



Programa Interdisciplinar de Pós-Graduação em

**Computação Aplicada**

**Mestrado Acadêmico**

Luziana Ferronato Carraro

Uma Abordagem para o Problema de Carregamento de  
Navios-Contêineres através do Emprego de Metaheurísticas  
baseadas na Codificação por Regras

São Leopoldo, 2013

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS  
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO  
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO  
EM COMPUTAÇÃO APLICADA  
NÍVEL MESTRADO

LUZIANA FERRONATTO CARRARO

Uma Abordagem para o Problema de Carregamento de Navios-  
Contêineres através do Emprego de Metaheurísticas baseadas na  
Codificação por Regras

São Leopoldo

2013

Luziana Ferronato Carraro

Uma Abordagem para o Problema de Carregamento de Navios-  
Contêineres através do Emprego de Metaheurísticas baseadas na  
Codificação por Regras

Dissertação apresentada como requisito parcial  
para a obtenção do título de Mestre, pelo  
Programa Interdisciplinar de Pós-Graduação  
em Computação Aplicada da Universidade do  
Vale do Rio dos Sinos – UNISINOS

Orientador: Prof. Dr. Leonardo Dagnino Chiwiacowsky

São Leopoldo

2013

C313a Carraro, Luziana Ferronato

Uma abordagem para o problema de carregamento de navios-contêineres através do emprego de metaheurísticas baseadas na codificação por regras / por Luziana Ferronato Carraro. -- São Leopoldo, 2013.

112 f. : il. ; 30 cm.

Dissertação (mestrado) – Universidade do Vale do Rio dos Sinos, Programa Interdisciplinar de Pós-Graduação em Computação Aplicada, São Leopoldo, RS, 2013.

Orientação: Prof. Dr. Leonardo Dagnino Chiwiacowsky, Ciências Exatas e Tecnológicas.

1.Metaheurística – Transporte marítimo. 2.Navios-contêineres – Cargas e descargas. 3.Algoritmos genéticos – Transporte por containers. 4.Enxame de abelhas – Transporte por containers. 5.Navios de carga – Metaheurística. I.Chiwiacowsky, Leonardo Dagnino. II.Título.

CDU 004.421:656.61

004.421:656.073.235

656.614.3:004.421

Catálogo na publicação:  
Bibliotecária Carla Maria Goulart de Moraes – CRB 10/1252

Luziana Ferronato Carraro

Uma Abordagem para o Problema de Carregamento de Navios-  
Contêineres através do emprego de Metaheurísticas baseadas na  
Codificação por Regras

Dissertação apresentada como requisito parcial  
para a obtenção do título de Mestre, pelo  
Programa Interdisciplinar de Pós-Graduação  
em Computação Aplicada da Universidade do  
Vale do Rio dos Sinos – UNISINOS

BANCA EXAMINADORA

---

Prof. Dr. Alexandre Cesar Muniz de Oliveira

---

Prof. Dr. Arthur Tórgo Gómez

---

Prof. Dr. Leonardo Dagnino Chiwiacowsky

São Leopoldo

2013

*A minha família, meus pais Nelso e Maria Lúcia,  
meu irmão Matheus, por todo o apoio e força.*

*Ao meu querido marido André pela compreensão e  
incentivo.*

*"Nunca sabemos do que somos capazes até  
o tentarmos fazê-lo." (Charles Dickens)*

## **AGRADECIMENTOS**

Em especial agradeço ao meu orientador Prof. Dr. Leonardo Dagnino Chiwiacowsky, pela paciência em meu aprendizado, principalmente na parte da programação, como também pela disponibilidade e sabedoria compartilhada.

Aos professores, que contribuíram em meu aprendizado.

À CAPES, pelo apoio financeiro.

A minha Família, por todo o incentivo, força e coragem.

Ao meu esposo André, pela compreensão e apoio em todos os momentos.

## RESUMO

Com a expansão do transporte marítimo, passou a ser adotado o uso de contêineres para o transporte de cargas, sendo evidenciados alguns problemas. Dentre eles, um dos principais, é o problema de carregamento e descarregamento de contêineres em navios. O problema surge devido aos altos custos operacionais gerados a partir da movimentação de contêineres. Este problema é o foco desta pesquisa, que tem como objetivo principal elaborar planos de carga eficientes que gerem um número mínimo de movimentações de contêineres, nas operações de carga e descarga de navios-contêineres, diminuindo assim os custos de operação. Neste trabalho, é proposta a aplicação da metaheurística Algoritmo Genético e da metaheurística Enxame de Abelhas, resolvendo o problema através de uma codificação baseada em regras de carregamento e descarregamento. A codificação por regras é compacta e adequada, assegurando que as soluções do problema sejam factíveis e de simples representação, acelerando o processo de solução. Nos experimentos realizados, as duas metaheurísticas foram empregadas, assumindo diferentes configurações de regras, com o intuito de comparar o seu desempenho. A proposta de novas regras de carregamento e descarregamento, em complemento às existentes na literatura, trouxeram bons resultados. Desta forma, foram obtidas soluções de boa qualidade e melhores que aquelas encontradas na literatura que abordam o mesmo problema.

**Palavras-Chave:** Carregamento de Navios-Contêineres. Algoritmo Genético. Enxame de Abelhas. Otimização Combinatória.

## ABSTRACT

With the expansion of maritime transportation, the use of containers for goods transportation has increased, being evidenced some problems. Among these problems, the container ship stowage problem arose as one of the main problems due to the high operational costs related to movement of containers. This problem is the focus of this research, where the main objective is the formulation of stowage plans that generate a minimum number of container shiftings in the operations of loading and unloading performed in port calls of container ships. In order to determine a suitable stowage plan, the application of Genetic Algorithm and Bee Swarm Optimization metaheuristics are proposed to solve the problem by using a rule-based encoding for the solution. The solution encoding based on loading and unloading rules is compact and suitable, ensuring the feasibility of solutions and also the simple representation of it, speeding up the solution procedures. In the performed experiments, both metaheuristics were applied assuming different rules settings with the objective to compare each performance. The proposal of new rules of loading and unloading, in addition with those existing in literature, has produced good solutions. Thereby, good quality solutions were achieved and also better than that found in the literature which discuss the same problem

**Keywords:** Container Ship Stowage. Genetic Algorithms. Bees Swarm. Combinatorial Optimization.

## Lista de Abreviaturas

ABC	<i>Artificial Bee Colony</i>
AG	Algoritmos Genéticos
BCO	<i>Bee Colony Optimization</i>
BS	<i>Bee System</i>
BSO	<i>Bee Swarm Optimization</i>
FO	Função Objetivo
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
HBA	<i>Honey Bee Algorithm</i>
NP	Classe de problemas de tempo polinomial não-determinísticos
OX	<i>Order Crossover</i>
PAB	Problema de Alocação de Berços
PC	Probabilidade de Cruzamento
PCC	Problema de carregamento de contêineres
PM	Probabilidade de Mutação
SB	<i>Swarm Bees</i>
TEU	<i>Twenty-foot Equivalent Units</i>
VBA	<i>Virtual Bee Algorithm</i>

## LISTA DE FIGURAS

Figura 1 - Domínio do Transporte Marítimo em Exportações .....	18
Figura 2 - Terminal Portuário .....	22
Figura 3 - Estrutura celular de um navio-contêiner .....	24
Figura 4 - Fluxograma AG .....	27
Figura 5 - Coleta de Alimento .....	32
Figura 6 - Abelhas no processo de coleta de alimento .....	34
Figura 7 - Fluxograma ABC .....	35
Figura 8 - Matriz de Ocupação Inicial .....	45
Figura 9 - Matriz de Transporte .....	47
Figura 10 - Matriz de Ocupação após Regra C1 .....	48
Figura 11 - Matriz de Ocupação da Regra C2 .....	48
Figura 12 - Matriz de Ocupação da Regra C3 .....	49
Figura 13 - Matriz de Ocupação da Regra C4 .....	49
Figura 14 - Matriz de Ocupação da Regra C5 .....	49
Figura 15 - Matriz de Ocupação da Regra C6 .....	50
Figura 16 - Antes e Depois aplicação da regra D1 .....	50
Figura 17 - Antes e Depois aplicação da regra D2 .....	51
Figura 18 - Antes e Depois aplicação da regra D3 .....	51
Figura 19 - Codificação do Cromossomo .....	52
Figura 20 - Pai 1 e Pai 2 para o cruzamento .....	53
Figura 21 - Indivíduo após Cruzamento Uniforme .....	53
Figura 22 - Operador Um Ponto de Corte .....	54
Figura 23 - Mutação Uniforme .....	54
Figura 24 - Operador Inversão de uma Subsequência .....	56
Figura 25 - Operador Troca Aleatória .....	56
Figura 26 - Métodos aplicados com 4 regras - Matriz tipo 1 .....	64
Figura 27 - Métodos aplicados com 4 regras - Matriz tipo 2 .....	64
Figura 28 - Métodos aplicados com 4 regras - Matriz tipo 3 .....	65
Figura 29 - Comparativo das melhores médias de resultados de cada cenário referente à ....	88
Figura 30 - Comparativo das melhores médias de resultados de cada cenário referente à ....	89
Figura 31 - Comparativo das melhores médias de resultados de cada cenário referente à ....	90
Figura 32 - Comparativo entre AG's Matriz 1 .....	91

Figura 33 - Comparativo entre AG's Matriz 2 .....	92
Figura 34 - Comparativo entre AG's Matriz 3 .....	93
Figura 35 - Evolução das soluções do AG .....	93
Figura 36 - Evolução das soluções do ABC .....	99
Figura 37 - AG x ABC (Matriz 1) .....	100
Figura 38 - AG x ABC (Matriz 2) .....	100
Figura 39 - AG x ABC (Matriz 3) .....	101

## Lista de Tabelas

Tabela 1 - Definição das Combinações das Regras de Movimentação de Contêineres .....	46
Tabela 2 - Movimentos Mínimos de contêineres .....	59
Tabela 3 - Resultados Médios do AG de Azevedo.....	60
Tabela 4 - Resultados Médios do AG.....	61
Tabela 5 - Resultados médios do ABC.....	62
Tabela 6 - Cenários do AG .....	66
Tabela 7 - Cenários do ABC .....	66
Tabela 8 - Cenário 1– AG 10 Portos .....	69
Tabela 9 - Cenário 1– AG 15 Portos .....	70
Tabela 10 - Cenário 1– AG 20 Portos .....	71
Tabela 11 - Cenário 1- AG 25 Portos .....	73
Tabela 12 - Cenário 1 – AG 30 Portos .....	74
Tabela 13 - Cenário 2- AG 10 Portos .....	76
Tabela 14 - Cenário 2- AG 15 Portos .....	77
Tabela 15 - Cenário 2- AG 20 Portos .....	78
Tabela 16 - Cenário 2- AG 25 Portos .....	79
Tabela 17 - Cenário 2- AG 30 Portos .....	80
Tabela 18 - Cenário 3- AG 10 Portos .....	82
Tabela 19 - Cenário 3- AG 15 Portos .....	82
Tabela 20 - Cenário 3- AG 20 Portos .....	83
Tabela 21 - Cenário 3- AG 25 Portos .....	83
Tabela 22 - Cenário 3- AG 30 Portos .....	84
Tabela 23 - Cenário 4- AG 10 Portos .....	85
Tabela 24 - Cenário 4- AG 15 Portos .....	86
Tabela 25 - Cenário 4- AG 20 Portos .....	86
Tabela 26 - Cenário 4- AG 25 Portos .....	87
Tabela 27 - Cenário 4- AG 30 Portos .....	87
Tabela 28 - ABC 10 Portos .....	94
Tabela 29 - ABC 15 Portos .....	95
Tabela 30 - ABC 20 Portos .....	96
Tabela 31 - ABC 25 Portos .....	97
Tabela 32 - ABC 30 Portos .....	98
Tabela 33 - GAP.....	102
Tabela 34 - AG - 18 combinações de regras .....	103

Tabela 35 - Movimentos mínimos para 1000 contêineres.....	104
Tabela 36 - Resultados Médios do AG para 1000 contêineres.....	105
Tabela 37 - GAP para AG 1000 contêineres .....	106
Tabela 38 - Melhor Cenário de cada Metaheurística.....	109

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>18</b>
1.1 OBJETIVO.....	19
1.2 <i>A Estrutura do Trabalho</i> .....	20
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>22</b>
2.1 O PROBLEMA .....	22
2.2 MÉTODOS DE SOLUÇÃO .....	25
2.2.1 <i>Algoritmo Genético (AG)</i> .....	26
2.2.2 <i>Enxame de Abelhas (Swarm Bees - SB)</i> .....	32
<b>3 REVISÃO BIBLIOGRÁFICA.....</b>	<b>38</b>
3.1 PROBLEMAS RELACIONADOS A OPERAÇÕES EM TERMINAIS PORTUÁRIOS .....	38
3.2 O PROBLEMA DE CARREGAMENTO DE CONTÊINERES RESOLVIDO POR AG .....	39
3.3 OUTRAS TÉCNICAS APLICADAS AO PROBLEMA DE CARREGAMENTO DE CONTÊINERES .....	40
3.4 A TÉCNICA DO ENXAME DE ABELHAS APLICADA A OUTROS PROBLEMAS .....	42
<b>4 METODOLOGIA .....</b>	<b>43</b>
4.1 MODELO MATEMÁTICO .....	43
4.2 FERRAMENTAS COMPUTACIONAIS.....	45
4.2.1 <i>Representação por Regras</i> .....	45
4.2.2 <i>A metaheurística Algoritmo Genético</i> .....	51
4.2.2.5 <i>Critério de Parada</i> .....	55
4.2.3 <i>A metaheurística Enxame de Abelhas</i> .....	55
<b>5 VALIDAÇÃO .....</b>	<b>57</b>
5.1 TIPOS DE MATRIZES DE TRANSPORTE .....	57
5.4 VALIDAÇÃO DOS MÉTODOS DE SOLUÇÃO .....	59
5.5 COMPARATIVO ENTRE ALGORITMOS .....	63
<b>6 EXPERIMENTOS .....</b>	<b>66</b>
6.1 CARACTERÍSTICAS COMPLEMENTARES DOS EXPERIMENTOS .....	67
6.1.1 <i>Experimentos com o Algoritmo Genético</i> .....	67

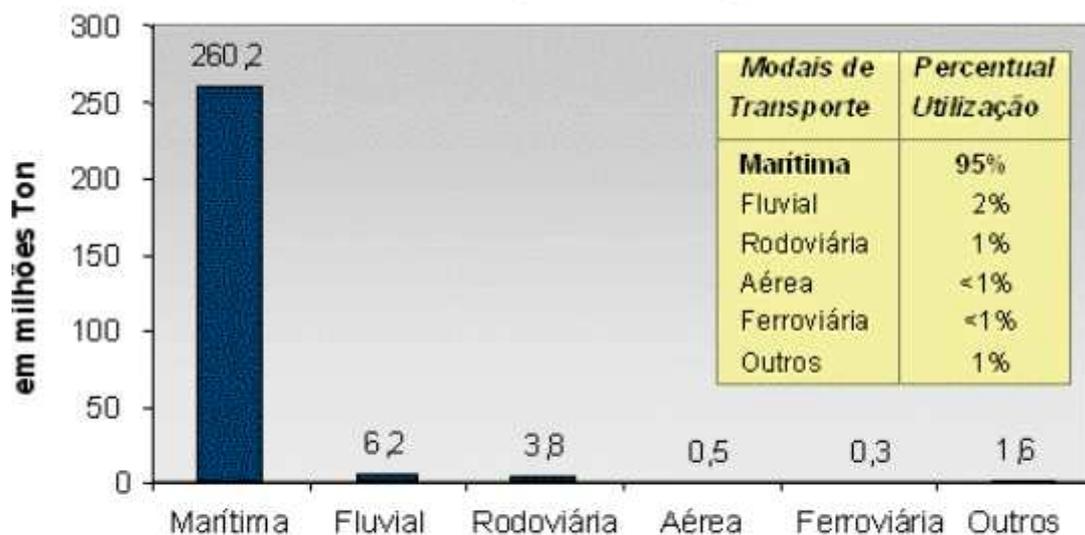
6.1.1.1 Cenário 1 .....	68
6.1.1.2 Cenário 2 .....	75
6.1.1.3 Cenário 3 .....	81
6.1.1.5 Comparativo entre cenários do AG .....	88
6.1.1.6 Comparativo entre AG's .....	91
6.1.1.7 Experimentos com o ABC (Enxame de Abelhas) .....	94
6.2 COMPARATIVO ENTRE OS MÉTODOS DE SOLUÇÃO .....	99
6.4 TESTES ADICIONAIS .....	104
6.4.1 Testes com 1000 contêineres .....	104
6.4.2 O caso com 5000 contêineres .....	106
<b>7 CONSIDERAÇÕES FINAIS .....</b>	<b>108</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>111</b>

## 1 INTRODUÇÃO

O transporte marítimo mercante há muitos séculos vem sendo utilizado como transporte de mercadorias e de passageiros. Desde sua origem, revelou-se importante pela realização de transações comerciais de grande porte, fazendo ligações entre a produção e o consumo. Porém, em meados dos anos 1960, o transporte marítimo revolucionou com o surgimento de contêineres para organização de cargas, tornando-o mais fácil, mais seguro, rápido e ainda proporcionando garantia ao transporte.

Com o passar do tempo, houve a expansão do transporte marítimo e o comércio global tornou-se cada vez mais competitivo com o uso de contêineres. A vantagem da utilização de navios-contêineres é relacionada a sua capacidade de carga, que pode alcançar até 10.000 unidades. Em estudos do BNDES, revelou-se que no Brasil 95% das exportações são por via marítima, e os outros 5% se dividem em transportes fluvial, rodoviário, aéreo, ferroviário e outros (PEDREIRA, 2006), conforme representado na Figura 1.

**Figura 1 - Domínio do Transporte Marítimo em Exportações**



Fonte: Ministério do Desenvolvimento, Indústria e Comércio Exterior - MDIC 2001  
Obtida na publicação do BNDES denominada - O Desafio das Exportações

Fonte: Pedreira (2006)

Com o aumento do emprego do transporte marítimo, diversos problemas começaram a surgir nos portos, que afetaram os custos operacionais e a qualidade produtiva. Dentre estes problemas, destaca-se a necessidade de um planejamento avançado para o carregamento de navios-contêineres em terminais portuários de contêineres. O objetivo deste planejamento é a minimização de custos, sabendo que para cada movimentação de contêineres são gerados custos aos portos e a seus usuários. Consequentemente, quanto maior o número de movimentações maior será o custo envolvido nas operações. Pensando nisso, neste trabalho é proposto o

emprego de métodos computacionais baseados em metaheurísticas, a fim de minimizar o número de movimentos de contêineres, nas operações de carga e descarga realizadas em terminais portuários de contêineres.

O navio para o transporte de contêineres possui características específicas quanto a sua estrutura e forma de carregamento. O navio porta-contêiner, ou navio-contêiner, possui uma rota pré-definida para carregar e descarregar contêineres passando por  $N$  portos. A forma de descarregar navios-contêineres é retirar os contêineres do topo de uma pilha, e caso o contêiner procurado não estiver no topo, serão realizadas remoções de contêineres até liberá-lo, realizando posteriormente, as reposições necessárias. Esta movimentação de contêineres, para liberar um contêiner a ser descarregado, também pode ser chamada de remanejamento.

Em virtude desta característica, alguns fatores devem ser levados em conta para realizar um planejamento adequado que otimize o processo de carregamento de um navio-contêiner. Inicialmente, os contêineres a serem descarregados no porto vizinho devem estar posicionados de maneira estratégica, facilitando o seu descarregamento e evitando movimentações desnecessárias que acarretariam em custos adicionais. Além do alto custo, as movimentações dependem tempo, que conseqüentemente gerariam atraso na liberação do navio ou ainda custos de sobre-estadia.

O problema de carregamento de navios-contêineres pode ser resolvido assumindo restrições adicionais como, por exemplo, condições de estabilidade, esforços estruturais do navio e outras mencionadas no próximo capítulo. Neste trabalho, será abordado o problema somente considerando a minimização do número de movimentações de contêineres.

## 1.1 Objetivo

A principal motivação para o estudo do problema apresentado é a sua importância para a economia mundial e brasileira, e também por ser um problema para o qual as soluções propostas ainda se mostram insuficientes frente à exigência das empresas de transporte marítimo.

