quando está em um estado  $x_r = \langle x_{r-1}, i \rangle$ , se nenhuma condição de parada estiver satisfeita, ela se move para o nó  $j$  em sua vizinhança  $N^k(x_r)$ , isto é, para o estado  $\langle x_r, j \rangle \in X$ . Quando uma formiga constrói uma solução candidata, mover-se para estados inviáveis são proibidos na maioria das aplicações, seja através do uso de memória da formiga, ou através de valores heurísticos adequadamente definidos  $\eta$ .
  - Ela escolhe um movimento através da aplicação regra de decisão probabilística. Esta regra de decisão probabilística é função de:
    - Trilhas de feromônio disponíveis localmente e valores heurísticos.
    - Memória privada da formiga que guarda seu estado corrente.
    - Restrições do problema.
  - Quando adiciona o componente  $c_j$  para o estado corrente, ela pode atualizar a trilha de feromônio associado a ele ou a conexão correspondente.
  - Uma vez que ela construiu uma solução ela pode refazer o mesmo caminho no sentido contrário e atualizar as trilhas de feromônio das conexões usadas.

Formigas agem concorrentemente e independentemente. Além disso, embora cada formiga seja complexa o suficiente para construir uma solução (provavelmente de baixa qualidade) para o problema considerado, somente a cooperação entre formigas pode propiciar a construção de soluções de boa qualidade (DORIGO e STÜTZLE, 2002). Esta cooperação é feita através de uma comunicação indireta pelas informações que as formigas leem ou escrevem em variáveis que guardam os valores das trilhas de feromônio.

### 2.2.3. A Metaheurística

Informalmente, um algoritmo ACO pode ser pensado como a execução de três componentes: *Construção de Soluções*, *Atualização dos Feromônios* e *Ações Globais*. Estes componentes são definidos a seguir:

*Construção de Soluções*: um conjunto de formigas artificiais constrói soluções a partir dos elementos do conjunto de componentes  $C$ . A construção da solução inicia com uma solução parcial vazia  $s^p = \emptyset$ . Então, a cada passo de construção, a solução parcial  $s^p$  é estendida pela adição de um componente viável do conjunto de vizinho viáveis  $N(s^p) \subseteq C$ . O processo de construção de soluções pode ser considerado como um caminho no grafo  $G_C$ . Por isto, os caminhos permitidos em  $G_C$  são implicitamente definidos pelo mecanismo de construção de solução que define o conjunto  $N(s^p)$ . A escolha de um componente de  $N(s^p)$  é feita probabilisticamente a cada passo da construção. As regras para a escolha do componente variam de acordo com as diversas variações da ACO.

*Atualização dos Feromônios*: O objetivo da atualização do feromônio é incrementar os valores de feromônio associados a soluções boas ou promissoras e decrementar os valores de feromônio associados às soluções ruins. Isto é feito (i) pelo decremento de todos os valores de feromônio através da *evaporação de feromônio*, e (ii) pelo incremento dos valores de feromônio associado com um conjunto escolhido de soluções. A evaporação do feromônio é necessária para impedir a convergência rápida do algoritmo. Ela implementa uma espécie de perda de memória da colônia, o que é muito útil, pois possibilita a exploração de novas áreas no espaço de busca.

*Ações Globais (Daemon Actions)*: Implementa ações centralizadas e/ou dependentes do problema, que não podem ser feitas pelas formigas individualmente. A maioria das *Daemon Actions* consiste na aplicação de métodos de busca local para as soluções construídas.

De forma geral a metaheurística ACO pode ser descrita pelo pseudocódigo na Figura 4. O procedimento principal da metaheurística ACO gerencia a programação dos três componentes algorítmicos acima citados.

Iniciar parâmetros, iniciar rastros de feromônio

Agendar Atividades

Construção de Soluções

Ações Globais [Opcional]

Atualização dos Feromônios

Fim do Agendamento

Figura 4: Pseudocódigo da metaheurística ACO

#### 2.2.4. Aplicação ao problema

Para ilustrar como a ACO trabalha e é aplicada em um problema específico será mostrado o algoritmo que melhor exemplifica as estratégias da ACO na aplicação a um problema de otimização. O primeiro exemplo de um algoritmo que segue a estratégia da ACO é o *Ant System* (AS) desenvolvido por Marco Dorigo, Vittorio Maniezzo e Alberto Coloni em 1996 (DORIGO, 2004) para resolver o Problema do Caixeiro Viajante (*Traveling Salesman Problem* - TSP). O AS alcançou resultados animadores inicialmente, mas ficou evidente que seu desempenho é inferior se comparado com os melhores algoritmos propostos para resolver o TSP. No entanto sua importância reside principalmente em ser a inspiração prévia de uma série de extensões que melhoraram significativamente o desempenho e estão atualmente entre os algoritmos ACO mais bem sucedidos. Além disso, do ponto de vista didática, o AS funciona bem como algoritmo ilustrativo do funcionamento da ACO.

Na fase de construção do AS,  $m$  formigas artificiais constroem uma solução do TSP. A cada passo da construção, a formiga  $k$  aplica uma regra de escolha probabilística para decidir qual será a próxima cidade a ser visitada. Em particular, a probabilidade da formiga  $k$ , estando na cidade  $i$ , escolher ir para a cidade  $j$  é

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} \quad (2.1)$$

Onde  $n_{ij} = \frac{1}{d_{ij}}$  é o valor heurístico disponível a priori,  $d_{ij}$  é a distancia entre a cidade  $i$  e a cidade  $j$ ,  $\alpha$  e  $\beta$  são os dois parâmetros que determinam a influência relativa do feromônio e da informação heurística, e  $N_i^k$  é a vizinhança factível para a formiga  $k$  estando na cidade  $i$ , isto é, o conjunto de cidades que a formiga  $k$  ainda não visitou. Através desta regra probabilística, a probabilidade de uma aresta  $(i,j)$  ser escolhida aumenta a medida que a trilha de feromônio  $\tau_{ij}$  e o valor de informação heurística  $\eta_{ij}$  associados aumentam.

AS não realiza nenhuma ação global, nem busca local. Assim, a próxima fase, após a construção da solução, é a atualização do feromônio, que envolve tanto o incremento do feromônio, quanto a sua evaporação e é realizado por todas as formigas. A atualização é definida pela equação (2.2) a seguir

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k, \quad \forall (i,j) \in L \quad (2.2)$$

Onde  $0 < \rho \leq 1$  é a taxa de evaporação do feromônio e  $\Delta\tau_{ij}^k$  é a quantidade de feromônio que a formiga  $k$  deposita nas arestas que ela visitou.  $\Delta\tau_{ij}^k$  é definida a seguir:

$$\Delta\tau_{ij}^k = \begin{cases} 1/C^k, & \text{se } (i,j) \in T^k, \\ 0, & \text{caso contrario} \end{cases} \quad (2.3)$$

onde  $C^k$  é o tamanho do percurso feito pela formiga  $k$ . Assim as arestas que pertencem ao melhor percurso encontrado receberão mais feromônio e, portanto, são mais propensas a ser escolhidas por formigas em futuras iterações do algoritmo.