Há diversos trabalhos na literatura que propõem o emprego de diferentes técnicas para solução do problema de carregamento de navios-contêineres, assumindo diferentes restrições e propondo modelagens variadas (AVRIEL E PENN; WILSON E ROACH; DUBROVSKY; AZEVEDO, 1993, 1999, 2002, 2010). Neste trabalho, é proposta a solução do problema de minimização de movimentação de contêineres, quando da operação de carga e descarga de navios-contêineres, através do emprego de dois métodos distintos, as metaheurísticas Algoritmo Genético e Enxame de Abelhas.

Observa-se que, com relação às técnicas adotadas, foram encontrados na literatura somente trabalhos empregando o Algoritmo Genético para solução do problema em estudo, sendo então o método do Enxame de Abelhas uma contribuição deste trabalho. Outra contribuição deste trabalho é o emprego de regras de carregamento e descarregamento de contêineres para representação das soluções. Regras para codificação de soluções para o problema de carregamento de navios contêineres foram inicialmente empregadas por Azevedo

(2010). Entretanto, no presente trabalho são propostas novas regras, em adição às aquelas apresentadas em (AZEVEDO, 2010), sendo destacadas também como uma contribuição desta dissertação.

Com base no exposto, este trabalho tem como objetivo geral a abordagem do problema de minimização das movimentações de contêineres em operações de carga e descarga de navios-contêineres, através do emprego das metaheurísticas Algoritmo Genético e Enxame de Abelhas, empregando as novas regras em conjunto com as já existentes. Os objetivos específicos desta pesquisa são:

- Desenvolver e aplicar a metaheurística Algoritmo Genético, baseada em uma representação por regras, na solução do problema de carregamento de navios-contêineres;
- Desenvolver e aplicar a metaheurística Enxame de Abelhas, baseada em uma representação por regras, na solução do problema de carregamento de navios-contêineres;
- Avaliar o desempenho de cada estratégia de solução, assumindo diferentes configurações para os parâmetros utilizados em cada um dos métodos;
- Identificar e avaliar a melhor configuração de cada estratégia de solução;
- Realizar um comparativo entre os métodos, estabelecendo seu desempenho em relação à qualidade da solução final e ao tempo de execução;
- Comparar o desempenho dos algoritmos com trabalhos encontrados na literatura.

## 1.2 A Estrutura do Trabalho

Esta dissertação está estruturada em sete capítulos. Este primeiro capítulo de Introdução, e outros seis especificados a seguir.

O capítulo 2 apresenta os conceitos básicos do problema de carregamento de navios-contêineres, sendo detalhada a estrutura do navio. Além disso, são apresentados os conceitos necessários para entender o funcionamento dos métodos de solução adotados na resolução deste problema.

No capítulo 3, uma revisão bibliográfica é apresentada, abordando vários trabalhos publicados na literatura e relacionados com as operações de terminais portuários. A pesquisa explora trabalhos que resolvem o problema, através de técnicas distintas às propostas neste estudo, como também o problema de carregamento de navios-contêineres resolvido pelo Algoritmo Genético, que é um dos algoritmos empregado neste trabalho, e ainda trabalhos abordando problemas relacionados a outras atividades portuárias. Em relação ao Enxame de Abelhas, são apresentados trabalhos encontrados na literatura que abordem o emprego de tal método em outros problemas.

Em continuação, o capítulo 4 aborda a modelagem matemática adotada para o problema e sua codificação, através do emprego de regras. Neste espaço, são apresentadas todas as regras e suas características particulares. Em seguida, conceitua-se o emprego das metaheurísticas adotadas, para após apresentar uma definição quanto aos parâmetros e operadores que serão utilizados nos algoritmos, na etapa de implementação.

O capítulo 5 é destinado à validação dos métodos. Antes da validação, são definidos os tipos de Matriz de Transporte empregados, os problemas testes que serão utilizados na validação e nos experimentos, os parâmetros usados em cada algoritmo na etapa de validação e os resultados médios obtidos com as técnicas de solução adotadas.

No capítulo 6, são efetivados os experimentos, definindo as variações dos algoritmos e os operadores utilizados, através da definição de diferentes cenários de simulação. Neste mesmo capítulo são analisados os resultados dos experimentos, realizando comparativos entre os cenários, entre as metaheurísticas e comparativos entre o número de regras utilizadas. Em complemento, é apresentada a análise de alguns testes adicionais que foram efetuados.

Por fim, no capítulo 7, são apresentadas as considerações finais deste trabalho e suas contribuições. É feita uma análise geral dos experimentos e do desempenho das metaheurísticas. Após, são propostos os trabalhos futuros que darão continuidade ao estudo realizado.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos necessários para o desenvolvimento do trabalho de pesquisa. Inicialmente, é conceituado o problema em questão e as suas restrições. Após, é realizado um estudo acerca dos métodos de solução que serão empregados, cuja implementação e validação serão abordadas nos próximos capítulos.

### 2.1 O problema

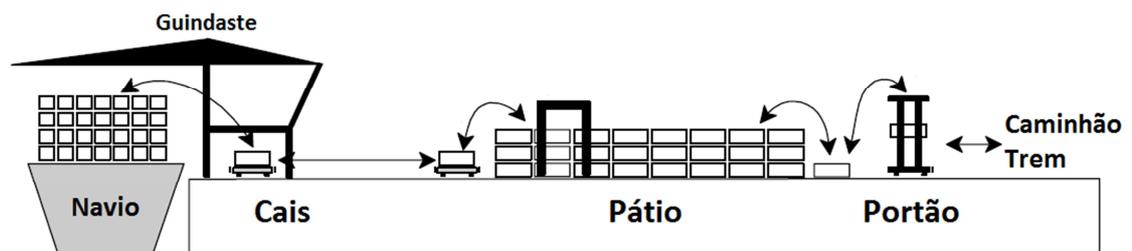
O transporte marítimo é considerado uma das modalidades de transporte mais antigas e de papel importante na história. As descobertas de novas terras e o transporte de mercadorias para países vizinhos são exemplos da importância cultural e comercial deste meio de transporte.

Ao longo dos anos o transporte marítimo se desenvolveu ganhando espaço no mercado. Por meio dele, os mais variados tipos de cargas podem ser transportados. Com os avanços tecnológicos, o transporte de cargas foi padronizado através do uso de contêineres, que facilitaram a realização das operações de carga e descarga de navios, reduzindo o tempo das operações em terminais portuários.

Estes avanços acarretaram mudanças nos portos. Os portos são terminais nos quais são realizadas as operações de recebimento e envio de contêineres para outros portos. Para realizar o descarregamento e o carregamento de contêineres, é necessária uma infraestrutura organizada com equipamentos especializados. Neste sentido, os portos investiram em equipamentos como guindastes, caminhões e empilhadeiras para manuseio de carga (PEDREIRA, 2006).

A estrutura de um terminal portuário pode ser dividida em partes, conforme Figura 2. Uma dessas partes é o pátio, onde estão alocados os contêineres, sendo acessado por caminhões que levam os contêineres a um local mais próximo do navio-contêiner, o cais. No cais, existem guindastes que transferem esses contêineres para dentro do navio e, da mesma forma, é realizada a descarga. Para facilitar o carregamento, é necessário ter um planejamento de quais contêineres devem ser carregados e a sua ordem de carregamento no navio, denominado plano de carga. Na elaboração do plano de carga, deve-se procurar alocar cargas para portos mais distantes mais abaixo do que as que serão descarregadas em portos mais próximos.

Figura 2 - Terminal Portuário



Fonte: Adaptado de (CHOO, 2006)

Com o incremento do uso de contêineres no transporte marítimo de cargas, surgiram problemas quanto à eficiência dos portos, e estes estavam diretamente associados ao plano de carga dos navios-contêineres. Um plano de carga adequado pode refletir na redução de tempo para carregamento e em consequência no custo de operação (PEDREIRA, 2006).

Desta forma, o problema de carregamento de navios surge e se torna relevante quando relacionado à eficiência de um terminal portuário. Por esse motivo, é importante o estudo acerca do problema, em busca da determinação do melhor plano de carregamento, também denominado plano de estiva (*stowage plan*).

O carregamento de navios contêineres é um problema que carece de solução diariamente. O problema consiste em determinar a melhor forma de carregar contêineres, minimizando o tempo de carga e descarga de contêineres, considerando as restrições de operação que o envolvem (AZEVEDO, 2010).

Diferentes restrições podem ser consideradas quando da elaboração de um plano de carga. As seguintes restrições podem ser consideradas (SCIOMACHEN, 2007):

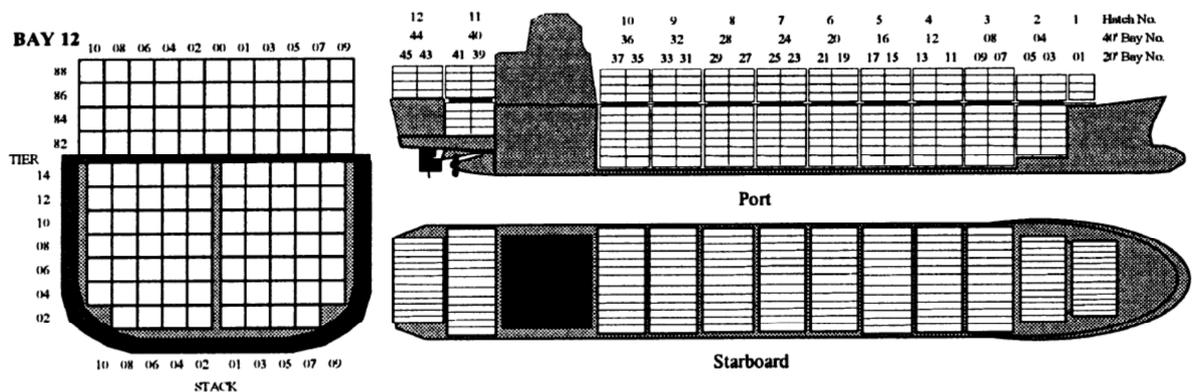
- Minimização das movimentações de contêineres;
- Estrutura do navio: tipo, tamanho e peso dos contêineres;
- Restrições de estabilidade: ao carregar contêineres é desejável considerar o peso do contêiner posicionado abaixo na mesma pilha, pois é preferível que este possua um peso maior do que aquele que se deseja sobrepor. Da mesma forma, o peso das cargas deve ser distribuído de forma a manter o equilíbrio do navio, a fim de evitar esforços estruturais e garantir condições de estabilidade;
- Restrições relativas à localização de contêineres, dependendo do tipo de carga transportada, como cargas perigosas ou refrigeradas;
- Uso da água de lastro para controle de estabilidade e esforços aos quais o navio é submetido;
- Uso de escotilhas, para possibilitar o acesso a diferentes compartimentos do navio.

O emprego da primeira restrição é muito discutido na literatura, sendo conhecida também como remanejamento ou movimentações de contêineres. Esta restrição estará em foco no presente trabalho, uma vez que é a principal restrição adotada no controle do tempo e do custo de operação de um navio no terminal. A redução de movimentações mostra-se necessária nas operações portuárias atuais, pois a cada movimentação de contêiner existe um custo financeiro real relacionado, aproximadamente no valor de US\$ 200,00 (DUBROVSKY, 2002). Se considerada a capacidade de um navio, que varia entre 8.000 a 10.000 contêineres de vinte pés (TEU), e que um navio contêiner pode percorrer uma rota de até 25 portos, percebe-se a importância de um planejamento adequado para evitar movimentações desnecessárias no percurso (DUBROVSKY, 2002) e (CHIWIACOWSKY, 2012).

Neste cenário, o conhecimento da estrutura celular do navio pode auxiliar na elaboração de planos de carga. Para armazenar contêineres, os navios possuem células que são agrupadas, sendo também conhecidas como seções ou baias. Cada agrupamento de células (baia) pode

alocar em torno de 40 contêineres de vinte pés. As baias são organizadas em linhas e colunas numeradas, enquanto os contêineres são empilhados em colunas verticais nestas baias. A Figura 3 apresenta a estrutura celular de um navio.

Figura 3 - Estrutura celular de um navio-contêiner



Fonte: Wilson e Roach (2000)

Quanto ao descarregamento, os contêineres que estão dispostos em pilhas são desembarcados exclusivamente pelo topo. Esta forma de acessar contêineres pode acarretar problemas no percurso. Um exemplo disso é de o navio-contêiner sair do Porto 1 para percorrer uma rota de dez portos, com parada para desembarque de um contêiner em um Porto no meio da rota, que esteja posicionado mais abaixo de uma das pilhas. Para que seja possível realizar esta descarga, é necessário retirar todos os contêineres que estão acima, e após realocá-los novamente para serem descarregados nos seus portos de destino. Este movimento de realocação pode ser minimizado com a programação adequada da operação de carga.

O problema de carregamento de navios-contêineres é um problema de otimização combinatória e de complexidade computacional NP-Completo, conforme apresentado pelos autores Avriel e Penn (1993) e Botter (1992). Para problemas NP (não determinísticos em tempo polinomial) não existem algoritmos rápidos que os resolvam, sendo que os problemas da classe NP-Completo são os mais difíceis da classe NP (SCHRIJVER, 2012).

O problema de carregamento é classificado como um problema de otimização combinatória, e para medir a sua exata dimensão é preciso se ter a informação relativa à capacidade do navio. Admitindo-se que o navio possua capacidade de 2000 TEUs, o número de arrumações possíveis neste caso será de aproximadamente  $3,3 \times 10^{5735}$  (SCIOMACHEN, 2007). Assim, o número de possíveis soluções torna-se elevado, representando um problema de grande escala, e com isso torna-se inviável a determinação de soluções utilizando sistemas manuais (SCIOMACHEN, 2007) e (WILSON, 1999). Os algoritmos propostos para resolução deste tipo de problema são as metaheurísticas, sendo aqui proposto o emprego das metaheurísticas Algoritmo Genético e Enxame de Abelhas, apresentadas na próxima seção.

## 2.2 Métodos de Solução

Como visto anteriormente, o problema de carregamento de contêineres é um problema de otimização combinatória pertencente à classe NP-Completo de problemas e de difícil resolução (SCIOMACHEN, 2007) e (AVRIEL et al., 2000), sendo indicado o emprego de metaheurísticas para a sua solução. Para a solução de alguns problemas de otimização, é possível a aplicação de métodos exatos. Para o problema de carregamento de navios-contêineres, existem trabalhos que o resolvem relacionando-o com o problema de coloração de grafos (AVRIEL, 2000), aplicando o método *Branch and Bound* (WILSON, 1999) e ainda, programação inteira binária (AMBROSINO, 2006), todos aplicáveis a problemas teste de pequeno porte. Entretanto, algumas formulações têm sua solução possível somente através do emprego de métodos aproximativos (heurísticas e metaheurísticas), devido ao tempo computacional exigido por um método exato, ser impeditivo. Para a resolução do problema de carregamento de navios-contêiner de grande porte, é necessário o emprego de uma técnica eficiente e adequada ao problema, sendo recomendável o emprego de metaheurísticas.

A palavra metaheurística foi inicialmente utilizada por Glover (1986), derivada de duas palavras gregas: “heuriskein” originando o termo “heuristic”, que significa encontrar, e “meta” que significa “em um nível superior”. Então, metaheurística exprime a ideia de se encontrar em um nível superior, ou seja, uma busca mais avançada para problemas de nível complexo.

Outras definições de metaheurísticas são encontradas na literatura, como métodos aproximativos e não-determinísticos que guiam o processo de busca (CORDENONSI, 2008). Metaheurísticas são métodos que exploram o espaço de busca de maneira inteligente, desenvolvendo estratégias para a obtenção de soluções promissoras e adaptações para descobrir soluções sempre melhores (CORDENONSI, 2008).

O foco deste trabalho é a utilização de metaheurísticas inspiradas na natureza. A metaheurística Algoritmo Genético se baseia na evolução das espécies, e a metaheurística Enxame de Abelhas analisa o comportamento das abelhas na coleta de alimento.

A metaheurística pode ser caracterizada também por possuir momentos de intensificação e diversificação. A diversificação no algoritmo significa descobrir pontos no espaço de busca que ainda não foram explorados, gerando soluções diversificadas. Por outro lado, a intensificação consiste na exploração local de uma boa solução encontrada em uma região promissora, com o intuito de melhorá-la até convergir ao resultado ótimo. Devido à importância destes conceitos, compreende-se que o ideal seria aplicar a diversificação em conjunto com a intensificação, ou seja, combinar as duas estratégias, de forma adequada, em um mesmo algoritmo. Desta forma, seria obtido um equilíbrio, garantindo que seja alcançada a solução desejada (YANG, 2010).

Algoritmos metaheurísticos podem ser classificados de diferentes formas, como por exemplo algoritmos baseados na busca em trajetória e aqueles baseados em busca populacional (YANG, 2010). Algoritmos baseados em busca populacional caracterizam-se por trabalhar com

um conjunto de indivíduos ou partículas. Exemplos deste tipo de algoritmo são o Algoritmo Genético e o Enxame de Partículas. Algoritmos caracterizados pela busca em trajetória iniciam a exploração pelo espaço de busca a partir de uma solução viável, sendo melhorada ao longo da trajetória através da geração de soluções vizinhas.

Além desta classificação, os algoritmos metaheurísticos podem ser divididos em métodos inspirados na natureza e não inspirados na natureza. Para exemplificar, têm-se algoritmos como Otimização por Colônia de Formigas, Algoritmo Genético e Enxame de Abelhas que são inspirados na natureza por sua relação com comportamentos encontrados na natureza. Dentre os Algoritmos não inspirados na natureza têm-se o Scatter Search e a Busca Tabu que realizam uma busca baseada em um conjunto elite de soluções e uma busca em vizinhança, respectivamente.

Sendo assim, neste trabalho serão adotados dois métodos para serem aplicados na solução do problema de carregamento de navios-contêineres. Os métodos adotados são classificados como Metaheurísticas Populacional e Inspiradas na Natureza (YANG, 2010). Desta forma, a seguir, será apresentado um detalhamento do funcionamento das duas técnicas adotadas: Algoritmo Genético (*Genetic Algorithm*) e Algoritmo Enxame de Abelhas (*Swarm Bees Algorithm*).

### 2.2.1 Algoritmo Genético (AG)

O Algoritmo Genético é inspirado na teoria da evolução natural proposta por Charles Darwin, desenvolvido por John Holland em meados da década de 70. Desde então, diversas variações deste algoritmo foram propostas para solução de problemas de otimização de nível complexo (HÜE, 1997).

O algoritmo é baseado na evolução das espécies, onde indivíduos se reproduzem herdando características de seus ascendentes e adaptando-se ao ambiente natural. Os indivíduos que apresentarem menor nível de adaptação ao meio ambiente morrerão e a sobrevivência se dará aos mais aptos.

O componente do algoritmo que representa os indivíduos é o cromossomo. Na biologia, as características presentes no cromossomo são chamadas de genótipo, sendo constituído por genes, os quais são utilizados para representação dos componentes de uma solução. Cada gene possui diferentes valores, estes são denominados alelos (HÜE, 1997).

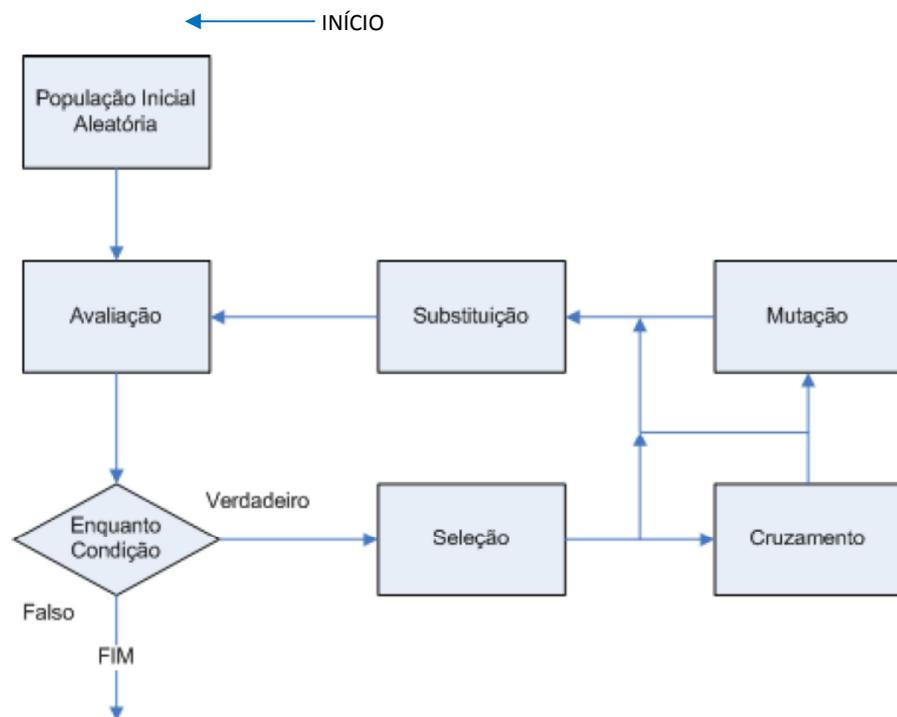
O cromossomo representa as possíveis soluções do problema e para sua representação é necessário definir a codificação genética. A codificação pode ser feita de diversas maneiras conforme o problema. Exemplos disso são codificações através do emprego de números binários, inteiros e/ou reais. Após a codificação, são geradas soluções viáveis de maneira aleatória, compreendidas dentro do espaço de busca, compondo um conjunto inicial de soluções ou indivíduos, a população inicial.