## 2.3. Variações da ACO

### 2.3.1. Elitist Ant System (EAS)

É o primeiro melhoramento do AS, proposto por Dorigo (1992). A ideia é fornecer um forte reforço adicional às arestas que pertencem ao melhor percurso encontrado desde o inicio da execução do algoritmo. Este percurso é denotado por  $T^{bs}$  (do inglês *best-so-far tour*). Esta é uma estratégia similar a estratégia elitista dos

algoritmos evolucionários. Percebe-se que esse reforço adicional de feromônio é um exemplo de *ação global* que pode ser tomada.

- **Atualização do Feromônio**

A seguir temos a equação que descreve atualização do feromônio para este algoritmo ACO:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k + e\Delta\tau_{ij}^{bs} \quad (2.4)$$

Onde  $\Delta\tau_{ij}^{bs}$  é definido como:

$$\Delta\tau_{ij}^k = \begin{cases} 1/C^k, & \text{se } (i,j) \in T^{bs}, \\ 0, & \text{caso contrario} \end{cases} \quad (2.5)$$

Sendo  $e$  um parâmetro definido pelo usuário.

### 2.3.2. Rank-Based Ant System (AS<sub>rank</sub>)

Outro melhoramento do AS, foi proposto por (BULLNHEIMER et al, 1999). Neste algoritmo cada formiga deposita uma quantidade de feromônio que diminui com seu rank. Adicionalmente, assim como no EAS, o melhor percurso encontrado até a interação corrente recebe um reforço adicional de feromônio.

- **Atualização do Feromônio**

Antes de atualizar as trilhas de feromônio, as formigas são classificadas na ordem decrescente da qualidade do seu percurso. Então as  $w - 1$  melhores formigas depositam de acordo com a equação (2.6)

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{r=1}^{w-1} (w - r)\Delta\tau_{ij}^r + w\Delta\tau_{ij}^{bs} \quad (2.6)$$

Onde  $r$  é o rank da formiga,  $\Delta\tau_{ij}^r = 1/C^r$  e  $\Delta\tau_{ij}^{bs} = 1/C^{bs}$ . De acordo com Bullnheimer et al (1999), AS<sub>rank</sub> é levemente melhor que o EAS e significativamente melhor que o AS.

### 2.3.3. Ant Colony System (ACS)

Este algoritmo foi proposto por DORIGO e GAMBARDILLA (1997). Através de elitismo, faz uso de mais de intensificação do que o AS. O ACS usa duas regras de atualização do feromônio: uma local e outra global onde somente a melhor formiga faz acréscimo de feromônio no seu percurso.

- **Fase de construção**

No ACS, estando localizada em vértice  $i$ , a formiga  $k$  se move para o vértice  $j$  de acordo com uma regra, dada por

$$j = \begin{cases} \operatorname{argmax}_{l \in N_i^k} \{\tau_{il} [\eta_{il}]^\beta\}, & \text{se } q \leq q_0; \\ J, & \text{caso contrario;} \end{cases} \quad (2.7)$$

Onde  $q$  é uma variável aleatória uniformemente distribuída em  $[0, 1]$ ,  $q_0$  ( $0 \leq q_0 \leq 1$ ) é um parâmetro, e  $J$  é uma variável aleatória selecionada de acordo com a equação (2.1), com  $\alpha = 1$ .

Em outras palavras, com probabilidade  $q_0$ , a formiga  $k$  se move para o melhor vertice possível de acordo com o que as trilhas de feromônio e a informação heurística indicam. Enquanto com probabilidade  $(1 - q_0)$  a formiga segue a regra de escolha do AS. O ajuste do parâmetro  $q_0$  permite a modulação do grau de exploração, favorecendo ou a intensificação, o que faz a busca se concentrar em torno das melhores soluções, ou a diversificação, o que permite a exploração de outros percursos.

- **Atualização do Feromônio**

Conforme citado anteriormente, o ACS tem duas regras de atualização do feromônio, uma global e um local. A *regra de atualização do feromônio global* define que somente a formiga que encontrou o melhor percurso pode acrescentar feromônio depois de cada interação, e definida pela equação a seguir:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs}, \quad \forall (i, j) \in T^{bs} \quad (2.8)$$

onde  $\Delta\tau_{ij}^{bs} = 1/C^{bs}$ . É importante notar que, com essa regra, em ACS, a atualização do feromônio, tanto a evaporação quanto o depósito de feromônio, acontece

somente nos nas arestas que pertencem a  $T^{bs}$ . Isto é importante, porque deste modo a complexidade computacional da atualização de feromônio em cada interação é reduzida de  $O(n^2)$  para  $O(n)$ , onde  $n$  é o tamanho da instância a ser resolvida.

Em adição à regra global o ACS tem uma *regra de atualização do feromônio local* que é aplicada imediatamente após a passagem pela aresta  $(i, j)$  durante a construção do percurso e é denifina por

$$\tau_{ij} \leftarrow (1 - \xi)\tau_{ij} + \xi\tau_0 \quad (2.9)$$

onde  $\xi$ , com  $0 < \xi < 1$ , e  $\tau_0$  são dois parametros. O valor de  $\tau_0$  é configurado para ser o mesmo valor do feromônio inicial em todos as arestas. O efeito da regra de atualização local é que cada vez que uma formiga usa uma aresta  $(i, j)$  seu valor de feromônio  $\tau_{ij}$  é reduzido, de modo que a aresta torna-se menos desejável para as seguintes formigas. Em outras palavras, isto permite um aumento na exploração dos arcos que não tenham sido visitados ainda e, na prática, é uma tentativa de evitar que o algoritmo acabe convergindo prematuramente.

Neste trabalho o algoritmo baseado na ACO escolhido para ser aplicado ao PAB, foi o ACS. No capítulo 4 onde é descrito a proposta deste trabalho, esta aplicação será descrita.

### 3. PROBLEMA DE ALOCAÇÃO DE BERÇOS

O problema da alocação de berços (PAB) a navios em portos graneleiros consiste em determinar qual a melhor ordem que os navios devem atracar em cada berço de forma que o custo seja o mínimo possível, tal custo pode ser mensurado de várias maneiras, tais como tempo total de operação ou multa por sobrestada. Um berço é um local específico ao longo do cais onde o navio irá atracar para ser carregado ou descarregado. Cada berço pode acomodar um navio por vez, quando todos os berços estão ocupados os navios que chegarem, posteriormente, terão que esperar sua vez até que algum navio desocupe um berço. Assim o PAB pode ser definido como o planejamento do tempo e da posição de atracação referentes a cada navio (MOON, 2000 apud BARROS; CORDEAU et al., 2005 apud BARROS; BARROS, 2010). Se não houver posição de atracação adequada para o navio, este deve aguardar em uma fila de espera (FERNANDES, 2001 apud BARROS, 2010). A Figura 5 descreve um cenário típico de porto graneleiro.

Neste trabalho a metaheurística ACO será aplicada ao *problema de alocação de berços heterogêneo*, onde, como o nome indica, os berços são tratados como heterogêneos, isso porque cada um possui suas próprias características tais como velocidade de carga e descarga e tipo de carga que cada um manipula. É importante observar essas características, porque tratá-los como homogêneos levará a encontrar soluções que em teoria parecem satisfatórias, mas na prática não resolvem de fato o problema.

Portos podem estar associados a importantes indústrias nacionais ou internacionais, logo é muito importante que estes funcionem de maneira ágil para que essas grandes empresas possam receber ou enviar seus produtos através destes. Para garantir tal agilidade é muito importante manter o estoque das matérias-primas em níveis seguros o suficiente para não comprometer o funcionamento da empresa.