Para explorar as possíveis soluções, é criada uma população de cromossomos, ou soluções, com tamanho dado pela quantidade de cromossomos presentes nela. O tamanho da população pode variar conforme o problema, porém quanto maior a população maior serão as

chances de se encontrar a solução para o problema, entretanto, por outro lado, maior será o tempo de processamento do algoritmo (DIAS, 2006).

Como pode-se observar na Figura 4, o algoritmo inicia com a criação da população inicial. O segundo passo é avaliar cada indivíduo (cromossomo) da população de acordo com a função objetivo. Nesta avaliação, é determinado o valor de aptidão destes indivíduos. Se após feita a avaliação (aptidão), o algoritmo não atingir o critério de parada, é realizada a seleção, através de um operador adequado, onde é escolhido um par de cromossomos pais, sendo estes combinados, através de uma operação de cruzamento, originando novos cromossomos. Observa-se na Figura 4 que, em alguns casos, o cruzamento não é realizado, isto se deve ao fato de a taxa de cruzamento não ser 100%, existindo assim a possibilidade de não ocorrer o cruzamento. Em seguida, realiza-se a mutação destes cromossomos, com uma determinada probabilidade, em geral pequena. A mutação realiza a substituição de valores de um ou mais genes, dependendo do operador de mutação adotado. Após este passo, é realizada a substituição dos indivíduos, gerando uma nova população, e assim ocorre o encerramento do ciclo, que será repetido até que o critério de parada seja satisfeito e se encontre a solução ótima (CHAN, 1998).

**Figura 4 - Fluxograma AG**



Fonte: Adaptado de (FILITTO, 2008).

Há diferentes tipos de abordagens quanto ao critério de parada. Como critério de parada pode ser adotado um número máximo de gerações, ou seja, ao alcançar um determinado número de gerações o algoritmo para. Outro critério que também pode ser empregado é a realização de iterações até que, através de uma avaliação, seja percebido que não houve mais melhora nos indivíduos da população. Ou ainda, pode ser encerrado o algoritmo quando a maioria dos

membros da população se tornem semelhantes, o que indica que o algoritmo perdeu diversidade e, possivelmente, não encontrará soluções melhores (HÜE, 1997)

### 2.2.1.1 Função de Aptidão

A função objetivo é a função que deve ser otimizada, sendo também denominada função de aptidão. Desta forma, no momento de avaliação dos indivíduos da população, esta função determina o nível de aptidão de cada indivíduo.

Se a função objetivo for de minimização, quanto menor o valor da função objetivo do indivíduo avaliado, mais apto para ser mantido na população. Logo, o indivíduo avaliado com o menor valor de aptidão será a melhor solução. Após a avaliação da aptidão, é realizado um ranqueamento dos indivíduos presentes na população.

### 2.2.1.2 Seleção

A seleção tem o papel de escolher os indivíduos mais aptos para a reprodução, na esperança de que sua descendência seja ainda melhor. Os indivíduos são selecionados aleatoriamente utilizando uma probabilidade, a qual se baseia no nível de aptidão dos indivíduos. Porém, uma seleção adequada não deve levar em conta somente elementos altamente aptos, pois extingiria a diversidade de busca que será importante ao longo do processo evolutivo do AG (MITCHELL, 1996).

Existem diferentes operadores de seleção, neste trabalho serão apresentados os mais usuais. Conforme os autores Goldberg (1999) e Mitchell (1996), destacam-se os seguintes operadores:

- Seleção proporcional ou roleta: atribui uma fatia da roleta para cada indivíduo conforme sua aptidão. Depois de obtido o valor da função de aptidão de cada indivíduo obtém-se a aptidão total de todos os indivíduos do grupo conforme a equação (1):

$$f_T = \sum_{i=1}^{N_i} f_i , \quad (1)$$

onde  $N_i$  é o número de indivíduos na população e  $f_i$  é a função de aptidão do indivíduo  $i$ .

A probabilidade de seleção dos indivíduos, individualmente, é encontrada conforme a equação (2):

$$p_i = \frac{f_i}{f_T} . \quad (2)$$

Então, a roleta é girada  $N_i$  vezes, e aleatoriamente é escolhido um valor  $r$  entre 0 e 1. O valor que estiver acima de  $r$  é escolhido.

- Seleção por ranqueamento: usada quando existe disparidade entre os valores de aptidão dos indivíduos. Os indivíduos de uma população são então ranqueados de acordo com sua aptidão, ou seja, é realizada uma classificação.
- Seleção por torneio: determina-se uma probabilidade de aceitação no início do algoritmo. Após, dois indivíduos da população são escolhidos aleatoriamente. Então, de forma aleatória, é selecionado um número entre 0 e 1. Se este número for menor que o valor da probabilidade definida inicialmente, o indivíduo de maior aptidão é selecionado, caso contrário o de menor aptidão é selecionado. Com este operador de seleção, as chances de indivíduos mais aptos serem selecionados são altas, mas em alguns casos há chances de escolha também para os menos aptos.

### 2.2.1.3 Reprodução

O operador de cruzamento é necessário para a exploração de novas regiões do espaço de busca. O cruzamento (reprodução) consiste na troca de genes entre um par de pais selecionados, originando novos indivíduos. A probabilidade de cruzamento é definida no início do algoritmo e depois avaliada através de um valor gerado aleatoriamente. Caso ocorrer a opção de não cruzamento, poderão ser copiados os pais tais como são (CHAN,1998).

Há diferentes tipos de operadores de cruzamento que variam conforme a codificação escolhida. Exemplos de operadores que podem ser utilizados para a codificação binária são: cruzamento de um ponto, cruzamento de múltiplos pontos e cruzamento uniforme. Se for empregada codificação real, existem o cruzamento simples, cruzamento aritmético e cruzamento geométrico, conforme em (MALAQUIAS, 2006), (GOLDBERG,1999):

- No cruzamento de um ponto, define-se um ponto de corte e são trocados genes de um pai para outro, conforme o exemplo a seguir nas equações (3) e (4) com os pais 1 e 2:

$$P_1 = p_1^1/p_1^2/p_1^3 \dots/p_1^n , \quad (3)$$

$$P_2 = p_2^1/p_2^2/p_2^3 \dots/p_2^n . \quad (4)$$

Assumindo o corte realizado entre os genes 2 e 3, os cromossomos resultantes são conforme as equações (5) e (6) :

$$C_1 = p_1^1/p_1^2/p_2^3 \dots/p_2^n , \quad (5)$$

$$C_2 = p_2^1/p_2^2/p_1^3 \dots/p_1^n . \quad (6)$$

- O cruzamento de múltiplos pontos é semelhante ao cruzamento de um ponto, mas este possui dois ou mais pontos de corte.

- No cruzamento uniforme, para cada gene é sorteado um dos pais para que contribua com a sua informação na mesma posição.
- O Cruzamento simples usa uma ideia semelhante ao cruzamento de um ponto, mas na forma de codificação real.
- O Cruzamento aritmético utiliza uma combinação linear para encontrar dois cromossomos filhos  $C_1$  e  $C_2$ , a partir dos cromossomos pais  $P_1$  e  $P_2$ , conforme as equações (7) e (8):

$$C_1 = \beta P_1 + (1 - \beta)P_2, \quad (7)$$

$$C_2 = (1 - \beta)P_1 + \beta P_2, \quad (8)$$

onde  $\beta \in U(0,1)$ .

- O Cruzamento geométrico gera apenas um filho a partir de dois cromossomos pais, conforme a equação (9):

$$C = P_1^a \cdot P_2^{1-a}, \quad (9)$$

onde  $a \in [0,1]$ .

#### 2.2.1.4 Mutação

A mutação consiste na alteração de um ou mais pontos (genes), escolhidos aleatoriamente. O seu papel é diversificar a população e acrescentar pontos que não haviam sido explorados até o momento (DIAS, 2006).

Geralmente a taxa de mutação é reduzida, diferentemente do cruzamento, uma vez que a mutação tem como objetivo diversificar a população, e se empregada uma taxa elevada, a mutação estará diversificando exageradamente a população, e em consequência poderão ser perdidas soluções que já estivessem muito próximas do ótimo global.

Há diferentes tipos de operadores de mutação dependendo da codificação adotada, por exemplo, se a codificação da solução emprega números binários, basta fazer a troca de um bit por outro. Outros exemplos são a mutação uniforme, mutação não-uniforme e mutação de contorno (CATARINA, 2006), (HÜE, 1997) e (MICHALEWICZ, 1996):

- Mutação Uniforme: consiste na substituição de um gene, cuja posição e valor são escolhidos de forma aleatória.
- Mutação Não-Uniforme: um indivíduo é definido como  $x^t = [x_1 \dots x_n]$ , sendo que um componente  $x_k$  é escolhido aleatoriamente e o indivíduo após a mutação é representado na equação (10):

$$x^{t+1} = [x_1 \dots x'_k \dots x_n], \quad (10)$$

obedecendo a regra conforme as equações (11) e (12):

$$x'_k = \begin{cases} x_k + \Delta(t, a - x_k) & \text{com 50\% de probabilidade} \\ x_k - \Delta(t, x_k - b) & \text{com 50\% de probabilidade} \end{cases} \quad (11)$$

$$\Delta(t, y) = y(1 - r^{(1-\frac{t}{T})^P}), \quad (12)$$

onde  $a$  e  $b$  são os limites inferior e superior de  $x_k$ . A função  $\Delta(t, y)$  resulta em um valor que está em  $[0, y]$ , ou seja, quanto menor for o valor de  $t$  mais o operador fará buscas globais, e a medida que  $t$  aumenta as buscas se tornam mais locais.

- **Mutação de Contorno:** seleciona uma variável  $k$  aleatoriamente, com distribuição uniforme, e após é dado um limite máximo ou mínimo ao parâmetro, conforme a equação (13):

$$x'_k = \begin{cases} l_{\text{sup}}(k), & \text{se } rand = 0 \\ l_{\text{inf}}(k), & \text{se } rand = 1 \end{cases} \quad (13)$$

### 2.2.1.5 Substituição

Novos indivíduos são substituídos na população formando uma nova população e eliminando indivíduos antigos. Este processo se dá no último passo de cada ciclo do algoritmo.

Após obter os filhos de cada geração, é necessário um método que defina quais membros devem ser substituídos por novos. Para isso existem dois métodos, o de atualização geracional e atualização não-geracional.

A atualização geracional consiste em gerar a mesma quantidade de filhos que o tamanho da população, com o intuito de os novos filhos substituírem completamente a população inicial. Porém, neste método, o indivíduo só é combinado com indivíduos da mesma geração.

Na atualização não-geracional, os indivíduos novos são inseridos na população tão logo sejam gerados, levando à substituição de um membro da população. O indivíduo a ser excluído pode ser ou o pior membro ou, além disso, também o mais antigo na população. Outra forma de substituição, quando empregada a atualização não-geracional, baseia-se no torneio. Este torneio pode ser realizado entre os piores membros da população, ou ainda entre os mais semelhantes. Desta forma, permanecem na população os membros mais aptos e mais diversificados.

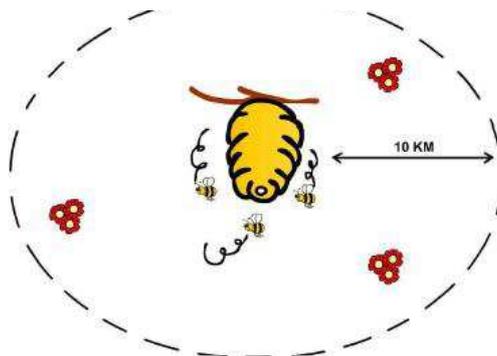
Outro método bem eficiente é o elitismo. O elitismo é um procedimento que força reter um determinado número de melhores indivíduos da população. A cada geração, todos os indivíduos da população são substituídos com exceção dos indivíduos da elite, que são mantidos e replicados na próxima geração. Esta técnica é utilizada para melhorar a convergência do algoritmo, o qual prioriza sempre as melhores soluções, não permitindo que soluções melhores sejam substituídas por piores (MITCHELL, 1996).

### 2.2.2 Enxame de Abelhas (*Swarm Bees - SB*)

Nos últimos anos, foram desenvolvidos diversos algoritmos baseados no comportamento das abelhas, levando em conta diversas características deste comportamento, tendo sido criados algoritmos que cuidavam do comportamento da coleta de alimentos, do comportamento de acasalamento e do conceito de abelha-rainha. O motivo de tamanha exploração da ideia de funcionamento dos enxames de abelhas se deve à alta capacidade organizacional que possuem, sendo um dos principais insetos a serem estudados (SERAPIÃO, 2009).

Neste estudo, serão desenvolvidas aplicações do algoritmo com foco no comportamento das abelhas na coleta de alimentos. Para a coleta de alimento, uma abelha pode percorrer até 10 km de distância da colmeia, em múltiplas direções, conforme representado na Figura 5. Onde houver uma quantidade maior de alimento (néctar), mais abelhas se deslocam para este local, e de forma análoga, nos locais com menor quantidade de comida se deslocam poucas abelhas (PHAM, 2006).

**Figura 5 - Coleta de Alimento**



Fonte: SIKORA (2010)

No período do verão, uma colônia de abelhas armazena comida suficiente para o inverno, pois neste período as fontes de alimento diminuem. As abelhas devem adaptar-se à quantidade de fontes localizadas ao redor de sua colmeia, que podem se alterar dependendo da distância ou da época do ano. Esta capacidade de adaptação é importante para o bom aproveitamento das fontes de alimento e manutenção da colônia de abelhas (SIKORA, 2010).

Para a coleta, as abelhas são extremamente organizadas, elas se comunicam por sons, substância química, tato e danças (estímulos). As abelhas que iniciam a exploração por alimento são chamadas de batadoras ou operárias, e são espalhadas pelos campos, em uma busca exploratória por comida em todas as direções. Ao retornarem à colmeia, depositam o pólen e vão para a chamada “pista de dança” da colmeia e realizam a *waggle dance*, comunicando às outras abelhas informações como direção, distância e qualidade do alimento. No meio da dança, a abelha pode interromper para dar uma gota do néctar colhido para as abelhas seguidoras, na tentativa de convencê-las de segui-la na colheita. Após a dança, as abelhas seguidoras seguem

para o campo que é mais vantajoso para realizarem a colheita, a partir da qual o processo se repete (YANG, 2010).

#### 2.2.2.1 Algoritmo Colônia Artificial de Abelhas (ABC)

Os algoritmos encontrados na literatura, baseados nas abelhas, são criados em analogia ao seu comportamento natural. Alguns focalizam o comportamento de acasalamento, outros a coleta de alimento e também o conceito de abelha-rainha. O algoritmo empregado neste trabalho é o *Artificial Bee Colony* (ABC) que é baseado na coleta de alimento e apropriado para problemas de otimização. O algoritmo ABC foi proposto pela primeira vez por Karaboga em 2005.

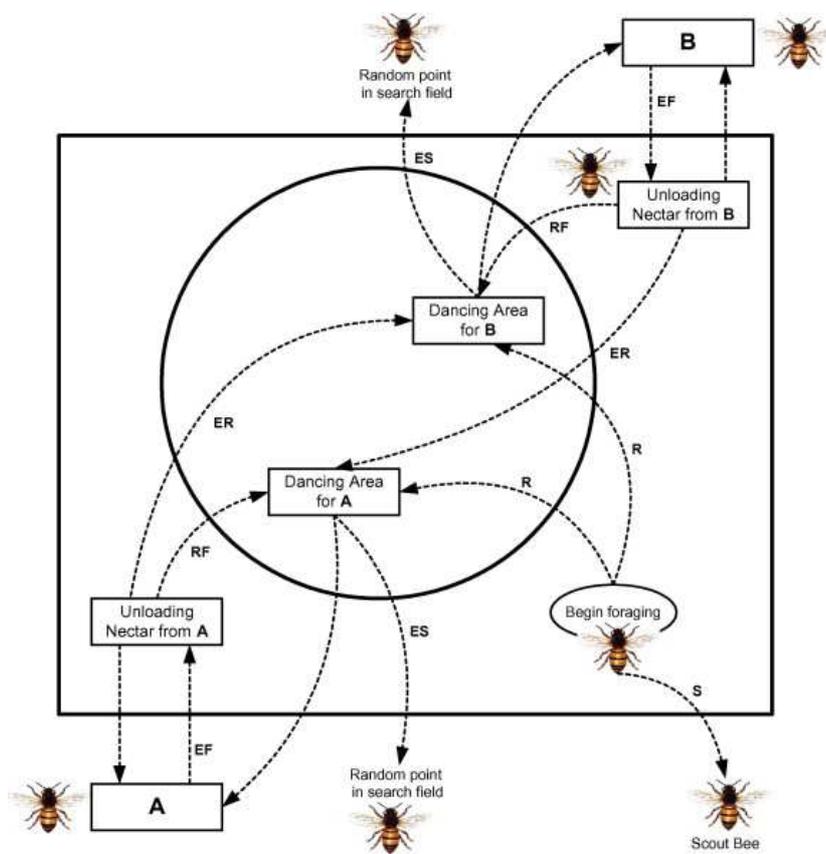
No algoritmo ABC, as abelhas de uma colônia são divididas em abelhas operárias (campeiras), abelhas seguidoras (observadoras) e abelhas escudeiras. As abelhas operárias são designadas para realizarem a procura por fontes de alimento, as abelhas seguidoras escolhem as fontes que irão seguir e as abelhas escudeiras procuram fontes de alimento aleatoriamente.

Inicialmente, as abelhas operárias saem da colmeia em busca de uma fonte de alimento particular. Ao retornarem, compartilham informações, com certa probabilidade, com as abelhas seguidoras. Para a escolha de uma fonte por parte das abelhas seguidoras, a quantidade de néctar de cada fonte é avaliada, ou seja, quanto maior a quantidade mais chances das seguidoras escolherem tal fonte. Esta exploração contínua das fontes de alimento leva ao seu esgotamento. Quando a fonte da abelha operária for esgotada, a abelha operária torna-se uma abelha escudeira e volta a explorar o ambiente em busca de novas fontes de alimento de maneira aleatória, conforme representado na Figura 6.

Uma consideração importante para o desenvolvimento do algoritmo é que no início, na colmeia, 50% das abelhas são operárias e a outra metade são abelhas seguidoras. Para toda abelha operária existe uma fonte de alimento, ou seja, o número de abelhas operárias é o mesmo que o de fontes, uma vez que cada abelha é associada a apenas uma fonte de alimento.

Ainda, no algoritmo, a posição da fonte é considerada uma solução possível do problema e a quantidade de néctar presente em cada fonte de alimento é o termo que avalia a aptidão ou a qualidade da fonte (SERAPIÃO, 2009).

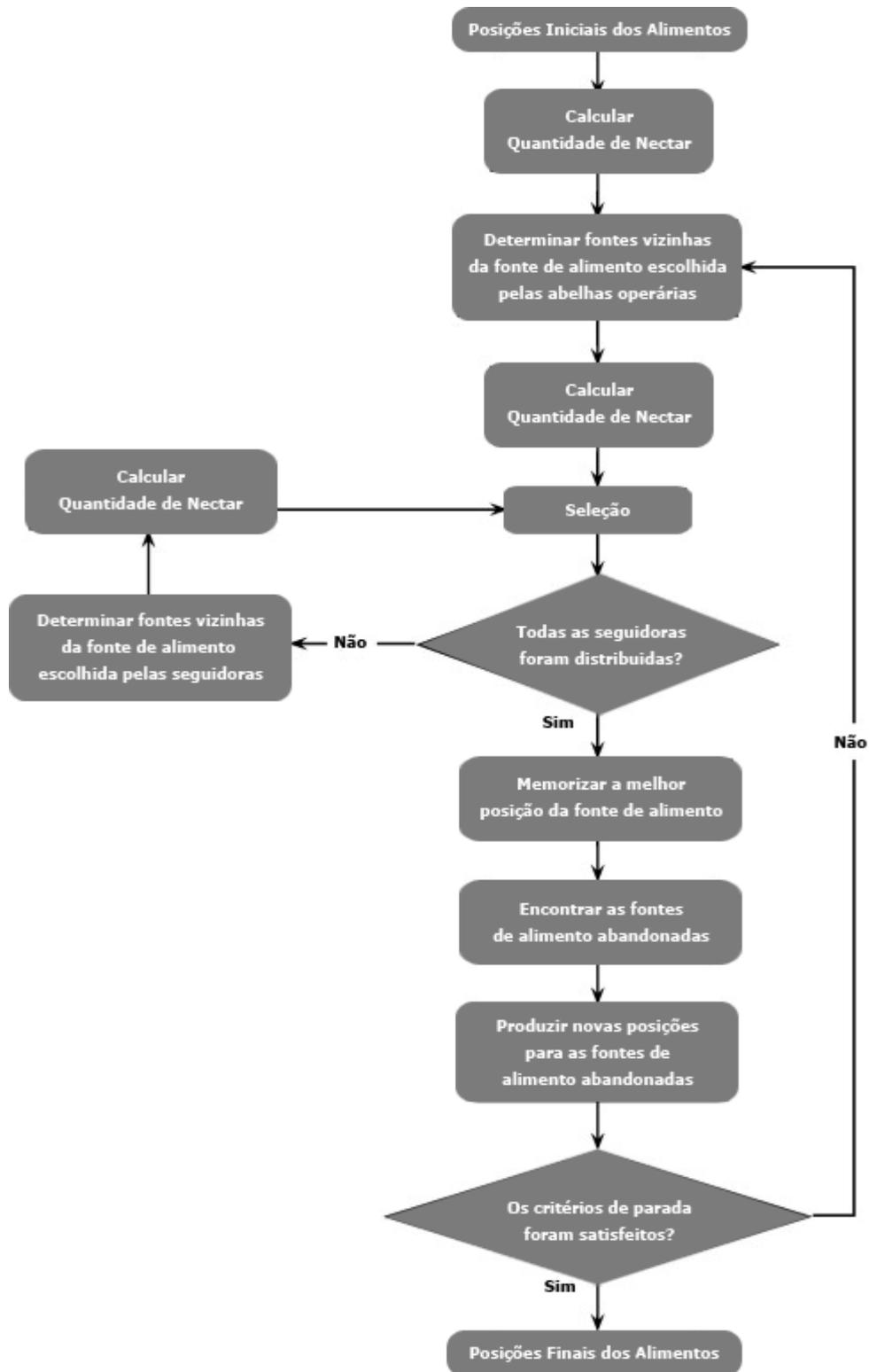
**Figura 6 - Abelhas no processo de coleta de alimento**



Fonte: Özbakir (2009)

O Algoritmo ABC, passo a passo, é apresentado a seguir (KARABOGA; SIKORA; SZETO, 2010, 2010, 2010) e pode ser acompanhado, através do diagrama da Figura 7:

Figura 7 - Fluxograma ABC



- **Passo 1**- Posições Iniciais: Iniciar gerando as posições das fontes de alimento aleatoriamente (fase da abelha operária). Para a inicialização do algoritmo pode ser considerada a definição conforme a equação (14):

$$X_{mi} = l_i + rand(0,1) * (u_i - l_i) \quad (14)$$

onde  $l_i$  e  $u_i$  são os limites inferior e superior do parâmetro  $X_{mi}$ , respectivamente;

- **Passo 2** – Cálculo do néctar: Calcula-se a aptidão de cada fonte de alimento (quantidade de néctar). Pode ser calculada pela fórmula da equação (15):

$$fit_m(x_m) = \begin{cases} \frac{1}{1+f_m(x_m)} & se\ f_m(x_m) \geq 0 \\ 1 + abs(f_m(x_m)) & se\ f_m(x_m) < 0 \end{cases} \quad (15)$$

onde  $f_m(x_m)$  é a função objetivo e  $x_m$  é a solução avaliada;

- **Passo 3** - Vizinhança: Determina fontes vizinhas;
- **Passo 4** – Cálculo do néctar: Mesmo cálculo realizado no Passo 2;
- **Passo 5** - Seleção: Cada abelha seguidora seleciona uma fonte de alimento de acordo com a sua qualidade e segue explorando-a. A probabilidade de escolha de uma determinada fonte, pela abelha seguidora, pode ser definida conforme a equação (16):

$$P_i = \frac{F(x_i)}{\sum_{j=1}^S F(x_j)}, \quad (16)$$

onde  $x_i$  corresponde ao local da fonte e  $F$  a quantidade de néctar. O símbolo  $S$  representa a quantidade de fontes de alimento. A seleção é baseada no método de seleção roleta, que avalia a probabilidade de escolhas. Após a abelha seguidora ter selecionado uma fonte de alimento, cada abelha deve encontrar uma fonte vizinha da fonte selecionada. Para isso, aplica-se um operador de vizinhança adequado para a codificação no próximo passo.

- **Passo 6** - Fontes distribuídas: Efetua-se um ciclo onde são distribuídas as abelhas seguidoras e geradas as novas soluções vizinhas, até que seja distribuída uma fonte para cada abelha. Quando todas as abelhas seguidoras forem distribuídas, o algoritmo segue para o próximo passo.

A expressão da exploração da abelha escudeira por novas fontes é dada pela equação (17):

$$v_{ij} = x_{ij} + \phi_{ij} \times (x_{ij} - x_{kj}), \quad (17)$$

onde a posição das novas fontes é  $v_{ij}$ , e  $k$  pertence ao intervalo de 1 até o número inicial de abelhas operárias. O símbolo  $\phi_{ij}$  é um número aleatório entre  $[-1,1]$ ,  $k \neq i$ .

- **Passo 7** - Memorização: Memorizar a melhor fonte encontrada até o momento;
- **Passo 8** - Encontrar fontes abandonadas: Determinar a fonte abandonada e destinar uma abelha operária, que torna-se escudeira, para procurar uma nova fonte de alimento;
- **Passo 9** - Produzir novas posições: Após encontrar as fontes abandonadas, as novas posições de fontes são produzidas aleatoriamente, de forma a criar fontes vizinhas às fontes abandonadas. Estas novas posições tem o intuito de diversificar a população.
- **Passo 10** - Parada: Se o critério de parada não for satisfeito, retornar ao passo 2. Ao atingir o critério de parada, apresentar a solução.

### 3 REVISÃO BIBLIOGRÁFICA

Neste capítulo é apresentada uma discussão de trabalhos que abordam os diferentes problemas que surgem em um terminal portuário, os quais estão diretamente relacionados ao problema de carregamento de navios-contêineres.

Os trabalhos aqui discutidos aplicam os métodos adotados na presente pesquisa. Com relação ao método do Algoritmo Genético, a sua aplicação a problemas relacionados ao tema de pesquisa é encontrada em vários artigos, porém em relação ao Método de Enxame de Abelhas (SB) não foram encontrados na literatura trabalhos aplicados a problemas relativos à atividade de terminais portuários. Desta forma, serão discutidos trabalhos que utilizam o AG, bem como outras técnicas, quando empregadas na solução do problema em discussão. Já em relação à técnica SB, serão discutidos trabalhos que apresentam aplicações a outros temas de pesquisa.

#### 3.1 Problemas relacionados a operações em terminais portuários

O problema de carregamento vem sendo estudado desde 1970 por W.C. Webster e P. Van Dyke, da Hydronautics Inc (KAISAR, 2006), com foco no carregamento e descarregamento de contêineres, e levando em conta a estabilidade do navio. Somente em 1981 foi realizado um grande estudo de um pré-planejamento feito por computador, usando técnicas de simulação.

Diferentes problemas relacionados ao problema de carregamento de navios-contêineres vêm sendo abordados na literatura, entre eles destaca-se o Problema de Alocação de Berços (PAB). Este problema trata da alocação de navios a posições do cais de atracação em terminais portuários, com o objetivo de maximizar o uso do espaço e minimizar o tempo de serviço e permanência do navio no porto (VACCA, 2007). Para resolução deste problema, Lorena e colaboradores (2010) utilizaram métodos como *Clustering Search* e *Simulated Annealing*. Kim e Moon (2003) também utilizaram o algoritmo *Simulated Annealing* para resolver o problema modelado por programação inteira mista, que determinou tempos e posições de navios para a atracação em terminais de contêineres. Chen e Hsieh (1999) consideraram o horário de atracação e desatracação de cada navio utilizando programação inteira e o algoritmo *Branch and Bound* para resolver o mesmo problema. Nishimura, Imai e Papadimitriou (2006) resolveram o mesmo problema por meio de uma heurística baseada no Algoritmo Genético, que gerou boas soluções.

Outros problemas relacionados a terminais portuários estão ligados às operações nos terminais, envolvendo a programação das tarefas de guindastes e ainda a minimização do tempo de conclusão de operações de carga ou descarga. Kim e Park (2004) abordam o problema para minimizar o tempo das operações e encontrar a melhor sequência de carregamentos através de guindastes, utilizando o *Branch and Bound* e a metaheurística GRASP. Sammarra et al. (2007), decompõem o problema em dois, um problema de roteamento e um de escalonamento. O problema de roteamento é resolvido pelo algoritmo Busca Tabu, e para o problema de escalonamento é utilizada uma técnica de busca local. Moccia et al. (2006) propõem uma nova

formulação para o problema e o resolvem através do método *Branch and Cut*, superando os resultados encontrados pelo método *Branch and Bound*.

A gestão de operações no pátio é outro problema relacionado que vem sendo abordado recentemente na literatura, onde busca-se a determinação de políticas a serem adotadas no armazenamento de contêineres em pilhas de acordo com o seu peso, tamanho e destino. Ng e Mak (2005) propõem uma solução exata para o escalonamento de trabalhos dos guindastes no pátio a fim de minimizar o tempo total de trabalho através do algoritmo *Branch and Bound*. Lee et al. (2006) resolvem o problema de alocação de cargas no pátio adotando um método sequencial e um método de geração de colunas com o intuito de minimizar o número de guindastes no pátio.

Ainda, trabalhos sobre as operações de transferência de contêineres do cais para o pátio, com o objetivo de minimizar a frota de veículos utilizados para o processo, podem ser encontrados. Vis et al. (2005) utilizam áreas neutras para transferência de contêineres, dividindo a operação em dois processos: descarga e transporte. Através do método de programação inteira, busca-se a minimização do tamanho da frota de maneira que cada contêiner seja carregado dentro de seu tempo. Lee et al. (2007) apresentam um modelo para resolver o escalonamento do sistema de dois *transtainer* (equipamento utilizado para arrumação de contêineres no pátio), de forma a minimizar o tempo de carregamento total através do método *Simulated Annealing*.

### **3.2 O problema de carregamento de contêineres resolvido por AG**

Dubrovsky (2002) resolve o problema de carregamento de contêineres (PCC) com o objetivo de minimizar o número de movimentos de contêineres através do AG, usando uma codificação compacta para resolução do problema. A codificação compacta utilizada se mostrou eficiente, pois reduziu o espaço de busca e em consequência reduziu também o custo computacional, permitindo uma convergência mais rápida e uma avaliação simplificada da qualidade da solução. O Algoritmo Genético aplicado contou com operadores de seleção, cruzamento e epidemia, descartando o operador de mutação. É também proposta uma implementação paralela do AG, sendo que a cada iteração é feita uma combinação de dois pais com descendência de dois filhos, que são avaliados em seguida. Esta avaliação foi realizada com múltiplos processadores, acelerando o processo.

Em (AZEVEDO, 2010), foi empregado também o método do Algoritmo Genético para resolver o PCC. Este estudo tem como objetivo minimizar os remanejamentos de contêineres e neste sentido foi empregado um sistema de regras de carregamento e descarregamento de contêineres para representação da solução. A codificação dos indivíduos do AG se dá na forma das regras de carga e descarga de contêineres, consequentemente a solução obtida indica as regras adequadas a serem utilizadas no plano de carga.

Em (CHIWIACOWSKY, 2011), também foi aplicado o AG ao PCC, através de uma codificação compacta baseada no trabalho de Dubrovsky (2002). A codificação compacta possibilitou a redução do espaço de busca e em consequência uma busca possível por soluções

promissoras. Esta codificação consiste na divisão do problema em seções, destinando quatro seções para cada porto, distribuindo contêineres entre as seções. O Algoritmo Genético foi aplicado utilizando três configurações diferentes, variando os operadores genéticos utilizados. A melhor configuração do AG foi obtida com os operadores de seleção, cruzamento, mutação e epidemia. Porém, para um tamanho maior da rota, esta configuração, denominada completa, necessitou um maior número de gerações para fornecer boas soluções.

### 3.3 Outras técnicas aplicadas ao problema de carregamento de contêineres

Ao longo da pesquisa, foram identificadas outras técnicas para resolução do problema de carregamento de navios-contêineres. Tais técnicas possuem características interessantes de serem estudadas e servem para conhecer melhor o que já foi trabalhado e quais técnicas já foram utilizadas para a resolução do problema em estudo. Ainda, pode-se identificar as abordagens mais adequadas no tratamento do problema abordado.

O problema da coloração de grafos circulares se assemelha muito ao problema de carregamento de navios-contêiner, ambos modelados como um problema de minimização. Na coloração de grafos, os seus vértices devem ser coloridos de tal forma que não haja dois vértices adjacentes que compartilhem a mesma cor, o objetivo é encontrar o menor número de cores para colorir tal grafo. Em (AVRIEL, 2000), é mostrada a relação entre os dois problemas utilizando matrizes de transporte. Para encontrar a solução, basta localizar uma coluna que não possua nenhuma movimentação em um plano de carga de um navio, o que seria equivalente a encontrar o número cromático da coloração de grafos.

Outro método encontrado na literatura foi o da heurística suspensória (AVRIEL, 1998). Neste método, o planejamento de carga de um navio-contêiner é realizado através do emprego de uma matriz de transporte que contém as quantidades de contêineres e os respectivos destinos. Logo, são criadas outras  $N - 1$  matrizes de dimensão  $R \times C$ , onde  $N$  representa o número de portos,  $R$  as linhas e  $C$  as colunas que compõem as seções do navio, para o planejamento de carregamento dos portos  $i = 1, \dots, N-1$ . O plano de carga é feito para cada porto, onde as matrizes são configuradas em três etapas, *hanging*, *filling* e *dropping*. O primeiro passo é o *hanging*, que efetua a atribuição de contêineres às linhas de uma coluna na ordem decrescente. Quando da aplicação do *filling*, ainda existem contêineres que não foram atribuídos à matriz e nesta etapa são colocados no local onde possuir mais colunas vazias. O *dropping* é uma movimentação para baixo, ou seja, contêineres que estão suspensos em uma determinada coluna serão abaixados o quanto for possível. Desta forma, a matriz vai se adequando a cada porto à medida que são realizados os passos descritos acima, considerando os descarregamentos feitos no percurso (AVRIEL, 1998).

O trabalho de Gümüs e colaboradores (2008) aplica uma heurística de decomposição multi-estágio para resolução do PCC. A decomposição é feita em quatro estágios, no primeiro é focada a questão da eficiência do guindaste nos portos e a minimização do número de remanejamentos. No segundo estágio, é considerado o impacto da localização das escotilhas na

arrumação. No terceiro estágio, uma atribuição de *slots* é realizada baseada na saída do segundo estágio. No último estágio, o plano de estiva é gerado.

Vários estudos podem ser encontrados na literatura aplicando o método para resolução do *3D-Bin Packing Problem* (3D-BPP) com o objetivo de reduzir o tempo de carregamento de navios-contêiner. Neste método, são consideradas baías de três dimensões e ainda contêineres de diferentes tamanhos, o que leva a dificuldades de localização e violação das regras de tamanho dos contêineres dos navios. O carregamento é iniciado no canto inferior do lado esquerdo, continuando o preenchimento de forma vertical, porém a forma ideal seria o preenchimento horizontal, pois leva em conta a estabilidade do navio. Contêineres mais pesados e de maior volume tem prioridade no carregamento, o que pode violar a lei dos navios, pois as baías possuem tamanhos padronizados e cada espécie de contêiner possui uma baía adequada ao seu tamanho. O problema de carregamento de navios-contêineres pode ser baseado no problema do *Bin-Packing* tridimensional (SCHIOMACHEN, 2004)

Os autores Wilson e Roach (1999) empregaram o método *Branch and Bound* e a metaheurística Busca Tabu compondo um método híbrido para solucionar o PCC. O método *Branch and Bound* é aplicado para uma representação generalizada, de modo que todos os contêineres sejam alocados em blocos (não *slots*). Segundo os autores, os blocos localizados acima do convés de carga podem ser colocados entre duas seções da escotilha, criando blocos de *slots* de carga relacionados a seções das escotilhas. Um dos objetivos deste método é minimizar o número de blocos de carga ocupados por contêineres, para cada destino, e ainda minimizar a abertura das escotilhas. Já o objetivo da Busca Tabu, foi de alocar os contêineres em *slots* de forma a minimizar os movimentos. Antes da representação generalizada em um determinado bloco, cada contêiner é alocado heurísticamente para um *slot*. A aplicação de uma heurística sensível facilita o processo de otimização na configuração de arrumação de contêineres.

A heurística *Beam Search* foi empregada por Azevedo (2006) e consiste na adaptação do método *Branch and Bound*. A heurística se baseia na técnica de decisão em árvore, onde é analisado um número determinado de nós, e cada nó equivale a uma solução. A árvore contém  $N-1$  níveis, onde  $N$  é o número de portos. A cada nível são feitas atribuições de regras de carregamento e descarregamento, sendo que no último porto é realizado somente descarregamento, e por este motivo não são necessárias regras, sendo o planejamento feito apenas para  $N-1$  portos. O *Beam Search* define a sequência de regras de carregamento e descarregamento que fornece o menor número de remanejamentos de contêineres nos  $N$  portos, sendo também considerada em sua formulação a restrição de estabilidade.

No artigo de Ambrosino (2006), é empregado um modelo de programação inteira binária que consiste em minimizar o tempo total de carga de navios-contêiner, levando em conta restrições de peso, tamanho e estabilidade. Para a resolução do PCC são seguidos três passos. Primeiramente, é denotada uma árvore de ramificações, a qual divide o navio em porções diferentes, onde contêineres são associados a diferentes subconjuntos de baías. No passo seguinte, é encontrada a maneira mais adequada para carregar contêineres em cada partição do navio, através da resolução do problema de Programação Linear. Deste modo, cada partição do navio limita a busca por contêineres, o que reduz o espaço de soluções. O último passo consiste na exclusão de possíveis inviabilidades da solução global.

### 3.4 A Técnica do Enxame de Abelhas aplicada a outros problemas

A técnica de Enxame de Abelhas foi pouco explorada, pois esta técnica é recente. Apesar de não terem sido encontrados trabalhos que resolvam o problema de carregamento de navios-contêineres através do Enxame de Abelhas, outros trabalhos foram encontrados na literatura aplicando este método.

Szeto (2011) resolve o problema de roteamento de veículos via algoritmo da colônia artificial de abelhas (ABC). Em seu trabalho, é proposta uma versão melhorada do algoritmo com o intuito de aprimorar a solução se comparada com a versão original. Utilizando instâncias de *benchmark*, foram encontradas boas soluções quando comparadas às soluções já conhecidas. Pham et al. (2007) e Serapião (2009) resolveram o problema de escalonamento de *job shop*. Pham et al. (2007) resolveram os problemas de junção de soldas e de projeto de células de manufatura através do ABC, e em 2008 resolveram também o problema da cinemática inversa de robôs.

Karaboga e Gorkemli (2011) propuseram o ABC para solução do problema do caixeiro viajante. Li et al. (2011) também resolveram este problema, mas propuseram o ABC em conjunto com a abordagem de vizinhança. Akay e Karaboga (2012) apresentaram um algoritmo ABC para problemas de grande escala na otimização de projetos de engenharia, aumentando também o tamanho da colônia de abelhas à medida que o problema se tornava maior.