O nível de estoque depende de um processo constante de consumo e produção de graneis (minério de ferro, bauxita, soja, etc.). Considerando cada taxa de consumo/produção de cada carga somada à quantidade já armazenada os pátios

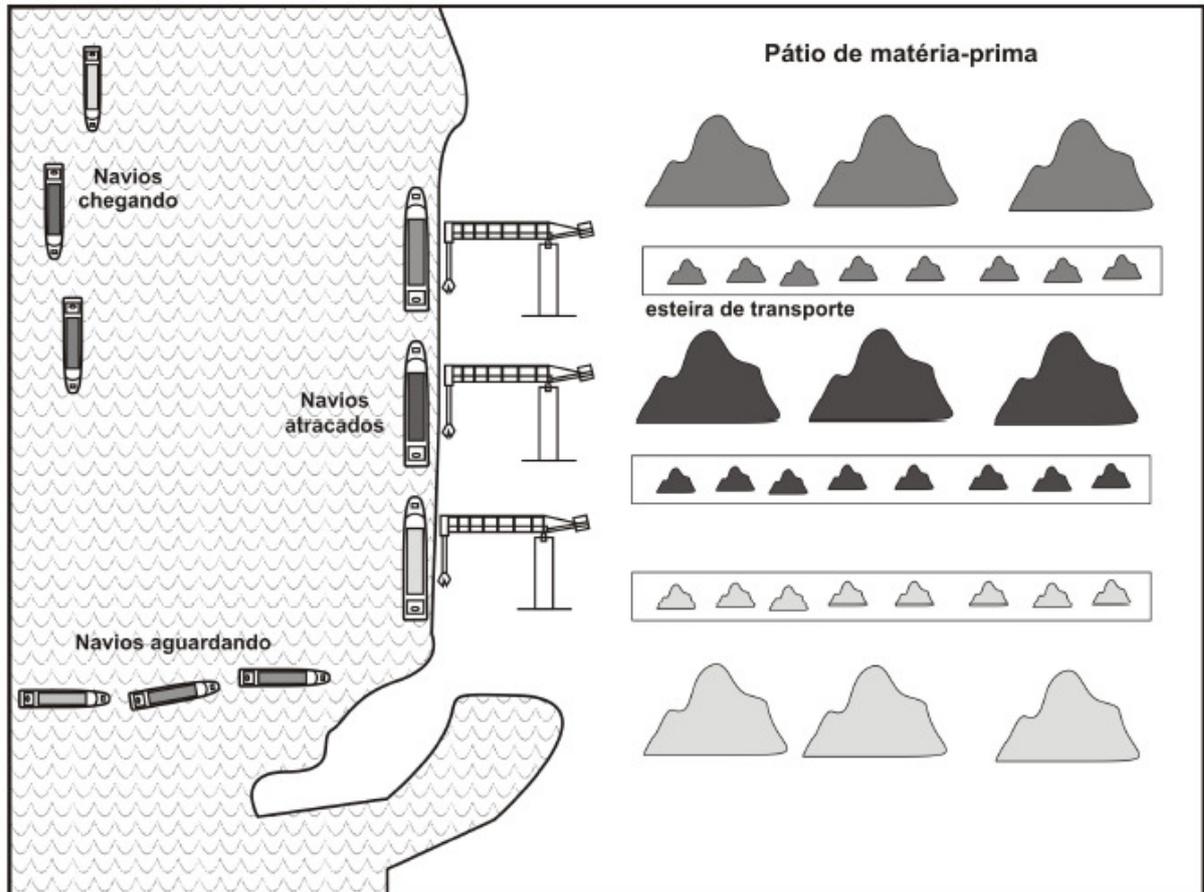


Figura 5: Cenário típico de um porto graneleiro. Fonte: (BARROS, 2009, pp. 13).

é possível controlar os estoques e evitar que navios permaneçam atracados ao porto sem a carga para carregá-los ou que a produção pare por falta de insumos e matérias-primas. A decisão de carregar ou descarregar um navio deve visar manter os estoques nos intervalos aceitáveis, dependendo da capacidade de consumo e/ou produção, bem como do espaço físico para armazená-los.

Sendo um problema de otimização combinatória, o Problema de Alocação de Berços, para instâncias de grande e médio porte, se torna difícil e excessivamente custoso de resolver por métodos exatos, por isso se justifica o uso de metaheurísticas que são algoritmos aproximados de caráter geral que são rápidos e nos garantem soluções boas e até mesmo ótimas.

### 3.1. Modelagem Matemática

A seguir tem-se a modelagem do problema de alocação de berços em terminais portuários com berços heterogêneos proposta em (Barros et al., 2009). A questão da heterogeneidade dos berços está ligada à infraestrutura que estes

comportam. Infraestruturas diferentes resultam em vazões, ou velocidades de atendimento, diferentes. Neste modelo não há restrições de navios e berços, ou seja, qualquer navio pode atracar em qualquer berço e todas as cargas podem ser movimentadas por todos os berços. O cenário com berços heterogêneos é claramente o mais próximo da realidade, portanto justifica-se o enfoque sobre este modelo.

O fato de haver janelas de tempo disponíveis para atracação conduz ao tratamento do tempo na forma discreta, supondo que os períodos com condições favoráveis sejam suficientemente pequenos. Em geral, para portos com grande amplitude dos níveis de maré, são fornecidos pequenos para as operações de atracamento e desatracamento. Dessa forma, o horizonte de planejamento pode ser dividido em  $|M|$  janelas de tempo. Neste trabalho, o acrônimo TTW (Tidal Time Window) é utilizado para designar essas janelas de tempo.

Assim o PAB é modelado na forma discreta como um problema transporte no qual  $N$  navios são vistos como consumidores e os berços  $L$  e as janelas de maré favoráveis  $M$ , como fornecedores. Cada navio  $i$  deve ser alocado a um berço  $l$  em subconjunto de TTW's, cuja cardinalidade depende do número de TTW's necessárias  $h_{il}$  para que a operação de carga/descarga seja completada. A Figura 6 a seguir mostra o PAB modelado como um problema de transporte considerando apenas as janelas de marés.

As variáveis de decisão são dadas por  $x_{ij}$ ,  $u_{il}$  e  $y_{ijl}$ , onde

$$x_{ij} = \begin{cases} 1, & \text{se o navio } i \text{ estiver alocado a TTW } j \\ 0, & \text{de outra forma} \end{cases} \quad (3.1)$$

$$u_{ij} = \begin{cases} 1, & \text{se o navio } i \text{ estiver alocado ao berço } l \\ 0, & \text{de outra forma} \end{cases} \quad (3.2)$$

Portanto neste modelo temos duas decisões a fazer. A primeira é determinar uma sequência de  $x_{ij}$ 's para cada navio  $i$  que determine o subconjunto de TTW's em que o navio permanece atracado, sendo atendido. E a segunda é determinar o berço  $l$  que atenderá o navio  $i$ . Vale notar que um navio só pode atracar em um dos berços portanto somente um dos  $u_{il}$ 's receberá o valor 1.

Para associar as variáveis  $x_{ij}$  e  $u_{il}$  sem tornar o modelo não-linear, utiliza-se uma técnica onde cria-se uma variável auxiliar  $y_{ijl}$ .

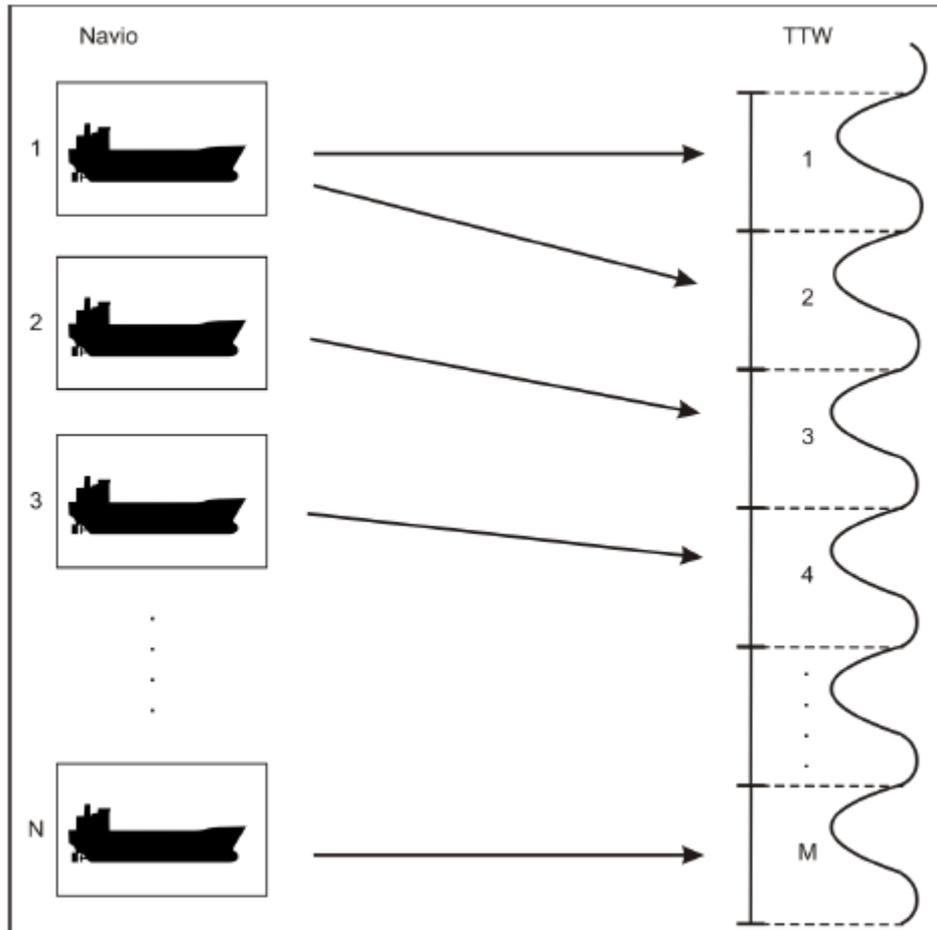


Figura 6: PAB modelado como um problema de transporte. Fonte: (BARROS, 2009, pp. 33).

### 3.1.1. Dados de entrada

Os dados de entrada para o PAB neste modelo são os seguintes:

- $N$ : conjunto de navios;
- $M$ : conjunto de marés;
- $L$ : conjunto de berços;
- $K$ : conjunto de matérias-primas operadas no porto;
- $a_i$ : TTW de chegada do navio  $i$ ;
- $v_l$ : velocidade de carregamento/descarregamento do berço  $l$ ;

- $e_k$ : nível de estoque inicial da carga granel  $k$ ;
- $c_k$ : quantidade de consumo ou produção da carga granel  $k$ ;
- $q_{ik}$ : capacidade de carga do navio  $i$  em relação a carga  $k$ .

Outro parâmetro usado no modelo,  $h_{il}$ , diz respeito ao tempo de processamento necessário para um navio  $i$  ser atendido no berço  $l$  (*handling time*). Este parâmetro, no entanto, é definido pela expressão a seguir:

$$h_{il} = \left\lceil \frac{\sum_{k=1}^{|K|} q_{ik}}{v_l} \right\rceil \quad (3.3)$$

Ou seja, é definido como uma função dos parâmetros  $v_l$  e  $q_{ik}$ . Este parâmetro é apenas uma aproximação pois outros fatores, como o tempo de inicialização não são levados em consideração.

### 3.1.2. Função Objetivo

O objetivo, para o problema considerado, é minimizar o custo acumulado sobre todas as operações em um horizonte de planejamento dado. Esse custo está relacionado ao tempo de serviço de todos os navios. A equação a seguir define o custo:

$$s_i = \sum_{j=a_i}^M \sum_{l=1}^L \left\lceil \frac{j - a_i + 1}{h_{il}} \right\rceil \times y_{ijl} \quad (3.4)$$

Minimizar o somatório dos tempos de serviço  $s_i$  de cada navio corresponde, portanto, à função objetivo:

$$\min \sum_{i=1}^N \sum_{j=a_i}^M \sum_{l=1}^L \left\lceil \frac{j - a_i + 1}{h_{il}} \right\rceil \times y_{ijl} \quad (3.5)$$

### 3.1.3. Restrições

A seguir temos as equações que descrevem as restrições do problema:

$$\sum_{i=1}^{a_i-1} x_{ij} = 0, \quad \forall i \in N \quad (3.6)$$

$$\sum_{j=a_i}^M x_{ij} = \sum_{l=1}^L h_{il} \times u_{il}, \quad \forall i \in N \quad (3.7)$$

$$\sum_{i=1}^N y_{ijl} = 1, \quad \forall j \in M, l \in L \quad (3.8)$$

$$\sum_{i=1}^L u_{il} = 1, \quad \forall i \in N \quad (3.9)$$

$$\sum_{z=1}^{j-1} x_{iz} - j \times x_{ij-1} + j \times x_{ij} \leq j, \quad \forall i \in N, j > a_i \in M \quad (3.10)$$

$$\sum_{i=1}^N \sum_{j=a_i}^M \sum_{l=1}^L \frac{q_{ik}}{h_{il}} \times y_{ijl} \geq j \times w_k - e_k, \quad \forall j \in M, k \in K \quad (3.11)$$

$$y_{ijl} \geq x_{ij} + u_{il} - 1 \quad \forall j \in M, j \in M, l \in L \quad (3.12)$$

$$y_{ijl} \leq x_{ij} \quad \forall j \in M, j \in M, l \in L \quad (3.13)$$

$$y_{ijl} \leq u_{il} \quad \forall j \in M, j \in M, l \in L \quad (3.14)$$

As equações 3.6 e 3.7 obrigam os navios atracarem após suas TTW's de chegada,  $a_i$ , e permanecem em operação durante  $h_{il}$  TTW's para completar a operação. O número de navios alocados a uma TTW não pode exceder o total de posições de berços, a equação (3.8) descreve essa restrição. A equação (3.9) obriga que somente um navio esteja atracado em um berço.

A equação (3.10) evita soluções com interrupções prematuras nas operações do navio. A equação (3.11) garante que os níveis mínimos de estoque sejam mantidos maiores que zero. As equações (3.12) a (3.14) garantem a consistência do modelo. O objetivo é obter  $y_{ijl} = 1$  quando  $x_{ij} = 1$  e  $u_{il} = 1$ ,  $y_{ijl} = 0$  quando pelos uma das variáveis  $x_{ij}$  ou  $u_{il}$ , seja igual a zero.