Zhang et al. (2011) descreveram um ABC melhorado com base na escala de *fitness* e também na teoria caótica, utilizando o algoritmo para treinar uma rede neural para o problema de classificação de imagens de ressonância magnética do cérebro.

Akbari et al. (2011) introduziram um método ABC para resolver problemas de otimização com múltiplos objetivos. Rubio-Largo et al. (2011) aplicaram o ABC multi-objetivo ao problema de roteamento estático e também ao problema de atribuição de comprimento de onda.

Várias foram as modificações aplicadas ao ABC para melhorar o seu desempenho e vários foram os trabalhos que utilizaram este método na resolução de problemas diferenciados. Além dos trabalhos apresentados, referentes à aplicação do ABC, existem outros trabalhos aplicados que são baseados em diferentes características comportamentais das abelhas como o *Bee Colony Optimization Metaheuristic* (BCO) encontrado em (TEODORIC E DELL'ORCO, 2005), *Bees Swarm Optimization* (BSO) encontrado em (DRIAS ET AL., 2005), *Virtual Bee Algorithm* (VBA) encontrado em (YANG, 2005), *Bee System* (BS) encontrado em (LUCIC E TEODOROVIC, 2001), *Honey Bee Algorithm* (HBA) encontrado em (NAKRANI e TOVEY, 2003), entre outros que podem ser encontrados na literatura. Vale ressaltar que o ABC foi escolhido por ser mais adequado a problemas de otimização combinatória, que é o foco deste trabalho.

## 4 METODOLOGIA

Neste capítulo é apresentada a modelagem matemática do PCC adotada neste trabalho, e também a forma de emprego das ferramentas computacionais propostas para a solução do problema de carregamento de navios-contêiner. As ferramentas computacionais mostradas no capítulo 2, o Algoritmo Genético e o Enxame de Abelhas, são detalhadas visando à solução do PCC. Para cada método, será feito um estudo detalhado da codificação que será empregada, através do uso de regras de carregamento e descarregamento. Neste capítulo são também apresentados os operadores propostos para a aplicação dos algoritmos, como também suas principais características, visando à construção de planos de carga que minimizem a quantidade de movimentações.

### 4.1 Modelo Matemático

A formulação matemática adotada para a modelagem do problema foi baseada nos trabalhos de Avriel e Penn (1993) e Avriel et al. (1998). É utilizado um modelo de programação inteira binária com a função objetivo de minimização e as restrições sendo as mesmas dos trabalhos referenciados. As restrições que serão estudadas se referem à alocação de contêineres em pilhas, à garantia de no máximo um contêiner ocupar uma posição em cada segmento de rota, movimentos de troca (remanejamentos) e capacidade do navio.

A modelagem do problema baseia-se em um navio-contêiner que possui apenas uma baía para armazenamento. Esta baía possui a mesma estrutura mencionada no Capítulo 2, com linhas horizontais e verticais. Na prática, uma baía tem formato tridimensional, mas neste trabalho será tratada como bidimensional, uma vez que não são consideradas restrições de estabilidade, sem perder seus princípios elementares, facilitando a utilização de matrizes.

As linhas horizontais são identificadas por  $r = 1, \dots, R$ , onde o valor 1 indica a parte inferior do navio e  $R$  indica o topo. As linhas verticais são identificadas por  $c = 1, \dots, C$ , onde o valor 1 indica esquerda e  $C$  indica direita. Desta forma, a quantidade máxima que a baía poderá alocar é de  $R \times C$  contêineres.

O navio inicia vazio no Porto 1, em seguida visita os portos 2, 3, ...  $N$ . Em cada porto  $i = 1, \dots, N - 1$ , o navio carrega e descarrega contêineres com destino  $j = i + 1, \dots, N$ . Quando chegar ao último porto do trajeto, o navio deve ficar vazio. O modelo define também a matriz de transporte,  $T = [T_{ij}]$  de tamanho  $(N - 1) \times (N - 1)$ , onde  $T_{ij}$  representa a matriz de contêineres com origem no porto  $i$  e destino no porto  $j$  (AVRIEL et al, 1998).

Assim, define-se uma variável binária  $x_{ijv}(r, c)$ , que pode assumir valor igual a 1 na posição  $(r, c)$  se for carregado um contêiner no Porto  $i$  com destino em  $j$  e movimentação em  $v$ , e 0 em caso o contrário. De maneira similar, é definida uma variável  $y_i(r, c)=1$ , se a posição  $(r, c)$  estiver ocupada por um contêiner, e 0 se não estiver.

Sem perda de generalidade, assume-se que o custo de movimentação por contêiner é o mesmo para qualquer contêiner e em qualquer porto. Assim, o objetivo é minimizar o número de movimentações de contêineres, com base no seguinte modelo de programação linear:

A Função Custo é definida conforme a equação (18):

$$f(x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \sum_{v=i+1}^{j-1} \sum_{r=1}^R \sum_{c=1}^C x_{ijv}(r, c), \quad (18)$$

sujeita às restrições definidas nas equações (19) – (23).

A equação (19) determina  $T_{ij}$ , ou seja, a quantidade de contêineres carregados no Porto de origem  $i$  e descarregados no Porto de destino  $j$ .

$$\sum_{v=i+1}^j \sum_{r=1}^R \sum_{c=1}^C x_{ijv}(r, c) - \sum_{k=1}^{i-1} \sum_{r=1}^R \sum_{c=1}^C x_{kji}(r, c) = T_{ij}; i = 1, \dots, N-1; j = i+1, \dots, N. \quad (19)$$

A equação (20) determina se a posição  $(r, c)$  está ocupada ou desocupada em cada segmento de rota. Garante que não existem rotas com mais de um contêiner nesta posição.

$$\sum_{k=1}^i \sum_{j=i+1}^N \sum_{v=i+1}^j x_{kqv}(r, c) = y_i(r, c); i = 1, \dots, N-1; r = i, \dots, R; c = i, \dots, C. \quad (20)$$

A restrição (21) garante o carregamento de contêineres em pilhas, pois se houver um contêiner em  $(r+1, c)$  então também deve existir um contêiner em  $(r, c)$ . Caso contrário, a solução será negativa e não obedecerá à restrição.

$$y_i(r, c) - y_i(r+1, c) \geq 0; i = 1, \dots, N-1; r = i, \dots, R-1; c = 1, \dots, C. \quad (21)$$

Na equação (22) é definido o movimento de remanejamento. Se um contêiner que ocupa a posição  $(r, c)$  é descarregado no porto  $j$ , então, ou não existem contêineres acima dele, ou o índice  $v$  do contêiner que ocupa o compartimento  $(r+1, c)$  não é maior que  $j$ .

$$\sum_{i=1}^{j-1} \sum_{p=j}^N x_{ipj}(r, c) + \sum_{i=1}^{j-1} \sum_{p=j+1}^N \sum_{v=j+1}^p x_{ipv}(r+1, c) \leq 1; j = 2, \dots, N; r = 1, \dots, R-1; c = 1, \dots, C. \quad (22)$$

Restrição (23) garante a integralidade das variáveis.

$$x_{ijv}(r, c) = 0 \text{ ou } 1; \quad y_i(r, c) = 0 \text{ ou } 1. \quad (23)$$

## 4.2 Ferramentas computacionais

Nesta seção, são apresentados os detalhes necessários para a implementação das técnicas de solução. Antes da aplicação das técnicas, é explicado como será feita a representação das soluções, através do emprego de regras. As técnicas de solução são baseadas nas regras apresentadas a seguir, capazes de representarem boas soluções. Quanto aos algoritmos AG e SB, são abordados os tipos de operadores e codificação que serão aplicados.

### 4.2.1 Representação por Regras

A proposta para a implementação dos algoritmos está baseada na representação das soluções através de regras, conforme proposto em Azevedo (2010). Logo, para ocupar o navio são definidas regras de carregamento em conjunto com regras de descarregamento de contêineres, empregadas para desocupação do navio.

A partir da análise dos resultados de Azevedo (2010), aplicando quatro combinações de regras para resolver o problema, nota-se uma limitação no uso de apenas quatro combinações de regras, pois somente com estas regras não é possível abordar todas as possibilidades de arrumações dos contêineres no navio. Por esta razão, este estudo propõe novas regras de carregamento e descarregamento para testar em conjunto com as regras criadas por Azevedo (2010).

Para tornar viável a representação da localização dos contêineres no navio, será utilizada uma representação matricial que mostra a forma de ocupação do navio, denominada matriz de ocupação. A representação através de uma matriz possibilita a visualização da quantidade de contêineres, como também os espaços disponíveis no navio e a posição dos contêineres, de acordo com seu porto de destino. Um modelo de matriz de ocupação inicial é apresentado na Figura 8. Este modelo assume que o problema apresenta uma rota com 5 portos e um navio com capacidade para 16 contêineres.

**Figura 8 - Matriz de Ocupação Inicial**

0	0	0	0
2	2	3	2
4	4	2	3
5	5	5	5

Fonte: Azevedo (2010).

Assim, na Figura 8, pode-se observar os espaços vazios da matriz, indicados pelo número zero, e os espaços preenchidos, onde há números diferentes de zero, significando que existem contêineres. Desta forma, o elemento (1,1), no canto inferior esquerdo, igual a 5, significa que nesta posição existe um contêiner que tem como destino o Porto 5. Os elementos são acessados

conforme a estrutura celular do navio apresentada no capítulo 2. Por exemplo, para retirar o contêiner da posição (2,3), com destino no porto 2, é necessário primeiro retirar o contêiner que está posicionado na posição (3,3), mais acima, com destino no porto 3. Após a retirada do contêiner desejado, é feita a realocação do contêiner movimentado de maneira adicional.

As atividades que modificam a matriz de ocupação são as operações de carregamento e descarregamento de contêineres, realizadas em cada porto. A maneira como serão feitas essas operações fica a cargo das regras que serão adotadas. A vantagem da abordagem por regras está na garantia da obtenção de soluções factíveis, e também na abordagem mais compacta que outras encontradas na literatura (AZEVEDO, 2010).

A abordagem através de regras é definida a partir da combinação de uma regra de descarregamento e outra de carregamento para o navio, ou seja, uma combinação  $k$  usada em um porto  $j$  significa a união destas duas operações de carga e descarga. Em adição ao trabalho de Azevedo (2010), que criou quatro combinações, a partir da definição de duas regras de carregamento (C1 e C2) e duas regras de descarregamento (D1 e D2), nesta pesquisa destaca-se como contribuição a criação de outras quatro regras de carregamento (C3, C4, C5 e C6), e uma de descarregamento (D3), que combinadas com as regras já existentes totalizam 18 combinações de regras. Na Tabela 1 são definidas as regras de carga e descarga que resultam na respectiva combinação  $k$ . Ao longo do trabalho, para fins de padronização, quando for mencionada uma combinação  $k$  ( $k = 1, 2, 3, \dots, 18$ ), estará se tomando por base as regras de carga e descarga exibidas na Tabela 1.

**Tabela 1 - Definição das Combinações das Regras de Movimentação de Contêineres**

<b>Combinação <math>k</math> usada no porto <math>j</math></b>	<b>Regra de carregamento</b>	<b>Regra de descarregamento</b>
<b>1</b>	C1	D1
<b>2</b>	C1	D2
<b>3</b>	C2	D1
<b>4</b>	C2	D2
<b>5</b>	C3	D1
<b>6</b>	C3	D2
<b>7</b>	C4	D1
<b>8</b>	C4	D2
<b>9</b>	C5	D1
<b>10</b>	C5	D2
<b>11</b>	C6	D1
<b>12</b>	C6	D2
<b>13</b>	C1	D3
<b>14</b>	C2	D3
<b>15</b>	C3	D3

16	C4	D3
17	C5	D3
18	C6	D3

Para exemplificar as etapas de aplicação das regras, é utilizada uma matriz de transporte  $T$  (Figura 9), mostrando a quantidade de contêineres que devem ser carregados no porto de origem  $i$  ( $O_i$ ) e com destino em  $j$  ( $D_j$ ). Neste exemplo, assume-se como portos de origem, onde podem ser efetuados carregamentos, do Porto 1 até o Porto 4. Por outro lado, os portos de destino, onde podem ser efetuados descarregamentos, são assumidos do Porto 2 até o Porto 5. Desta forma, para a aplicação das regras no exemplo que segue, no Porto 1 são empregados somente procedimentos de carregamento, pois o navio inicia a viagem no primeiro porto vazio. Analogamente, no Porto 5 apenas os procedimentos de descarregamento serão aplicados, pois ao final do percurso o navio não efetuará mais nenhum carregamento, devendo finalizar a viagem vazio. Para os demais portos, são aplicadas efetivamente as combinações de regras de movimentação de contêineres, compostas por uma regra de carregamento e outra de descarregamento.

**Figura 9 - Matriz de Transporte**

<b>O1</b>	2	5	0	0
<b>O2</b>	0	2	3	1
<b>O3</b>	0	0	2	2
<b>O4</b>	0	0	0	1
	<b>D2</b>	<b>D3</b>	<b>D4</b>	<b>D5</b>

Fonte: Azevedo (2010)

Conforme a Figura 9, e admitindo que se esteja no Porto 1, pode-se interpretar que foram carregados dois contêineres com destino no Porto 2 e cinco contêineres com destino no Porto 3 e assim sucessivamente. Ao final, são descarregados, no Porto 5, um contêiner que teve origem no segundo porto, dois contêineres com origem no Porto 3 e um contêiner com origem no Porto 4.

As regras de carga são responsáveis pela entrada de novos contêineres no navio. Logo, as regras de descarga desempenham operações de desembarque de contêineres em seus portos de destino, e ainda, operações de realocações de contêineres bloqueantes, pois ainda não chegaram ao seu destino. A seguir, são definidas as regras de carga e descarga empregadas neste estudo.

- **Regra C1:** Nesta regra, é preenchida a matriz de ocupação inicial linha a linha, da esquerda para a direita, destinando a parte inferior da pilha para cargas com destinos mais distantes. Considerando a matriz de transporte da Figura 9, e assumindo que encontra-se no Porto 1, a matriz de ocupação resultante é representada na Figura 10.

**Figura 10 - Matriz de Ocupação após Regra C1**

0	0	0	0
0	0	0	0
3	2	2	0
3	3	3	3

Fonte: Azevedo (2010)

- **Regra C2:** Em um porto  $p$ , esta regra preenche a matriz de ocupação, a cada coluna, da esquerda para a direita, até a linha  $\theta_p$ . Para calcular qual linha representa a linha  $\theta_p$  na matriz, o cálculo é realizado após o descarregamento dos contêineres com destino em  $p$ . Conforme a equação (24), os contêineres existentes no navio são somados aos contêineres que serão carregados no porto  $p$ , dividido pela quantidade de colunas existentes na matriz de ocupação (AZEVEDO, 2010). Assim,

$$\theta_p = \left\lceil \frac{\sum_{i=1}^p \sum_{j=p+1}^N T_{ij}}{C} \right\rceil, \quad (24)$$

onde  $p$  é o porto atual,  $T_{ij}$  é o número de contêineres a serem embarcados no porto  $i$  e desembarcados no porto  $j$ , e  $C$  é o número de colunas da matriz de ocupação inicial. Aplicando esta equação, a matriz de ocupação, após o emprego da Regra C2, será preenchida até a linha 3.

Supondo que o navio se encontre no Porto 2, é aplicada a Regra C2 que carrega primeiramente os contêineres com destino mais distante, preenchendo a coluna até a altura máxima ( $\theta_p$ ) e assim percorrendo todas as colunas. Assim, a matriz de ocupação resultante, após a aplicação da Regra C2, é apresentada na Figura 11, baseada nas quantidades de contêineres definidas na matriz de transporte da Figura 9.

**Figura 11 - Matriz de Ocupação da Regra C2**

0	0	0	0
4	3	3	0
4	3	3	3
5	4	3	3

Fonte: Azevedo (2010)

- **Regra C3:** Nesta regra, a matriz de ocupação inicial é preenchida linha a linha, da direita para a esquerda, destinando a parte inferior da pilha para cargas com destinos mais distantes. Considerando que o navio se encontre no Porto 2, a matriz de ocupação resultante é representada na Figura 12.

**Figura 12 - Matriz de Ocupação da Regra C3**

0	0	0	0
0	3	3	3
3	3	3	3
4	4	4	5

- **Regra C4:** Da mesma forma que as demais regras, a Regra C4 destina a parte inferior do navio para cargas com destinos mais distantes. Os contêineres são carregados coluna a coluna, da direita para a esquerda, com limite até a linha  $\theta_p$ , definida pela equação (24). A matriz de ocupação resultante é apresentada na Figura 13, admitindo que o navio esteja no Porto 2.

**Figura 13 - Matriz de Ocupação da Regra C4**

0	0	0	0
0	3	3	4
3	3	3	4
3	3	4	5

- **Regra C5:** Nesta regra os contêineres são carregados da esquerda para a direita, coluna a coluna, sendo limitado até a coluna  $\beta_p$  da matriz. Para a determinação da coluna  $\beta_p$ , emprega-se a equação (25):

$$\beta_p = \left\lfloor \frac{\sum_{i=1}^p \sum_{j=p+1}^N T_{ij}}{R} \right\rfloor \quad (25)$$

A equação (25) é semelhante à equação (24) que calcula o  $\theta_p$ , porém emprega-se o número de linhas como denominador da expressão e não o número de colunas, o que resulta em uma coluna limitante da matriz de ocupação. No exemplo a seguir (Figura 14), a matriz de ocupação apresentada é resultante da aplicação da Regra C5, admitindo que o navio esteja no Porto 2 e tendo sido calculada a coluna limite em  $\beta_p = 3$ .

**Figura 14 - Matriz de Ocupação da Regra C5**

4	3	0	0
4	3	3	0
4	3	3	0
5	3	3	0

- **Regra C6:** Na última regra de carregamento, os contêineres são carregados da direita para a esquerda, coluna a coluna, tendo como limite a coluna  $\beta_p$  da matriz. Este valor limite é calculado conforme a equação (25). Assim, a matriz de ocupação resultante após empregada a regra C6, assumindo o navio no Porto 2, é mostrada na Figura 15.

Figura 15 - Matriz de Ocupação da Regra C6

0	0	3	4
0	3	3	4
0	3	3	4
0	3	3	5

- **Regra D1:** Ao chegar ao porto  $j$  (destino), são descarregados todos os contêineres com destino em  $j$ , conseqüentemente são descarregados também todos os contêineres bloqueantes, ou seja, aqueles que estão localizados acima dos contêineres que devem ser descarregados. A mesma regra D1 realoca os contêineres bloqueantes baseando-se na regra de carregamento C1. Considerando que o navio tenha chegado ao Porto 2, a Figura 16 apresenta a matriz de ocupação antes (à esquerda) e após (à direita) a aplicação da Regra D1.

Figura 16 - Antes e Depois aplicação da regra D1

0	0	0	0
5	5	2	2
3	2	3	5
2	3	3	4

0	0	0	0
0	0	0	0
5	3	3	5
5	3	3	4

Fonte: Azevedo (2010)

- **Regra D2:** Ao chegar ao Porto  $j$ , todos os contêineres são removidos permitindo que todas as pilhas sejam reordenadas. A reordenação dos contêineres com destino nos próximos portos é baseada na Regra C1, mas desenvolvida na própria regra D2. Supondo que o navio tenha chegado ao Porto 2, a Figura 17 apresenta a matriz de ocupação antes (à esquerda) e após (à direita) a aplicação da Regra D2.

**Figura 17 - Antes e Depois aplicação da regra D2**

0	0	0	0
5	5	2	2
3	2	3	5
2	3	3	4

0	0	0	0
0	0	0	0
3	3	3	3
5	5	5	4

Fonte: Azevedo (2010)

- **Regra D3:** Ao chegar ao porto  $j$ , são descarregados todos os contêineres com destino em  $j$ , e por consequência, os contêineres que estiverem bloqueando a retirada destes. Após realizar todo processo de descarga do navio, os contêineres bloqueantes que foram retirados, são realocados no navio, sendo inserido da esquerda para a direita, ocupando cada coluna onde existirem espaços vagos. Supondo que o navio tenha chegado ao Porto 2, a Figura 18 apresenta a matriz de ocupação antes (à esquerda) e após (à direita) a aplicação da Regra D3.