## 4. PROPOSTA

Este trabalho propõe a aplicação da Otimização por Colônia de Formigas (ACO) ao Problema de Alocação de Berços (PAB). Como visto, no Capítulo 2, um algoritmo ACO precisa de um grafo de construção que modela o problema que deseja-se resolver. Portanto, o PAB deve ser mapeado para uma representação que pode ser usada pelas formigas artificiais para criarem suas soluções.

### 4.1. Modelagem computacional do PAB

Para que um algoritmo ACO construa soluções é necessário que se conceba uma modelagem computacional do problema em questão. As formigas artificiais da metaheurística ACO fazem percursos em um grafo, e estes percursos representam soluções para o problema. Portanto, deve ser criado um grafo de construção, tal que, um determinado percurso nesse grafo represente uma solução para o PAB. Este capítulo descreve o grafo que foi concebido para modelar o PAB heterogêneo, e, por conseguinte, representar um espaço de busca em que a metaheurística buscará soluções.

Antes de descrever a modelagem computacional faz-se necessário explanar algumas decisões feitas com relação ao processo de construção de soluções para o PAB e, por conseguinte, à modelagem computacional deste.

#### 4.1.1. Decisões de projeto

Uma vez que este trabalho se propõe a resolver o PAB heterogêneo, construir uma solução para este problema, como visto na Seção 3.1, implica em tomar, para cada navio, duas decisões: (1) em qual berço atracar; (2) em qual TTW atracar. Tendo isso em vista, este trabalho propõe que essas duas decisões sejam feitas através de uma única. Mais precisamente, somente a decisão (1) é realizada, enquanto a decisão (2) é deduzida a partir de (1). Isto pode ser feito porque a sequencia de alocação de navios a marés pode ser deduzida de uma sequencia de alocações de navios a berços. Pois, com o berço escolhido, o tempo de carregamento/descarregamento (*handling time*) e a maré de chegada, é possível saber exatamente em que maré inicia e termina o atendimento de cada navio. Obviamente esta decisão, diminui o espaço de soluções possíveis, no entanto as

soluções encontradas foram satisfatórias, como será mostrado no Capítulo 5, onde os resultados deste trabalho são descritos.

A segunda decisão tomada diz respeito a ordem em que os navios são alocados. Foi decidido por criar uma ordem de prioridade com base na ordem de chegada. Assim a ordem de chegada dos navios define uma fila de navios, onde quem chega primeiro é alocado primeiro, ou seja, foi seguido um esquema FIFO (First In, First Out) ou FCFS (First-come, first-served). Vale notar que, este esquema não implica necessariamente que a ordem de atracação obedece a ordem de chegada. Formalmente se o navio  $i$ , chegou antes do navio  $j$ , ou seja  $a_i < a_j$  isto não implica que  $b_i < b_j$ . Esta ordem de prioridade estabelece somente a ordem em que os navios serão contemplados com um berço para atracar durante a construção da solução. A seguir temos a descrição do grafo de construção concebido para representar o PAB, de forma que as formigas artificiais da ACO possam construir soluções.

## 4.2. ACO aplicada ao PAB

Há duas questões comuns relacionadas a todas as metaheurísticas iterativas: a representação de soluções manipulada pelos algoritmos e a definição da função objetivo que vai orientar a pesquisa (TALBI, 2009). Estas questões estão extremamente ligadas ao problema que se pretende resolver, portanto serão aqui tratadas de forma relacionada ao PAB. Outra ponto importante é como as restrições do problema são implementadas. Além dessas questões comuns as metaheurísticas em geral, existe ainda uma questão referente à aplicação metaheurística ACO a um determinado problema que deve ser tratada, que é a função que define a informação heurística. Nesta seção descreveremos como essas questões foram tratadas neste trabalho.

### 4.2.1. Função Objetivo

A função objetivo  $f$  formula a meta antigir. Ela associa a cada solução do espaço de busca um valor real que descreve a qualidade ou *fitness* da solução,  $f: S \rightarrow \mathbb{R}$ . Então ela representa um valor absoluto e permite uma ordenação de todas as soluções do espaço de busca (TALBI, 2009). A função objetivo é um importante elemento no projeto de um metaheurística, porque ela guiará a busca para soluções

boas do espaço de busca. Na modelagem do PAB que foi apresentada boas soluções são as que minimizam o somatório dos tempos de serviço de todos os navios. A função objetivo deste problema é definida pela equação (3.5) descrita na Seção 3.2 do Capítulo 3 onde a modelagem matemática do problema é descrita.

#### 4.2.2. Representação

O PAB é modelado como um grafo acíclico direcionado  $G = (V, A)$ , onde  $V = N \times L$  (o conjuntos  $N$  e  $L$  estão descritos na Seção 3.1 do Capítulo 3), assim, o par ordenado  $(i, l) \in V$  representa a alocação do navio  $i \in N$  ao berço  $l \in L$ . A Figura 7 ilustra esse produto cartesiano.  $A = E \subset V \times V$ , onde  $E = \{(u, v) \mid u = (i_1, l_1), v = (i_2, l_2) \in V, i_1 \neq i_2 \text{ e } i_2 = i_1 + 1\}$ , ou seja,  $E$  é o conjunto de arestas entre dois vértices que não tem abcissa iguais e abcissa do segundo elemento do par difere em uma unidade da abcissa do outro. Essa definição de  $A$  implica que não será permitido a alocação de um navio em mais de um berço. Além disso, estabelece uma fila de prioridades baseada na ordem de chegada dos navios. Um percurso nesse grafo inicia em um dos vértices com abcissa  $i = 1$ , ou seja, inicia com alocação do primeiro navio, e termina em um dos vértices de abcissa  $i = |N|$ , ou seja, termina com alocação do ultimo berço. Assim ao passarem em um vértice desse grafo as formigas artificiais estarão tomando a decisão de atracar um determinado navio a um determinado berço. E ao completarem um percurso, terão construído uma solução para o PAB heterogêneo. A Figura 9 ilustra esse grafo e as possibilidades de movimentação das formigas nele.

É importante notar que um percurso nesse grafo é uma sequência de nós que representa uma solução. Entretanto a solução para o modelo do PAB descrito no Capítulo 3, é uma matriz de 0's e 1's. Assim é necessário que se tenha um procedimento que converta os percursos feitos pelas formigas em uma matriz de 0's e 1's. Ora, mas este procedimento é trivial, devido a modelagem, basta colocar 1 na posição da matriz que tem índices iguais às coordenadas de algum dos vértices visitados. A figura 8 ilustra essa transformação. Assim após esse processo tem-se os valores da variável  $u_{il}$  para cada navio  $i$  e berço  $l$ . Mas ainda falta atribuir o valores da variável  $x_{ij}$ . Isto foi implementado no procedimento descrito a seguir:

**procedimento**

seja  $bf$  um estrutura de dados que guarda a primeira TTW livre de cada berço  $l$ .

**para**  $i \in N$  **faça**

**para**  $l \in L$  **faça**

**se**  $u[i][l] = 1$  **então**

**para**  $j$  de  $bf[l]$  até  $bf[l] + h[i][l]$  **faça**

$$x[i][j] = 1$$

**fim-para**

$$bf[l] = bf[l] + h[i][l]$$

**fim-se**

**fim-para**

**fim-para**

**fim-procedimento**

Figura 7: Procedimento que deriva  $x_{ij}$  de  $u_{il}$

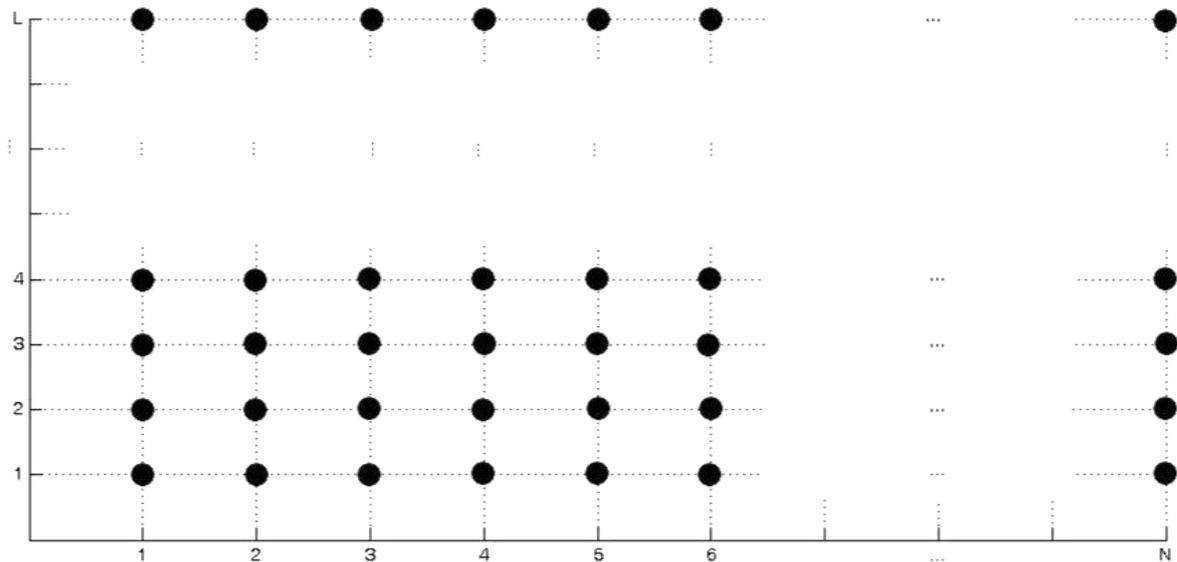
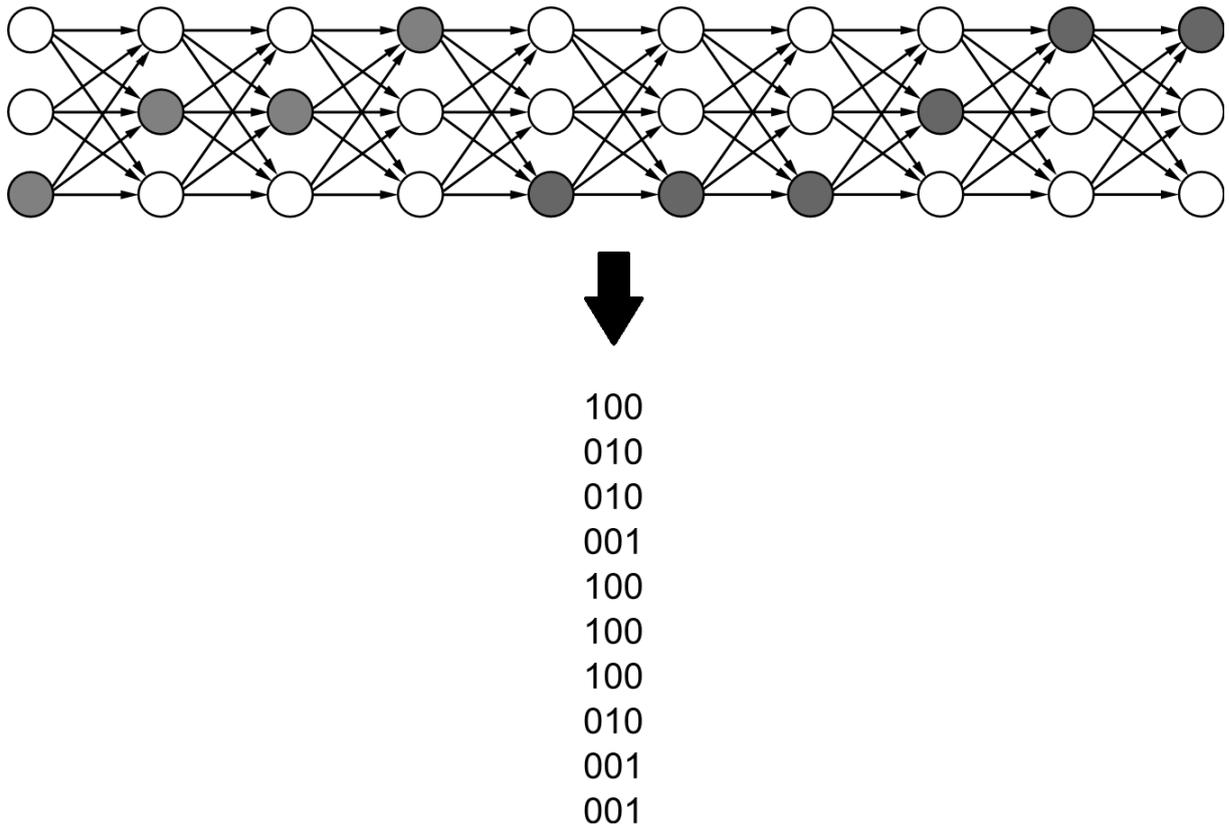


Figura 8: Produto cartesiano  $N \times L$



**Figura 9: Transformação de um percurso no grafo para uma matriz de navios x berços. Os vértices marcados são os vértices visitados**

Evidentemente este procedimento não contempla o espaço de busca inteiro, portanto no processo de construção de solução pelas formigas somente algumas soluções são consideradas. No entanto, julgou-se por bem implementar esta estratégia uma vez que com ela se tivermos berços livres, os navios ainda não atendidos não ficarão esperando para serem atendidos, o que diminui seu tempo de serviço. Além disso com este procedimento, soluções com navios que são atendidos em marés  $j$  fora do conjunto  $M$  ( $j > |M|$ ) podem ser construídas. Para induzir o algoritmo a construir soluções diferentes dessas, foram estabelecidas penalidades para soluções deste tipo. Como é mostrado no Capítulo 5, esta estratégia foi validada, mostrando bons resultados na maioria dos casos de teste.

### 4.2.3. Restrições

Outro aspecto da representação do problema é como serão implementadas as restrições. De alguma forma o algoritmo tem que construir ou buscar soluções que não violem as restrições do problema. Isto pode ser feito por meio de várias estratégias como, por exemplo, a penalização por restrição violada. A seguir tem-se

a descrição de como as restrições do PAB heterogêneo foram aqui implementadas. As restrições 3.6 a 3.10 foram implementadas ou simplesmente obedecidas via estratégia de rejeição que é não permitir que o algoritmo construa soluções inviáveis durante a busca. Ou seja, o ACS implementado só constrói soluções viáveis com relação a essas restrições. A restrição 3.11 que é a restrição de estoque mínimo, foi implantada pela estratégia de penalização. Assim soluções que implicam o na ultrapassagem do limite mínimo de estoque podem ser permitidas, no entanto, estas sofrem uma penalização em seu valor de *fitness* para que o algoritmo busque soluções melhores que essas. A penalidade atribuída é  $n \times |M|$ , onde  $n$  é o número de TTW's em que o nível mínimo de estoque é violado. As demais restrições 3.12 a 3.14 são obedecidas pela própria forma como a solução do problema é representada computacionalmente e como essa solução é decodificada a partir dessa representação.