**Figura 18 - Antes e Depois aplicação da regra D3**

0	0	0	0
5	5	2	2
3	2	3	5
2	3	3	4

0	0	0	0
3	0	0	0
5	0	3	5
5	3	3	4

O emprego de regras para a codificação das soluções do problema de carregamento de contêineres reduz o número de variáveis do problema, possibilitando uma varredura mais completa do espaço de soluções. Entretanto, o uso reduzido de regras pode limitar a representação das possíveis soluções para o problema. Por esta razão, neste trabalho estão sendo propostas quatro novas regras de carregamento (C3, C4, C5 e C6) e uma nova regra de descarregamento (D3). O emprego destas novas regras em conjunto com as regras propostas nas literaturas (C1, C2, D1 e D2) permite a criação de matrizes de ocupação factíveis, bem como soluções promissoras, pois adota uma representação mais compacta.

A seguir, são abordados os métodos aplicados na busca por soluções do problema, empregando a representação por regras.

#### 4.2.2 A metaheurística Algoritmo Genético

Para a implementação do Algoritmo Genético (AG), é necessário definir seus operadores e os valores dos parâmetros empregados para o seu funcionamento. A seguir, será

definida a forma de codificação da solução, o operador de cruzamento e o operador de mutação. Conforme Chaves (2010), o AG é uma das técnicas mais indicadas para solução de problemas de otimização.

#### 4.2.2.1 A codificação

A codificação empregada para a representação do cromossomo (indivíduo) é feita, através do uso das combinações de regras de carga e descarga. Na Figura 19 é apresentado um exemplo de codificação para uma rota constituída de apenas 3 portos.

**Figura 19 - Codificação do Cromossomo**

2	4	1
---	---	---

Fonte: Azevedo (2010)

Conforme é mostrado na Figura 19, o cromossomo é representado pelas combinações de regras a serem utilizadas nos três portos da rota. No Porto 1, será utilizada a combinação 2, no Porto 2 a combinação 4 e no Porto 3 a combinação de regras 1. A codificação aplicada emprega a representação por números inteiros e, em consequência, os demais operadores também são específicos para se trabalhar com números inteiros. Os operadores empregados são definidos de acordo com a representação por combinação de regras do problema.

#### 4.2.2.2 O Operador de Seleção

O operador de seleção adotado foi o torneio (MITCHELL, 1996). Inicialmente é definida uma probabilidade de 75% para aceitação do melhor indivíduo quando realizado o torneio. Para realizar o torneio são escolhidos dois indivíduos da população, aleatoriamente. Para selecionar um dos indivíduos, é feita a escolha de um valor randomicamente, com distribuição uniforme, no intervalo [0, 1]. Se o número estiver abaixo de 75%, o que acontecerá na maioria das vezes, seleciona-se o melhor indivíduo como o próximo pai, caso contrário seleciona-se o pior indivíduo. Para seleção de um segundo indivíduo pai, repete-se o mesmo processo de seleção.

#### 4.2.2.3 O Operador de Cruzamento

O operador de cruzamento é aplicado após a etapa de seleção, onde os dois cromossomos pais selecionados são empregados para gerar o(s) cromossomo(s) filho(s). Este operador tem a função de combinar dois indivíduos, a fim de produzir novos indivíduos que herdarão as características de seus antecessores. Ao aplicar o cruzamento, nesta pesquisa que

trabalha com a codificação do cromossomo através de regras, o cruzamento se dá na troca de regras (características) entre os indivíduos.

Neste estudo serão aplicados dois tipos diferentes de cruzamento, especificados a seguir. Os operadores de cruzamento empregados são, o Cruzamento Uniforme e o Cruzamento Um Ponto de Corte (MALAQUIAS; AZEVEDO, 2006, 2010).

O Cruzamento Uniforme consiste no sorteio de um cromossomo pai para a determinação de cada gene do indivíduo filho. Cada gene do indivíduo filho é preenchido com a característica localizada na posição correspondente do indivíduo pai selecionado. Considerou-se para este cruzamento que os dois indivíduos pais tem a mesma probabilidade de serem escolhidos. Assumindo o Pai 1 e o Pai 2 apresentados na Figura 20, o Filho resultante da aplicação do cruzamento uniforme é apresentado na Figura 21.

**Figura 20 - Pai 1 e Pai 2 para o cruzamento**

Pai 1	4	1	6	3	5	2
Pai 2	2	5	3	6	1	4

Na Figura 21, observa-se que o primeiro gene contém a regra 4, que teve origem no Pai 1, no segundo gene a regra 1 que teve origem no Pai 1, no terceiro gene, a regra 3 teve origem no Pai 2, e assim sucessivamente.

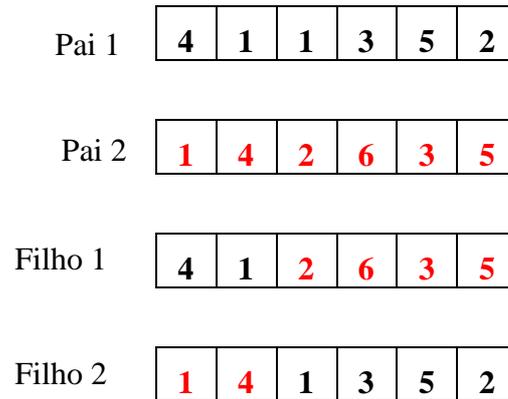
**Figura 21 - Indivíduo após Cruzamento Uniforme**

Filho	4	1	3	3	1	2
-------	---	---	---	---	---	---

O operador Cruzamento Um Ponto de Corte consiste no sorteio de um número aleatório, com distribuição uniforme, dentro do intervalo  $[1, N-1]$ , onde  $N$  indica o número de portos da rota a serem visitados. O número selecionado se refere ao ponto de corte que vai permitir que sejam gerados dois filhos, a partir de dois cromossomos pais. Ao selecionar o ponto de corte, o primeiro indivíduo filho será formado pelas posições à esquerda deste ponto, com base no Pai 1, sendo completadas com as posições a direita do ponto de corte no Pai 2.

No exemplo apresentado na Figura 22, foi selecionada, aleatoriamente, a posição 2 como ponto de corte. Logo, o Filho 1 resulta do Pai 1 até o gene 2 e do Pai 2 a partir do gene 3. Já o Filho 2 inicia com as características do Pai 2 até o ponto de corte, sendo completado com as características do Pai 1, à direita do ponto de corte. Depois de gerados os dois filhos, é selecionado, aleatoriamente, um dos filhos para prosseguir na próxima etapa do algoritmo.

Figura 22 - Operador Um Ponto de Corte



Para saber se o operador de cruzamento será aplicado, é sorteado um número aleatório com distribuição uniforme no intervalo  $[0, 1]$ , e após é avaliado se este valor atende à taxa de cruzamento ( $pc$ ), definida previamente. Neste estudo define-se, inicialmente, a taxa de cruzamento  $pc = 0,9$ , de forma que para todos os valores sorteados entre 0 e 0,9, será realizado o cruzamento. Para valores acima de 0,9 o cruzamento não é realizado, sendo os cromossomos pais escolhidos assumidos como cromossomo(s) filho(s), isto é, idênticos aos cromossomos pais (CHAN, 1998).

#### 4.2.2.4 O Operador de Mutação

O operador de mutação permite a diversificação da população, ou seja, que o algoritmo explore novas regiões do espaço de busca. Esta diversidade é obtida através da inserção de combinações de regras diferentes daquelas presentes no cromossomo. Neste trabalho, será aplicado o operador de Mutação Uniforme. A Mutação Uniforme é composta pelo sorteio de uma posição do cromossomo e seguida pelo sorteio de uma regra diferente da encontrada na posição sorteada (MITCHELL, 1996). No exemplo apresentado na Figura 23, foram sorteadas as posições 1, 3 e 4, sendo os valores das combinações de regras presentes nestas posições substituídos por valores diferentes dos originais, sorteados entre todas as combinações de regras possíveis.

Figura 23 - Mutação Uniforme



No caso de não ocorrer mutação, o cromossomo permanece idêntico ao cromossomo resultante após o cruzamento. Para saber se haverá mutação, o procedimento de avaliação realizado é da mesma forma que no cruzamento, sorteando-se um número com distribuição uniforme no intervalo  $[0, 1]$  e verificando-se a probabilidade de realização da mutação.

Para o emprego da mutação, é admitida uma taxa de execução previamente definida. Para a definição da probabilidade de mutação ( $pm$ ), foram utilizadas duas estratégias:

- Um valor fixo pequeno ( $pm = 0,05$ , por exemplo), mantido inalterado ao longo de toda a execução do AG. Uma vez avaliada a possibilidade de mutação de um indivíduo, o operador de mutação uniforme será aplicado em 20% dos genes do indivíduo, escolhidos aleatoriamente;
- Adaptativa, definida conforme o grau de aptidão do indivíduo e calculada através da equação (26) (AZEVEDO, 2010):

$$pm_{it} = 0,5 \times \left( \frac{Max(Fit(A_j^t)) - Fit(A_i^t)}{Max(Fit(A_j^t)) - \overline{Fit(A_j^t)}} \right), \quad (26)$$

onde  $Fit(A_i^t)$  corresponde à aptidão do indivíduo  $A_i^t$  na geração  $t$ ,  $\overline{Fit(A_j^t)}$  é a média da aptidão da população na geração  $t$ , e  $Max(Fit(A_j^t))$  é o maior valor de aptidão na geração  $t$ . Para todo o indivíduo da população, é avaliada a possibilidade de mutação gene a gene, ou seja, dependendo da qualidade do indivíduo, ele terá mais ou menos genes alterados. Quando efetuada, a alteração é realizada com base no operador de mutação uniforme.

#### 4.2.2.5 Critério de Parada

Além dos operadores genéticos citados, é fundamental determinar o critério de parada, que deve ser definido previamente. Neste trabalho, adotou-se o critério baseado no número máximo de gerações.

#### 4.2.3 A metaheurística Enxame de Abelhas

No algoritmo da Colônia Artificial de Abelhas (ABC), é necessário determinar parâmetros como, o tamanho da população e o número máximo de ciclos. Quanto à codificação das fontes de alimento, será análoga à codificação utilizada no Algoritmo Genético, através do emprego de combinações de regras.

O tamanho da população sugerido é de 50 fontes de alimentos e o número máximo de gerações será um valor não inferior a 1000. O nível de aptidão será avaliado conforme o néctar de cada fonte de alimento, determinando a qualidade da fonte de alimento para que as abelhas observadoras efetuem a sua escolha posteriormente. Como critério de parada, será considerado o número máximo de iterações (KARABOGA, 2010).

#### 4.2.3.1 Operador de vizinhança

Os operadores de vizinhança que serão utilizados serão apropriados à representação por números inteiros e a codificação por combinações de regras, tendo como objetivo a geração de soluções similares a uma solução (vizinho) porém com características diferentes. Os operadores propostos para emprego na aplicação do ABC são Inversão de uma Subsequência e Troca Aleatória, (SZETO; MARINHO, 2010, 2005).

O operador de Inversão de uma Subsequência seleciona aleatoriamente um determinado número de posições, em seguida os elementos das posições selecionadas são invertidos de forma que a primeira posição selecionada agora seja a última e assim sucessivamente. Por exemplo, suponha que as posições 3, 4 e 5 do indivíduo, correspondendo às combinações de regras 6, 3 e 5, tenham sido selecionadas. Com a aplicação do operador, as combinações das posições selecionadas são invertidas resultando em um novo indivíduo, conforme apresentado na Figura 24.

**Figura 24 - Operador Inversão de uma Subsequência**

4	1	6	3	5	2
4	1	5	3	6	2

O operador Troca Aleatória seleciona posições no indivíduo, aleatoriamente, e faz uma troca (*swap*) dos valores nas posições selecionadas. No exemplo da Figura 25, são selecionadas as posições 3 e 6, e através da realização da troca, o valor que estava na posição 3 mudará para posição 6, e da mesma forma acontece com o valor da posição 6 que mudará para a posição 3.

**Figura 25 - Operador Troca Aleatória**

4	1	6	3	5	2
4	1	2	3	5	6

Após determinar os operadores que serão utilizados em cada método e as combinações de regras de codificação do problema, a discussão passa a ser a forma de aplicação destes métodos. A seguir, no Capítulo 5, os experimentos de validação são discutidos, a fim de identificar as melhores configurações de simulação.

## 5 VALIDAÇÃO

Neste capítulo serão apresentados detalhes com relação à implementação do Algoritmo Genético e do Algoritmo Enxame de Abelhas, quando aplicados na solução do problema de carregamento e descarregamento de navios-contêineres. Com base nas instâncias empregadas em Azevedo (2010), será feita a validação dos algoritmos com base na comparação com os resultados apresentados pelo autor. Antes da etapa de validação, serão especificados detalhes acerca da implementação dos algoritmos, das instâncias, dos tipos de matrizes de transporte, dos parâmetros adotados e das combinações de regras utilizadas.

### 5.1 Tipos de Matrizes de Transporte

Para avaliação das técnicas de solução adotadas, serão utilizados diferentes tipos de matrizes de transporte: 1 – Distância Mista, 2 – Longa Distância e 3 – Curta distância, conforme proposto em (Avriel et al. 1998). Estes tipos são definidos de acordo com o tempo médio que um contêiner permanece a bordo no navio, seguindo a equação (27).

$$\sum_{i=1}^k \sum_{j=k+1}^n T_{ij} \leq R \times C \text{ para todo } k = 1, \dots, n - 1. \quad (27)$$

As matrizes de **curta distância** são construídas assumindo contêineres que ficam um período curto a bordo. Para a construção da matriz de transporte, são consideradas as restrições impostas pela equação (27) e a ordem em que os elementos da matriz foram selecionados. As matrizes são criadas iniciando pelo elemento do canto superior esquerdo e seguindo com os elementos da esquerda para a direita em todas as linhas. No entanto, os elementos próximos da diagonal tornam-se grandes. Isto ocorre devido à determinação antecipada de valores, antes mesmo de determinar os demais valores da linha. Com isso, torna-se alta a ocorrência de realocações de contêineres relativos a estas posições na matriz de transporte.

Nas matrizes de **distância mista**, os elementos são determinados em ordem aleatória. A cada etapa, os valores de  $i$  e  $j$  são escolhidos, aleatoriamente, com base em uma distribuição uniforme. Após esta escolha, inicia-se na posição  $(i, j)$  verificando os elementos em ordem crescente que contenham tal elemento. Caso não for encontrado, é verificado em ordem decrescente. Fixando  $(i, j)$ , o valor  $T_{ij}$  é escolhido aleatoriamente tal que a equação (27) seja satisfeita. Tal equação determina que os elementos próximos da diagonal sejam um pouco maiores do que os demais elementos.

As matrizes de **longa distância** são caracterizadas por contêineres que permanecem um longo tempo a bordo. As matrizes são criadas de forma semelhante às de distância mista, com a adição da restrição  $T_{ij} \leq \left(\frac{j-i}{n-1}\right)^2$ , que implica com que os elementos pertos da diagonal sejam relativamente pequenos (AVRIEL, 1998).

## 5.2 Instâncias do Problema

Para avaliação dos algoritmos abordados neste trabalho, são empregados alguns problemas teste. Estes problemas, também chamados de instâncias de teste, representam os dados de entrada de um algoritmo, sendo gerados conforme o problema.

Para realização dos experimentos, são utilizadas as instâncias disponibilizadas em Azevedo (2009), possibilitando a comparação entre os métodos deste trabalho, bem como com os resultados apresentados em (AZEVEDO, 2010). Através do emprego destas instâncias, foram realizados experimentos de validação e testes de desempenho das diferentes técnicas empregadas.

As instâncias empregadas são classificadas de acordo com o número de portos e o tipo de matriz de transporte utilizada. São utilizadas instâncias assumindo os três tipos de matriz de transporte, definidas anteriormente, sendo também considerados diferentes valores para a quantidade de portos compondo a rota de um navio. Serão assumidos valores entre 10 e 30 para a quantidade de portos na definição das rotas dos navios, com uma variação de 5 portos entre cada quantidade. Desta forma, serão cinco configurações de portos para os três tipos de matriz de transporte, resultando em um total de 15 problemas teste. Além disso, todas as instâncias assumem uma capacidade máxima do navio fixa de 300 contêineres, com dimensões de  $R = 6$  linhas e  $C = 50$  colunas (AZEVEDO, 2010).

## 5.3 Parâmetros

O objetivo dos testes, neste momento, é a comparação das metaheurísticas propostas com o estudo de Azevedo (2010). Por esta razão, serão definidos valores fixos para os parâmetros empregados em cada um dos algoritmos implementados, a fim de validá-los.

Na validação do AG, utiliza-se o Cruzamento Uniforme com probabilidade de 90% e a Mutação Uniforme com probabilidade fixa de 15%. As regras utilizadas são as mesmas 4 regras utilizadas por Azevedo (2010) em seus testes, duas regras de carregamento (C1, C2) e duas regras de saída (D1, D2). A população testada é de 50 indivíduos e o critério de parada é o número máximo de 1000 gerações. Cada resultado apresentado se refere ao valor médio dos testes realizados, encontrado após 30 execuções de cada algoritmo e para cada problema teste.

Na validação do ABC, utilizou-se a estrutura de vizinhança por troca, assumindo 100 fontes de alimento, que corresponde ao tamanho da população. O critério de parada é o número máximo de 1000 gerações e a média de resultados é calculada após 10 execuções do algoritmo. O número de execuções é menor em virtude de o algoritmo ABC apresentar uma exigência computacional maior se comparado ao AG. Se fossem geradas as 30 execuções do algoritmo, conforme foi feito para o AG, não haveria tempo hábil para finalizar o trabalho. Novamente, as regras empregadas foram as mesmas 4 regras utilizadas por Azevedo (2010) em seus testes.

## 5.4 Validação dos Métodos de Solução

Para a realização dos experimentos de avaliação dos algoritmos abordados neste trabalho, inicialmente os métodos são validados com todas as instâncias que serão utilizadas nestes experimentos. Ambos os algoritmos propostos nesta pesquisa foram implementados através do uso da ferramenta Matlab (MORAIS, 2012), e os experimentos para validação foram realizados em um computador com processador Dual Core 2.00GHz, memória RAM de 3GB e Sistema Operacional Windows 7.

Por ser um problema de minimização, os melhores resultados serão sempre os menores valores encontrados. Na Tabela 2, são expostos o número de movimentos mínimos para cada instância, definido com base nos valores que compõem cada Matriz de Transporte empregada e propostas por Azevedo (2009). Para fazer a contagem de movimentos, o algoritmo contabiliza os carregamentos de contêineres nos portos de origem e os descarregamentos de contêineres em seus portos de destino, acrescidos dos contêineres retirados desnecessariamente e que precisam ser realocados. Desta forma, o número de movimentos mínimo ideal deve indicar apenas os movimentos de carga e descarga, sem a existência de movimentos desnecessários.

**Tabela 2 - Movimentos Mínimos de contêineres**

<b>Navio-Contêiner com capacidade de 300 contêineres</b>			
<b>Portos</b>	<b>Tipo de Matriz de Transporte</b>		
	<b>1</b>	<b>2</b>	<b>3</b>
	FO	FO	FO
<b>10</b>	1322	750	2282
<b>15</b>	1580	778	3024
<b>20</b>	1990	784	4880
<b>25</b>	1664	944	5492
<b>30</b>	2262	1030	6380

Através dos valores apresentados na Tabela 2, pode-se observar, por exemplo, que o número mínimo de movimentos para o problema de 10 Portos e Matriz de Transporte do Tipo 1 é de 1322, e para um problema de 10 Portos e matriz do tipo 2, têm-se um mínimo de 750 movimentos, e assim sucessivamente para os demais casos. Para os testes de cada algoritmo, busca-se encontrar resultados próximos ou idênticos aos da Tabela 2.

Na Tabela 3, são exibidos os resultados médios obtidos por Azevedo (2010) em sua pesquisa, utilizando o Algoritmo Genético para resolver este problema. Os operadores utilizados foram o cruzamento OX com probabilidade de 80% e a mutação adaptativa que não possui probabilidade fixa. O tamanho da população foi fixado em 50 indivíduos e o critério de parada adotou um máximo de 1000 gerações.

O operador de cruzamento OX, que é muito semelhante ao operador de cruzamento um ponto de corte, consiste no sorteio de um número aleatório, com distribuição uniforme, dentro do intervalo  $[1, N-1]$ , onde  $N$  indica o número de portos a serem visitados. O número selecionado se refere ao ponto de corte, que desta vez vai gerar dois cromossomos filhos a partir de dois cromossomos pais.

Na Tabela 3, o valor de FO apresentado representa a média das 10 execuções do algoritmo, correspondendo ao total de movimentos realizados em todo o percurso para cada problema teste.

**Tabela 3 - Resultados Médios do AG de Azevedo**

<b>Navio-Contêiner com capacidade de 300 contêineres</b>			
<b>Portos</b>	<b>Tipo de Matriz de Transporte</b>		
	<b>1</b>	<b>2</b>	<b>3</b>
	FO	FO	FO
<b>10</b>	1438,00	1236,00	2288,00
<b>15</b>	2308,00	1194,40	3028,00
<b>20</b>	2810,20	1059,20	4880,00
<b>25</b>	2237,20	1673,00	5536,80
<b>30</b>	3235,20	1975,40	6384,00

**Fonte: Azevedo (2010)**

De acordo com os experimentos de Azevedo (2010), em seu estudo foi possível encontrar o resultado ótimo nos problemas testes com 10 Portos e Matriz tipo 3 e com 20 Portos e Matriz tipo 3. Vale observar que, os resultados relativos às instâncias da Matriz tipo 3 que não alcançaram o resultado ótimo, chegaram a soluções muito próximas. Em relação aos demais resultados, estes serão analisados em conjunto com os resultados encontrados na aplicação do AG e do ABC desta pesquisa.

#### 5.4.1 Algoritmo Genético

Os parâmetros adotados na validação do AG aqui proposto foram especificados na seção 5.3. Na Tabela 4, são expostos os resultados médios obtidos nos experimentos realizados com o Algoritmo Genético, e entre parênteses, como complemento, o valor do desvio padrão das 30 execuções. A fim de comparar o valor da FO obtido através do emprego do AG aqui desenvolvido com o AG proposto por Azevedo (2010), as regras de carga e descarga, empregadas em ambos os algoritmos, são as mesmas.

Segundo a Tabela 4, os resultados para o problema com 10 Portos e Matrizes tipo 1, 2 e 3 são idênticos aos resultados encontrados por Azevedo (2010). Já para o problema com 15 portos e Matrizes tipo 1 e 2, os resultados respectivos 2282 e 1172 encontrados se mostraram

melhores que os resultados 2308 e 1194,40 da Tabela 3. Apesar disso, os resultados obtidos ainda encontram-se distantes dos valores exatos, fornecidos pela Tabela 2.

No problema com 15 Portos e Matriz tipo 3, chegou-se à solução ótima também encontrada na Tabela 2. Para o problema com 20 portos, tanto a Matriz tipo 1 quanto a Matriz tipo 2, possuindo resultados médios de 2672,33 e 1022,33 alcançaram resultados melhores se comparados com a Tabela 3, onde os valores alcançados foram 2810,20 e 1059,20. Em relação à Matriz tipo 3, tanto o algoritmo proposto por Azevedo (2010) quanto o AG aqui desenvolvido encontraram a solução ótima.

**Tabela 4 - Resultados Médios do AG**

<b>Navio-Contêiner com capacidade de 300 contêineres</b>			
<b>Portos</b>	<b>Tipo de Matriz de Transporte</b>		
	<b>1</b>	<b>2</b>	<b>3</b>
	FO ( $\sigma$ )	FO ( $\sigma$ )	FO ( $\sigma$ )
<b>10</b>	1438,00 (0,00)	1236,00 (0,00)	2288,00 (0,00)
<b>15</b>	2282,00 (4,06)	1172,00 (0,00)	3028,00 (0,00)
<b>20</b>	2672,33 (10,62)	1022,33 (2,92)	4880,00 (0,00)
<b>25</b>	2136,86 (30,53)	1611,00 (33,45)	5520,66 (1,60)
<b>30</b>	2942,00 (43,69)	1872,00 (1,28)	6384,00 (0,00)

No problema teste com 25 Portos, todos os resultados médios obtidos se mostraram melhores do que os apresentados na Tabela 3, porém em nenhum dos Tipos de Matriz de Transporte foi alcançado o valor ótimo. Para o problema com 30 Portos, quando empregadas as Matrizes tipo 1 e 2, os resultados 2942,00 e 1872,00 respectivamente, foram melhores que os resultados encontrados por Azevedo (2010), com soluções 3235,20 e 1975,40, conforme a Tabela 3. Para o problema teste com Matriz tipo 3, ambos os algoritmos obtiveram o ótimo.

Em relação ao tempo gasto para as simulações, tem-se as médias para cada tamanho rota. Para a rota com 10 portos, é gasto em média 1 a 1,5 minutos, para 15 portos é gasto 2 a 2,5 minutos, na rota com 20 portos o tempo é de 2,5 a 4 minutos, para 25 portos de 4 a 4,5 minutos e para a rota com 30 portos é gasto de 4 a 5,7 minutos.

Como se pode observar, o Algoritmo Genético implementado nesta pesquisa teve desempenho por vezes idêntico e, na maioria das vezes, melhor que o Algoritmo Genético proposto por Azevedo (2010), o que valida o AG desenvolvido nesta pesquisa. Entretanto, com relação à obtenção de resultados próximos aos valores ótimos conhecidos, a implementação com 4 regras mostrou-se insuficiente.

#### 5.4.2 Enxame de Abelhas

Os parâmetros adotados na validação do ABC foram especificados na seção 5.3. As regras utilizadas no algoritmo, visando a sua validação, foram as mesmas utilizadas por Azevedo (2010). Os valores de FO apresentados na Tabela 5 correspondem à média de 10 execuções do ABC, para cada problema teste, e entre parênteses, como complemento, o valor do desvio padrão das soluções.

Na Tabela 5, observa-se que o valor médio obtido para a FO do problema com 10 portos e Matriz tipo 1, foi de 1461,63, que é mais elevado que o valor médio obtido com o AG aqui proposto (Tabela 4) e também maior que o valor médio obtido com o AG de Azevedo (2010), conforme Tabela 3. Para a mesma quantidade de portos e Matriz tipo 2, encontrou-se o valor médio de 1259,27 movimentos contra 1236,00 de Azevedo (2010), ou seja, a média encontrada foi de pior qualidade. Já para a Matriz tipo 3, a FO média obtida foi de 2288,00, idêntica aos resultados apresentados nas Tabelas 3 e 4.

Analisando o problema com 15 portos e Matrizes tipo 1 e 2, em ambos os casos foram obtidos valores médios de FO com qualidade inferior aos apresentados nas Tabelas 3 e 4. Vale observar que o desvio padrão do valor médio da FO é muito elevado, chegando a 82,60 para o caso da Matriz tipo 1 e 109,01 para o caso da Matriz tipo 2. O menor resultado encontrado para a Matriz tipo 1 foi de 2366,00 movimentos, que é muito próximo aos resultados apresentados nas Tabelas 3 e 4. Para a Matriz tipo 2, o menor resultado foi de 1182,00 movimentos, sendo melhor que a média apresentada por Azevedo (2010).

Para o caso da Matriz tipo 3, o valor médio obtido é idêntico ao valor obtido empregando-se o AG aqui proposto (Tabela 4) e ao valor fornecido por Azevedo (2010) (Tabela 3).

**Tabela 5 - Resultados médios do ABC**

<b>Navio-Contêiner com capacidade de 300 contêineres</b>			
<b>Portos</b>	<b>Tipo de Matriz de Transporte</b>		
	<b>1</b>	<b>2</b>	<b>3</b>
	FO ( $\sigma$ )	FO ( $\sigma$ )	FO ( $\sigma$ )
<b>10</b>	1461,63 (25,68)	1259,27 (13,39)	2288,00 (0,00)
<b>15</b>	2446,50 (82,60)	1273,33 (109,01)	3028,00 (0,00)
<b>20</b>	3121,14 (67,59)	1168,85 (134,09)	4881,00 (1,15)
<b>25</b>	2345,14 (49,31)	1706,66 (25,63)	5557,14 (14,68)
<b>30</b>	3539,14 (107,52)	2113,80 (88,12)	6384,80 (1,78)

No problema com 20 portos e Matrizes do tipo 1, 2 e 3, foram obtidos valores médios de FO de 3121,14, 1168,85, e 4881,00 com desvios-padrão de 67,59, 134,09 e 1,15 respectivamente. Estes valores médios indicam um maior número de movimentos de contêineres, significando resultados de pior qualidade, se comparados com o AG, apesar do valor médio da FO para o caso da Matriz tipo 3 ser muito próximo aos demais resultados para

o mesmo problema. Uma vez que para as instâncias com Matrizes de Transporte tipo 1 e 2 o desvio-padrão das soluções foi alto, são comparados também os menores valores encontrados para a FO em cada caso, tendo sido obtidos os valores de 2998,00 e 1072,00, sendo estes ainda piores que as soluções obtidas com o AG, apesar de muito próximas.

Com 25 Portos, o algoritmo ABC teve desempenho semelhante aos casos anteriores, tendo sido encontradas médias superiores aos resultados obtidos com o AG. O desvio padrão das soluções tem valor pequeno, não causando tanto impacto na análise dos resultados.

Para o problema com 30 portos, os resultados médios obtidos para as Matrizes do tipo 1, 2 e 3 foram maiores que aqueles apresentados por Azevedo (2010). Neste problema teste, os desvios-padrão obtidos para os casos das Matrizes 1 e 2 são 107,52 e 88,12. Com relação aos melhores resultados de FO, foram obtidos os valores 3388,00 e 1930, onde o segundo resultado é melhor que a média apresentada por Azevedo (2010) (Tabela 3). Em relação à Matriz tipo 3, a média da FO foi de 6384,80, ficando muito próxima da média encontrada com o AG, e conseqüentemente próxima da média encontrada em Azevedo (2010).

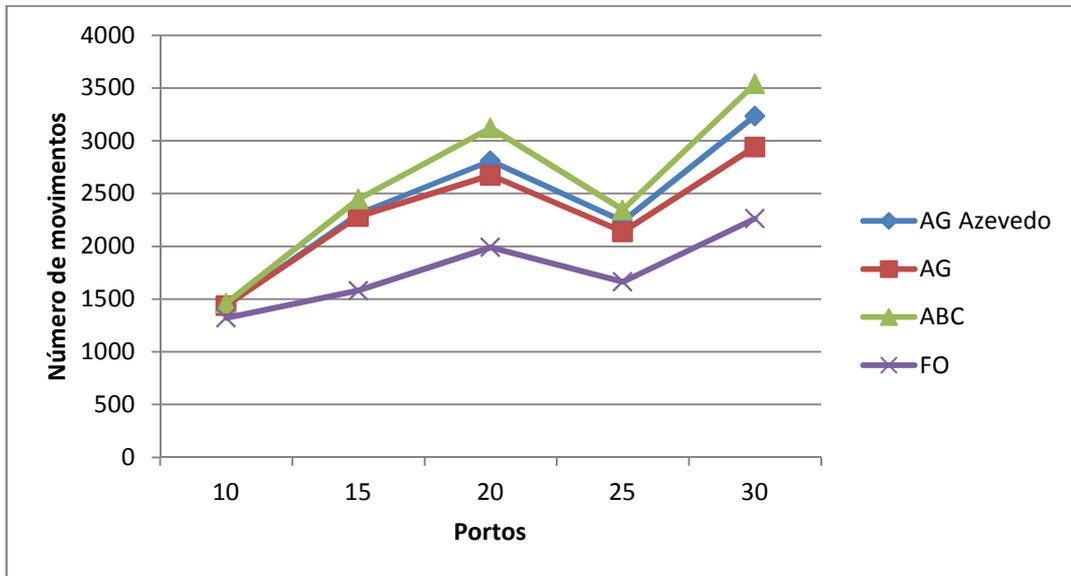
Em relação ao tempo gasto na realização das simulações, a seguir são apresentadas as médias para cada percurso. Para o percurso com 10 portos é gasto em média 1 a 1,5 horas, para 15 portos é gasto 1,5 a 2 horas, no percurso com 20 portos o tempo é de 2,5 a 2,7 horas, para 25 portos de 2,85 a 3 horas e para o percurso com 30 portos é gasto de 3,45 a 4 horas.

Como se pode notar, os valores médios de FO encontrados pelo algoritmo ABC foram piores que os encontrados por Azevedo (2010). Porém, o desvio-padrão destas soluções foi alto, ocasionado pela elevada diferença entre as soluções obtidas a cada execução do algoritmo. Mesmo encontrando resultados idênticos ou de pior qualidade que aqueles apresentados em (Azevedo, 2010), o algoritmo despendeu um tempo demasiado para tais experimentos. O tempo computacional médio foi de 2,5 horas, enquanto que o AG resolve o problema em um tempo médio de 3 minutos.

## 5.5 Comparativo entre Algoritmos

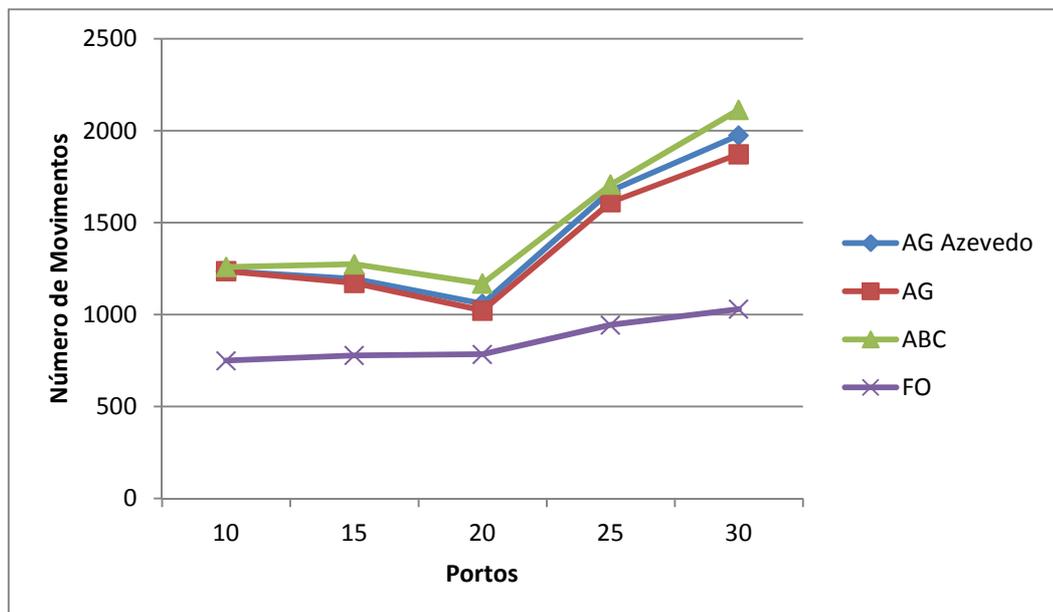
Nesta seção são apresentadas comparações entre os resultados obtidos com o emprego dos algoritmos AG de Azevedo (2010), AG proposto e ABC, com a finalidade de realizar uma análise complementar aos resultados apresentados na seção anterior. O gráfico da Figura 26 se refere aos valores médios de FO obtidos quando considerada a Matriz de Transporte tipo 1. As Figuras 27 e 28 referem-se à comparação quando assumidas as Matrizes de Transporte tipo 2 e tipo 3, respectivamente.

**Figura 26 - Métodos aplicados com 4 regras - Matriz tipo 1**



De acordo com a Figura 26, os algoritmos obtêm resultados muito próximos para os problemas com 10 e 25 portos, ao contrário dos problemas com 15, 20 e 30 portos, que apresentam uma maior discrepância nos seus valores médios. Conforme descrito na seção anterior, o desvio padrão foi elevado para algumas instâncias, coincidindo com aquelas que tiveram desempenho pior quando empregado o ABC. Os valores médios obtidos com os dois Algoritmos Genéticos comparados são idênticos quando a rota é com 10 e 15 portos. A partir de 20 portos, os valores médios se tornam distintos e assim acontece proporcionalmente, quanto maior o número de portos maior a diferença entre os resultados. Desta maneira, percebe-se que, mesmo para um percurso longo, o AG deste estudo teve desempenho adequado.

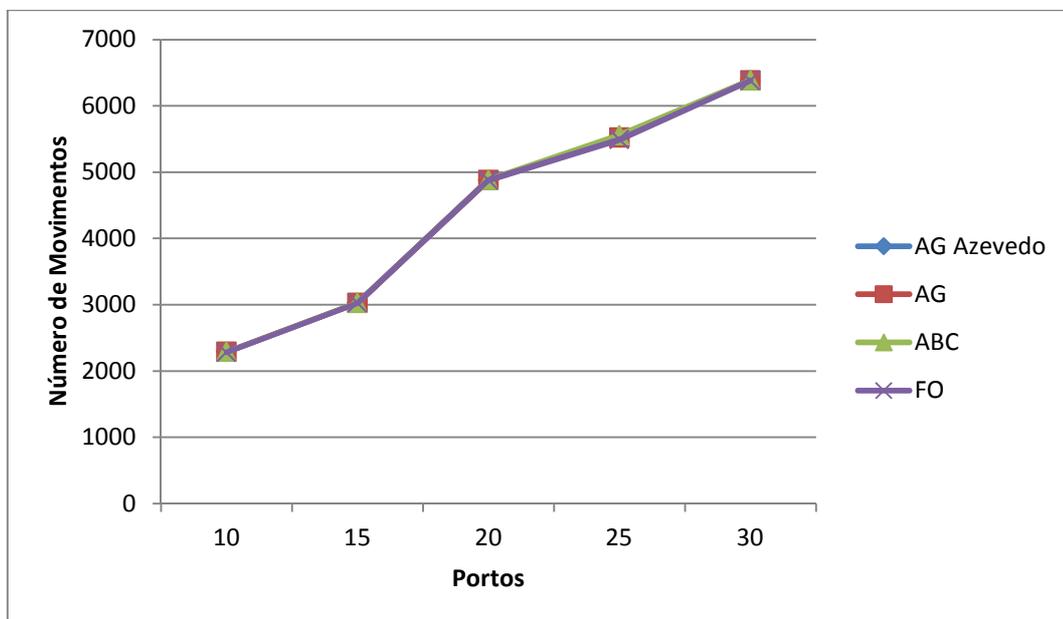
**Figura 27 - Métodos aplicados com 4 regras - Matriz tipo 2**



Na Figura 27, são representados os resultados referentes à Matriz tipo 2. Em análise, os valores médios obtidos com cada algoritmo estão mais próximos do que verificado para o caso da Matriz tipo 1. Isso é verificado devido à dificuldade do tipo da Matriz de Transporte. Em relação ao ABC, novamente os resultados relativos aos casos com 15, 20 e 30 portos estão em posição mais afastada se comparados com os demais. Para este tipo de Matriz, o ABC teve pior desempenho que os outros algoritmos em avaliação. Em contrapartida, o melhor algoritmo em estudo foi o AG proposto.

Na Figura 28, são apresentados os resultados relativos às instâncias que tratam da Matriz tipo 3 de curta distância.

**Figura 28 - Métodos aplicados com 4 regras - Matriz tipo 3**



Nota-se que as médias estão próximas e há uma sobreposição na representação dos resultados. Diferente dos resultados obtidos com uso das outras matrizes, esta é de mais fácil resolução por ser de curta distância, isto é, os contêineres permanecem a bordo um período curto. Apesar disso, o tempo computacional de simulação quando considerada esta matriz, é mais elevado que os demais casos, sendo observada a relação de quanto maior o percurso ou o número de portos, maior será o tempo computacional de simulação.

## 6 EXPERIMENTOS

Com a realização dos experimentos computacionais, pretende-se identificar a melhor sintonia dos parâmetros utilizados nos algoritmos empregados. Cada operador ou parâmetro tem uma finalidade diferente e, através da sua configuração adequada, o algoritmo poderá fornecer melhores resultados.

Neste sentido, o procedimento a ser realizado consiste na definição de cenários de teste. Os cenários que serão propostos irão variar o valor de parâmetros empregados nos operadores, taxas de probabilidade de utilização dos operadores e variações de operadores de cruzamento e mutação ou estruturas de vizinhança. Desta forma, o emprego dos cenários possibilita uma análise mais amplificada, podendo identificar a melhor configuração de cada algoritmo, e com isso priorizá-la na obtenção de melhores resultados.

Para o Algoritmo Genético, foram criados quatro cenários diferentes, conforme apresentado na Tabela 6:

**Tabela 6 - Cenários do AG**

	Cruzamento Uniforme	Cruzamento Um Ponto de Corte
Mutação Uniforme	Cenário 1	Cenário 2
Mutação Adaptativa	Cenário 3	Cenário 4

Para os Cenários 1 e 2, são propostas variações da probabilidade de cruzamento, assumindo-se os valores 100%, 90% e 80%, e da probabilidade de mutação, assumindo-se os valores 5%, 10% e 15%. Já para os Cenários 3 e 4, somente foi possível variar a probabilidade de cruzamento, assumindo-se os valores 100%, 90% e 80%, uma vez que a Mutação escolhida para tais cenários é a Mutação Adaptativa, conforme definida na seção 4.2.2.4.