#### 4.2.4. Informação Heurística

Para evitar que a busca de soluções das formigas seja completamente aleatória no início da execução do algoritmo, além do feromônio, a metaheurística possui um parâmetro que guia as formigas em seus percursos, este parâmetro é a informação heurística. A informação heurística codifica um conhecimento a priori que se tem do problema e indica os vértices ou ligações entre vértices que são considerados bons segundo determinado critério. Por exemplo, na aplicação do ACO ao clássico Problema do Caixeiro Viajante, a informação heurística com relação a aresta que liga o vértices  $u$  e  $v$  é  $\eta_{uv} = \frac{1}{d_{uv}}$ , onde  $d_{uv}$  é a distância entre  $u$  e  $v$ .

Para aplicação ao PAB pensou-se em duas possibilidades de definição para função que calcula a informação heurística. Falando nos termos referentes ao PAB, a primeira possibilidade é considerar que a alocação de um navio  $i$  a um berço  $l$  é boa quando o tempo de serviço  $h_{il}$  é pequeno. Assim a informação heurística fica definida pela função a seguir.

$$\eta_{il} = (h_{il})^{-1} \quad (4.1)$$

Isto traduz a inferência lógica de que quanto menor for o tempo de serviço de cada navio, menor será o somatório de todos. Ou seja, a informação heurística estabelece

que a sequencia de escolhas que implicam em tempos de atendimento pequenos, será o uma boa solução para o PAB. Este tipo de informação heurística seria o paralelo a informação heurística usada na aplicação do ACO ao Problema do Caixeiro Viajante. No entanto para este problema (PAB) este tipo de informação não se mostrou eficiente, pois como será visto no capítulo 5, a sua implementação apresentou resultados ruins.

Outra possibilidade seria considerar além do tempo de serviço  $h_{il}$ , as marés que os berços ficam livres. Com essa estratégia, na escolha de um berço, além de ser levado em consideração as velocidade de carregamento/descarregamento também é considerado a primeira maré em que o berço está livre, considerando que o berço que fica livre mais cedo, é melhor.

$$\eta_{il} = (FTTWF_l + h_{il})^{-1} \quad (4.2)$$

Onde  $FTTWF_l$  (First Tidal Time Window Free) é a primeira janela de maré livre no berço  $l$ . Esta estratégia tenta criar possibilidade de que berços mais lentos, que, no entanto são liberados mais cedo, sejam escolhidos, e assim pode-se evitar que berços lentos fiquem sem uso durante a operação de atendimento.

#### 4.2.5. Definição dos parâmetros

Para aplicar o ACS é necessário definir seus parâmetros. A tabela 1 mostra a configuração dos parâmetros usados em todos os testes de desempenho do algoritmo.

Parâmetro	Valor
$m$	10
$\beta$	10
$\rho$	0.6
$\xi$	0.1
$q_0$	0.6

Tabela 1: Configuração dos parâmetros do algoritmo

## 5. RESULTADOS

### 5.1. Ambiente de testes

Para comparar com os resultados alcançados pela metaheurística ACO, foi usado os resultados obtidos por Victor H. Barros (2010), ao utilizar o *solver* comercial CPLEX para derivar soluções exatas a partir do modelo matemático do PAB descrito no Capítulo 3 (BARROS, 2009). Além desses resultados usaremos para comparar uma heurística gulosa apresentada em (COSTA et al., 2012).

A metaheurística desenvolvida neste trabalho foi implementada com linguagem de programação C++. O ambiente de desenvolvimento usado foi o *Microsoft Visual C++ 2012 Express Edition*.

### 5.2. Instâncias

Em (BARROS et al, 2009) para validar o modelo do PAB proposto e que foi descrito no Capítulo 3 deste trabalho, foram feitos testes com dados gerados de forma semi-aleatória, correspondendo a possíveis cenários. A maior parte dos dados foi gerada aleatoriamente, mas com características que determinam instâncias de difícil resolução, onde cada conjunto de restrições tem influência na solução. Neste trabalho algumas dessas instâncias foram usadas para efetuar os experimentos. Foram usadas instâncias de 10, 15, 20 e 30 navios, variando o número de berços.

### 5.3. Descrição dos resultados

A seguir temos duas tabelas que mostram os resultados obtidos na aplicação do ACS para as instâncias do PAB. As tabelas indicam o nome, a quantidade de navios, berços e marés de cada instância, bem como valor alcançado pelo ACS implementado neste trabalho. Para efeito de comparação, as tabelas também mostram o valor ótimo encontrado pelo CPLEX, e o valor obtido pela heurística apresentada em (COSTA et al., 2012).

A primeira tabela mostra os resultados obtidos com o ACS implementado com a primeira função que descreve a informação heurística, proposta na Seção 4.3.1 do Capítulo 4. A segunda tabela mostra os resultados com o ACS implementado com segundo tipo de informação heurística proposto neste trabalho, que está descrito na

Seção 4.3.2. Cada resultado obtido é a média dentre 30 execuções do ACS proposto para a determinada instancia.

<b>Instância</b>	<b>Navios</b>	<b>Berços</b>	<b>TTW</b>	<b>Ótimo</b>	<b>Heurística</b>	<b>ACO</b>
10-15-3	10	3	15	31	-	34
10-15-4	10	4	15	21	-	23
10-20-3	10	3	20	44	45	50
10-25-2	10	2	25	43	46	51
15-30-4	15	4	30	77	77	90
15-35-3	15	3	35	86	101	109
15-40-2	15	2	40	121	144	176
20-40-3	20	3	40	108	128	171
20-40-4	20	4	40	105	123	176
20-55-2	20	2	55	297	318	371
30-70-4	30	4	70	-	230	480
30-75-3	30	3	75	190	229	432
30-80-2	30	2	80	-	679	719 <sup>1</sup>

**Tabela 2: Resultados obtidos com a implementação do primeiro tipo de informação heurística proposto**

<b>Instância</b>	<b>Navio</b>	<b>Berço</b>	<b>TTW</b>	<b>Ótimo</b>	<b>Heurística</b>	<b>ACO</b>
10-15-3	10	4	15	31	-	31
10-15-4	10	4	15	21	-	21
10-20-3	10	3	20	44	45	44
10-25-2	10	2	25	43	46	43
15-30-4	15	4	30	77	77	78
15-35-3	15	3	35	86	101	87

<sup>1</sup> Esta solução apresentou violação de estoque.

15-40-2	15	2	40	121	144	125
20-40-3	20	3	40	107	128	111
20-40-4	20	4	40	108	123	108
20-55-2	20	2	55	297	318	335
30-70-4	30	4	70	-	230	203
30-75-3	30	3	75	190	229	233
30-80-2	30	2	80	-	679	725 <sup>2</sup>

**Tabela 3: Resultados obtidos com a implementação do segundo tipo de informação heurística proposto**

Na Tabela 3, tem-se em destaque os resultados em que o algoritmo implementado foi melhor que a heurística gulosa.

---

<sup>2</sup> Esta solução apresentou violação de estoque.

## 6. CONCLUSÃO

Neste trabalho uma alternativa para a resolução do Problema de Alocação de Berços (PAB) foi apresentada. Para isso foi proposta a aplicação da Metaheurística Otimização por Colônia de Formigas (ACO). Assim a ACO, nesta aplicação, escolherá para um conjunto de navios, em qual berço e maré cada navio atracará, com o objetivo de otimizar o tempo de atendimento, respeitando as restrições do problema, em especial as restrições de estoque.

### 6.1. Contribuições

Como contribuição, a implementação da ACO para o PAB, seguiu uma das estratégias promissoras desenvolvidas para a metaheurística ACO, denominada *Ant Colony System* (ACS). Os resultados obtidos com esta implementação se mostraram satisfatórios.

Este trabalho também propõe duas formas de informação heurística, componente importante da ACO, para a aplicação específica ao PAB. A primeira considera, no seu cálculo, somente o *handling time*, já a segunda considera, além do *handling time*, as TTW's em que os berços. A primeira proposta apresentou resultados ruins, se comparados aos resultados da heurística gulosa. A segunda proposta se satisfatória, possibilitando que bons resultados fossem alcançados. No entanto na instância 30-80-2, as duas versões do algoritmo, criaram soluções inviáveis, pois violam a restrição de estoque. Portanto conclui-se que melhorias precisam ser feitas na proposta. Uma das melhorias possíveis é o melhor ajuste dos parâmetros.

A Tabela 3 nos mostra, que quando comparado a heurística gulosa proposta em (COSTA et al., 2012), o algoritmo ACO implementado, com o segundo tipo de informação heurística proposto, mostrou melhores resultados em 7 dos 11 casos de testes. Portanto, o algoritmo ACO implementado, com alguns ajustes, se mostra como uma alternativa promissora para problemas de Pesquisa Operacional, semelhantes ao PAB.

Diante do exposto, pode-se concluir que proposta apresentada cumpriu os objetivos deste trabalho. E, além disso, pode-se dizer que metaheurística ACO aplicada a problemas práticos semelhantes ao PAB trará benefícios, tendo em vista

a qualidade das soluções encontradas. E dado às definições feitas nesta proposta, que implicam em limitações no processo de busca da metaheurística, tornando o espaço de soluções menor, pode-se inferir que outras definições podem expandir o espaço de busca, tornando o algoritmo mais eficiente. Assim, este trabalho apresenta-se como ponto de partida para exploração da aplicação do ACO ao PAB e problemas semelhantes.

## **6.2. Trabalhos Futuros**

Este trabalho é ainda incipiente, se caracterizando como um ponto de partida para inúmeras possibilidades. Como trabalhos futuros tem-se:

1. Comparação do ACO com outras Metaheurísticas existentes na literatura, na aplicação ao PAB.
2. Proposta de outras definições na concepção do processo de construção de soluções, que permitam que o algoritmo explore regiões ainda não exploradas do espaço de busca.
3. Fazer o ajuste dos parâmetros com base em testes estatísticos.
4. Explorar o paralelismo de algoritmos e tornar o ACO competitivo com heurísticas simples, que perdem em termos de qualidade da solução, no entanto são mais eficientes em termos de tempo.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

- BARROS, V. H. (2009). Problema de alocação de berços heterogêneos em portos graneleiros com restrições de estoque e condições favoráveis de maré. *Simpósio de Engenharia de Produção*.
- BARROS, V. H. (22 de março de 2010). Problema de Alocação de Berços em Portos Graneleiros com restrições de estoque e condições de maré. São Luis, Universidade Federal do Maranhão.
- BECENARI, J. C., STEPHANY, S., VELHO, H. F., & NETO, A. S. (s.d.). *Otimização por Colônia de Formigas (Ant Colony Optimization)*. Acesso em 10 de Novembro de 2013, disponível em <http://mtc-m19.sid.inpe.br/col/sid.inpe.br/mtc-m19@80/2010/01.20.19.27/doc/cap7.pdf>
- DORIGO, M., & DI ICARO, G. (6-9 de Julho de 1999). Ant colony optimization: a new meta-heuristic. *Evolutionary Computation*, 3.
- DORIGO, M., & SOCHA, K. (2004). *Ant Colony Optimization*. Cambridge, Massachusetts, U.S.A: MIT Press.
- DORIGO, M., & SOCHA, K. (2007). An Introduction to Ant Colony Optimization. In: T. F. GONZALEZ, *Handbook of Approximation Algorithms and Metaheuristics* (pp. 401-411). Santa Barbara, U.S.A.: Chapman & Hall/CRC.
- DORIGO, M., & STÜTZLE, T. (2003). The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. In: F. GLOVER, & G. KOCHENBERGER, *Handbook of Metaheuristics* (pp. 251-285). Dordrecht: Kluwer Academic Publishers.
- GAERTNER, D., & CLARK, K. (2005). On Optimal Parameters for Ant Colony Optimization Algorithms. *Proceedings of the International Conference on Artificial Intelligence*.
- HAMMERL, T. (24 de Março de 2009). Ant Colony Optimization for Tree and Hypertree Decompositions. Viena, Viena University of Technology.
- MAURI, G., LORENA, L., & OLIVEIRA, A. C. (2010). Resolução do problema de alocação de berços através de uma técnica de geração de colunas. *Pesquisa Operacional*, 547-562.
- MAURI, G., LORENA, L., & OLIVEIRA, A. (2008). Heurística baseada no simulated annealing aplicada ao problema de berços. *GEPROS - Gestão da Produção, Operações e Sistemas*, 1, 113-127.
- OLIVEIRA, R. M., MAURI, G., & LORENA, L. A. (30 de Agosto a 3 de Setembro de 2010). Clustering search aplicado ao problema de alocação de berços. *XLII SBPO - Simpósio Brasileiro de Pesquisa Operacional*. Bento Gonçalves, RS.

- SILVA, V., & COELHO, A. (agosto de 2007). Uma visão sobre o Problema de Alocação de Berços. *Revista Produção*, 7, 85-98.
- SILVEIRA, T. (30 de Junho de 2010). A Utilização da Meta-Heurística de Otimização denominada Colônia De Formigas para o Restabelecimento de Sistemas Elétricos de Potência. Alfenas.
- SOCHA, K. (2004). Ant Colony Optimization for Continuous and Mixed-Variable Domains. *Lecture Notes in Computer Science Volume 3172*, pp. 25-36.
- TALBI, E.-G. (2009). *Metaheuristics: From Design to Implementation*. Hoboken, New Jersey: WILEY.
- VALDECY, V. J., SANTOS, R., COSTA, T., & OLIVEIRA, A. C. (Setembro de 2012). Greedy Heuristic for Berth Allocation in Tidal Bulk Ports. *In proceeding of: Jornada de Informática do Maranhão (JIM 2012)/Escola Regional de Computação dos Estados do Ceará, Maranhão e Piauí (ERCEMAPI 2012)*.
- YANG, X.-S. (2008). *Introduction to Mathematical Optimization: From Linear Programming to Metaheuristics*. Cambridge, United Kingdom: Cambridge International Science Publishing.