Para o algoritmo de Enxame de Abelhas, foram criados dois cenários estabelecendo variações na quantidade de fontes de alimento exploradas e na aplicação de dois tipos diferentes de estrutura de vizinhança, conforme tabela a seguir:

**Tabela 7 - Cenários do ABC**

	Vizinhança Inversão	Vizinhança Troca
Fonte de Alimento	Cenário 1	Cenário 2

No Cenário 1, a estrutura de vizinhança por inversão é aplicada e varia-se a quantidade de fontes de alimento, assumindo-se os valores 50 e 100. Já no Cenário 2, a estrutura de

vizinhança utilizada é a por troca e as fontes de alimento também foram assumidas com valores 50 e 100.

Os resultados dos testes serão analisados, nesta etapa, através da média, desvio padrão e erro relativo em relação à solução ótima (GAP), com base em um conjunto de replicações do mesmo algoritmo. Além da análise dos experimentos de cada algoritmo, serão feitas comparações entre cenários, quando empregado o mesmo algoritmo, e entre os próprios algoritmos. Espera-se, através da realização destes experimentos, ser possível identificar a melhor configuração de operadores e parâmetros de cada algoritmo, que será então utilizada na obtenção de melhores soluções.

## **6.1 Características complementares dos experimentos**

As simulações foram realizadas empregando-se os códigos (AG e ABC) desenvolvidos através da ferramenta Matlab, e executados em computadores com processador Dual Core 2.00GHz, memória RAM de 3GB e Sistema Operacional Windows 7, Core 2 Duo 2.00GHz, memória RAM 3GB e Sistema Operacional Windows 7 e AMD Sempron 3,6 2.00GHz, memória RAM 2,5GB e Sistema Operacional Windows 7.

Para a realização dos experimentos, foi admitido o tamanho da população de 50 indivíduos, com critério de parada definido pelo número máximo de 1000 gerações. Para cada problema teste foram efetuadas 30 execuções. O tempo gasto computacionalmente não foi considerado como parâmetro de avaliação, pois o algoritmo foi simulado em diferentes máquinas.

### **6.1.1 Experimentos com o Algoritmo Genético**

Os experimentos realizados utilizaram as regras propostas no trabalho de Azevedo (2010) e em conjunto com as novas regras criadas neste estudo. Em resumo, são seis regras de carregamento e três regras de descarregamento de contêineres, totalizando dezoito combinações de regras conforme apresentado na Tabela 1 do Capítulo 4.

Na apresentação dos resultados obtidos com as simulações, são utilizadas tabelas para cada cenário de teste de acordo com a quantidade de portos e assumindo os três tipos de Matriz de Transporte. Nas tabelas, são mostrados os resultados médios e seus desvios-padrão, após 30 execuções do programa.

Os resultados médios apresentados nas tabelas são representados pelo valor da FO que corresponde ao número de movimentos de contêineres ao longo de todo o percurso, ou seja, para um problema teste com 10 portos, a FO é a soma dos movimentos realizados até se chegar ao último porto. O desvio-padrão  $\sigma$ , também exposto nas Tabelas de resultado, é calculado com base na média obtida dos 30 valores de FO obtidos.

Além de calcular o número de movimentos, o código implementado mostra, ao final da sua execução, a sequência de combinações de regras que devem ser empregadas com o intuito de obter o número de movimentos calculados. Vale lembrar que, para se realizar o planejamento do carregamento do navio, o porto onde se inicia os carregamentos deve conter um plano de carga com a quantidade de contêineres a serem carregados em cada porto da rota e seus respectivos destinos, aqui representado pela Matriz de Transporte. Sendo assim, é possível, ainda no primeiro porto, fazer o planejamento das regras que serão empregadas em cada porto, minimizando os custos e as movimentações desnecessárias. Nas tabelas referentes aos Cenários 1 e 2, é possível verificar as variações de probabilidade de cruzamento e probabilidade de mutação. Já nas tabelas referentes aos Cenários 3 e 4, não existem variações de probabilidade de mutação, pois a mutação adotada é adaptativa.

#### *6.1.1.1 Cenário 1*

Neste cenário de teste, são apresentados os resultados médios obtidos nas simulações realizadas para o problema assumindo o percurso no intervalo de 10 até 30 portos, representados em cinco tabelas. Os operadores escolhidos para tal cenário são o cruzamento uniforme e a mutação uniforme, variando suas probabilidades de aplicação conforme especificado anteriormente.

Para cada Tabela, são destacados os melhores resultados médios de cada tipo de Matriz de Transporte (1, 2 e 3), a fim de identificar as taxas de cruzamento e mutação que forneceram os melhores resultados. Na análise, busca-se comparar os resultados obtidos quando empregadas 4 combinações de regras, que foram mostrados no capítulo de validação, com os resultados obtidos quando empregadas 18 combinações de regras. Esta comparação tem como objetivo avaliar o desempenho das novas regras.

Na Tabela 8, são apresentados os resultados dos experimentos para o problema com 10 portos. Para a Matriz tipo 1, de distância mista, destaca-se o resultado médio de 1335,03 movimentos com desvio-padrão de 2,09, obtido quando assumida a probabilidade de cruzamento de 90% e probabilidade de mutação de 15%, correspondendo à melhor média encontrada. O menor valor encontrado entre as simulações foi de 1332,00 movimentos, que está muito próximo ao resultado ótimo esperado de 1322,00 movimentos. Se estes resultados forem comparados com os resultados apresentados em Azevedo (2010) e aqueles obtidos quando aplicado o AG com 4 regras, percebe-se que para este caso foram obtidos, em ambos os algoritmos, um valor médio de 1438,00 movimentos e desvio-padrão zero. Isto demonstra que, com o uso de apenas 4 regras, o algoritmo ficou preso em um mínimo local, não conseguindo alcançar o ótimo ou se aproximar dele. Logo, a aplicação de um maior número de regras possibilitou uma variação maior nas configurações do plano de carga do navio e uma melhora significativa no resultado.

Para a Matriz tipo 2 de longa distância, todos os testes realizados com as diferentes taxas de mutação e de cruzamento alcançaram como resultado final o mesmo número de movimentos de 760,00 e com desvio padrão nulo, exceto para os testes com probabilidade de

mutação de 5%, que por sua vez forneceram resultados muito próximos aos resultados encontrados quando assumidas as demais probabilidades de mutação. Para este problema, o resultado ótimo é de 750,00 movimentos e a solução apresentada se encontra muito próxima, porém o desvio-padrão nulo indica que o algoritmo pode ter ficado preso em um mínimo local, não conseguindo alcançar o mínimo global. Desta forma, para resolver este problema é necessário investir em uma estratégia de diversificação, por exemplo, outro operador de mutação. Ao comparar o valor médio da FO encontrado por Azevedo (2010), de 1236,00 movimentos, com os 760,00 movimentos obtidos através da técnica proposta, uma melhora significativa nos resultados é verificada.

**Tabela 8 - Cenário 1- AG 10 Portos**

<b>Problema com 10 Portos</b>					
<b>Probabilidade de Mutação</b>	<b>Tipo da Matriz</b>		<b>Probabilidade de Cruzamento</b>		
			<b>100</b>	<b>90</b>	<b>80</b>
<b>5</b>	<b>1</b>	<b>FO</b>	1337,83	1336,06	1336,90
		<b><math>\sigma</math></b>	3,45	1,92	2,07
	<b>2</b>	<b>FO</b>	760,23	761,00	760,6
		<b><math>\sigma</math></b>	1,27	2,59	2,29
	<b>3</b>	<b>FO</b>	2282,00	2282,00	2282,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00
<b>10</b>	<b>1</b>	<b>FO</b>	1336,00	1335,86	1337,36
		<b><math>\sigma</math></b>	1,41	2,66	2,61
	<b>2</b>	<b>FO</b>	760,00	760,00	760,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00
	<b>3</b>	<b>FO</b>	2282,00	2282,00	2282,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00
<b>15</b>	<b>1</b>	<b>FO</b>	1335,90	1335,03	1335,56
		<b><math>\sigma</math></b>	1,64	2,09	1,13
	<b>2</b>	<b>FO</b>	760,00	760,00	760,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00
	<b>3</b>	<b>FO</b>	2282,00	2282,00	2282,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00

Quando examinados os resultados obtidos para o caso da Matriz tipo 3 de curta distância, percebe-se uma homogeneidade nos resultados, mesmo sendo testada com diversas

taxas de probabilidade, tanto de cruzamento quanto de mutação. Através de uma análise da evolução das soluções, quando da execução do algoritmo, foi possível perceber uma rápida convergência do processo de solução para o valor esperado.

Como informação adicional, o tempo computacional para gerar cada solução após a execução de 1000 gerações, variou de 1 a 2 minutos. Na Tabela 9, são expostos os resultados obtidos dos experimentos realizados para o problema com 15 portos.

**Tabela 9 - Cenário 1– AG 15 Portos**

<b>Problema com 15 Portos</b>					
<b>Probabilidade de Mutação</b>	<b>Tipo da Matriz</b>		<b>Probabilidade de Cruzamento</b>		
			<b>100</b>	<b>90</b>	<b>80</b>
<b>5</b>	<b>1</b>	<b>FO</b>	1616,26	1617,16	1618,4
		<b><math>\sigma</math></b>	4,84	5,24	4,35
	<b>2</b>	<b>FO</b>	790,26	790,86	791,2
		<b><math>\sigma</math></b>	2,28	2,48	2,28
	<b>3</b>	<b>FO</b>	3024,00	3024,00	3024,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00
<b>10</b>	<b>1</b>	<b>FO</b>	1616,33	1615,66	1618,53
		<b><math>\sigma</math></b>	3,44	3,12	6,6
	<b>2</b>	<b>FO</b>	789,66	789,86	789,73
		<b><math>\sigma</math></b>	2,00	2,44	2,04
	<b>3</b>	<b>FO</b>	3024,00	3024,00	3024,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00
<b>15</b>	<b>1</b>	<b>FO</b>	1615,6	1615,4	1618,03
		<b><math>\sigma</math></b>	3,85	5,28	4,75
	<b>2</b>	<b>FO</b>	789,66	789,16	789,56
		<b><math>\sigma</math></b>	2,38	2,26	1,85
	<b>3</b>	<b>FO</b>	3024,00	3024,00	3024,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00

Na Tabela 9, são expostas as soluções do problema com 15 Portos que fazem parte do mesmo cenário. Observa-se que para as Matrizes do tipo 1 e 2, os melhores resultados puderam ser encontrados quando assumida a probabilidade de cruzamento de 90% e a probabilidade de mutação de 15%, com resultados 1615,40 e 789,16 respectivamente. Embora os desvios-padrão destes resultados sejam de 5,28 e 2,26, maiores que os demais, ainda podem ser considerados valores de desvio-padrão baixos. Se comparados os resultados médios encontrados com o emprego de 18 combinações de regras com os resultados do

mesmo algoritmo com o emprego de apenas 4 combinações, e ainda com os resultados de Azevedo (2010), que forneceram os valores 2308,00 e 1194,40 movimentos respectivamente, pode-se concluir de forma clara que os valores das soluções obtidas com o emprego de 18 combinações são menores e por consequência melhores.

Em relação à Matriz tipo 3, o algoritmo alcançou a solução ótima esperada, tendo fornecido um valor médio de 3024,00 movimentos e desvio-padrão nulo em todos os casos. Logo, pode-se, novamente, notar uma melhora dos resultados obtidos com o AG proposto com 18 combinações de regras se comparado ao algoritmo de Azevedo (2010), que obteve 3028,00 movimentos para o mesmo problema. Vale observar que o tempo computacional do algoritmo com 18 combinações é superior ao tempo computacional verificado para o AG quando assumidas apenas 4 combinações de regras. Em contrapartida, os resultados são expressivamente melhores. O tempo médio gasto para realizar os testes apresentados na Tabela 9 apresentou-se entre 2 a 3 minutos.

**Tabela 10 - Cenário 1– AG 20 Portos**

<b>Problema com 20 Portos</b>					
<b>Probabilidade de Mutação</b>	<b>Tipo da Matriz</b>		<b>Probabilidade de Cruzamento</b>		
			<b>100</b>	<b>90</b>	<b>80</b>
<b>5</b>	<b>1</b>	<b>FO</b>	2041,16	2040,06	2047,93
		<b><math>\sigma</math></b>	8,67	8,99	15,73
	<b>2</b>	<b>FO</b>	817,03	815,9	819,6
		<b><math>\sigma</math></b>	9,44	9,67	9,12
	<b>3</b>	<b>FO</b>	4880,00	4880,00	4880,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00
<b>10</b>	<b>1</b>	<b>FO</b>	2037,93	2041,26	2041,6
		<b><math>\sigma</math></b>	8,41	10,70	11,31
	<b>2</b>	<b>FO</b>	815,83	817,23	819,63
		<b><math>\sigma</math></b>	10,09	10,3	9,04
	<b>3</b>	<b>FO</b>	4880,00	4880,00	4880,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00
<b>15</b>	<b>1</b>	<b>FO</b>	2037,7	2040,76	2040,3
		<b><math>\sigma</math></b>	10,07	8,46	10,74
	<b>2</b>	<b>FO</b>	817,13	813,00	815,6
		<b><math>\sigma</math></b>	8,53	11,1	8,83
	<b>3</b>	<b>FO</b>	4880,00	4880,00	4880,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00

Na Tabela 10, os resultados mostrados referem-se ao problema com 20 portos. Observando os resultados obtidos com as simulações, pode-se verificar que os melhores resultados, quando empregadas as Matrizes tipo 1 e 2, foram obtidos quando utilizada uma taxa de mutação de 15%. O emprego desta mesma taxa de mutação gerou bons resultados nos experimentos cujos resultados foram apresentados nas Tabelas 7 e 8.

Em relação aos experimentos quando assumida a Matriz tipo 1, o melhor resultado obtido foi 2037,70 com desvio-padrão de 10,07 e assumindo o valor 100% para a taxa de cruzamento. Por outro lado, quando assumida Matriz tipo 2, o melhor resultado foi de 813,00 movimentos com desvio padrão de 11,1 para uma probabilidade de cruzamento de 90%. Aliado a estes resultados, se também avaliadas as taxas de cruzamento que forneceram os melhores resultados nos problemas resolvidos anteriormente, percebe-se que as taxas de cruzamento de 100% e 90% apresentaram os melhores resultados, ou seja, os bons resultados estão concentrados nos casos em que as taxas de cruzamento são elevadas.

Com relação aos resultados obtidos considerando a Matriz tipo 3, foi possível observar que apresentam o mesmo desempenho verificado nas tabelas anteriores. Logo, novamente foi possível alcançar o resultado ótimo de 4880,00 movimentos com desvio padrão nulo. O tempo computacional verificado nas simulações realizadas variou entre 3,5 e 4 minutos.

Para o problema com 25 Portos, nas simulações assumindo a Matriz tipo 1, verificou-se um bom resultado médio no valor de 1695,06 movimentos quando assumida a taxa de cruzamento no valor de 100% e a taxa de mutação no valor de 15%. O valor obtido para o desvio padrão foi de 8,49, sendo o menor valor de desvio padrão encontrado quando empregada a Matriz tipo 1 nestas simulações. Ao serem comparados estes resultados médios com os resultados obtidos empregando-se 4 combinações de regras, onde foram obtidas médias no valor de 2237,20 (Azevedo, 2010) e 2136,86 movimentos (AG proposto), nota-se que o algoritmo com 18 combinações de regras é o que mais se aproxima do resultado ótimo de 1664,00 movimentos.

Para o caso da Matriz tipo 2, os experimentos que forneceram o menor valor médio de FO assumiu 80% para a taxa de cruzamento, sendo mantido os 15% de taxa de mutação. Em geral, o emprego de uma taxa de 80% para o cruzamento somente forneceu bons resultados quando a taxa de mutação foi fixada em 15%, pois quando a taxa de mutação foi de 5% e 10%, as soluções encontradas foram piores.

Para o caso da Matriz tipo 3, os resultados são homogêneos em todos os casos, alcançando o resultado ótimo de 5492,00 e desvio padrão nulo. O tempo de simulação para este problema foi em média de 4 a 5 minutos para cada execução de 1000 gerações.

Tabela 11 - Cenário 1- AG 25 Portos

Problema com 25 Portos					
Probabilidade de Mutação	Tipo da Matriz		Probabilidade de Cruzamento		
			100	90	80
5	1	FO	1700,4	1699	1703,95
		$\sigma$	12,17	11,95	14,61
	2	FO	987,6	991,36	989,86
		$\sigma$	8,47	9,24	10,6
	3	FO	5492,00	5492,00	5492,00
		$\sigma$	0,00	0,00	0,00
10	1	FO	1696,96	1700	1700,2
		$\sigma$	8,88	8,71	10,29
	2	FO	987,86	988,16	988,7
		$\sigma$	9,68	9,56	9,48
	3	FO	5492,00	5492,00	5492,00
		$\sigma$	0,00	0,00	0,00
15	1	FO	1695,06	1696,6	1696,26
		$\sigma$	8,49	9,55	10,67
	2	FO	987,73	990,73	985,76
		$\sigma$	8,97	9,44	8,86
	3	FO	5492,00	5492,00	5492,00
		$\sigma$	0,00	0,00	0,00

Para o problema com 30 portos e assumindo a Matriz tipo 1, o melhor resultado de 2311,43 movimentos e desvio padrão de 14,2 foi obtido com 90% de taxa de cruzamento e 15% de taxa de mutação. Este resultado é melhor que os resultados quando empregadas 4 combinações de regras, quando foram obtidos os valores médios de 3235,20 (Azevedo, 2010) e 2942,00 movimentos (AG proposto). Em relação ao desempenho do operador de cruzamento, quando empregada a taxa de 100%, os resultados são melhores do que quando empregada a taxa de 90%, independentemente se a taxa de mutação for 5% ou 10%. Já o uso de uma taxa de cruzamento de 80% só se mostrou melhor do que a taxa de 90% quando a mutação foi de 5%.

Assumindo o emprego da Matriz tipo 2, verificou-se um melhor desempenho quando empregada uma taxa de cruzamento de 100% e uma taxa de mutação de 10%. Com esta configuração, foi obtido um valor médio de 1090,80 movimentos de contêineres, que novamente é de qualidade superior às médias encontradas quando utilizadas apenas 4 combinações de regras, quando os resultados obtidos foram de 1975,40 movimentos, segundo

(Azevedo, 2010) e 1872,00 movimentos, pelo AG proposto. Se observado o desempenho quando empregadas as taxas de cruzamento de 90% e 80%, percebe-se que os resultados médios pioram à medida que a taxa de cruzamento diminui.

**Tabela 12 - Cenário 1 – AG 30 Portos**

<b>Problema com 30 Portos</b>					
<b>Probabilidade de Mutação</b>	<b>Tipo da Matriz</b>		<b>Probabilidade de Cruzamento</b>		
			<b>100</b>	<b>90</b>	<b>80</b>
<b>5</b>	<b>1</b>	<b>FO</b>	2311,86	2323,66	2318,86
		<b><math>\sigma</math></b>	13,26	14,28	12,71
	<b>2</b>	<b>FO</b>	1096,9	1096,96	1098,7
		<b><math>\sigma</math></b>	13,94	12,72	12,52
	<b>3</b>	<b>FO</b>	6380,00	6380,00	6380,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00
<b>10</b>	<b>1</b>	<b>FO</b>	2312,76	2314,5	2317,2
		<b><math>\sigma</math></b>	11,05	13,62	15,27
	<b>2</b>	<b>FO</b>	1090,8	1097,8	1098,9
		<b><math>\sigma</math></b>	11,91	14,54	14,88
	<b>3</b>	<b>FO</b>	6380,00	6380,00	6380,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00
<b>15</b>	<b>1</b>	<b>FO</b>	2313,43	2311,43	2317,63
		<b><math>\sigma</math></b>	10,02	14,2	14,02
	<b>2</b>	<b>FO</b>	1092,73	1094,5	1100,7
		<b><math>\sigma</math></b>	12,81	13,92	17,79
	<b>3</b>	<b>FO</b>	6380,00	6380,00	6380,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00

Com relação aos resultados obtidos com a Matriz tipo 3, foram alcançadas as soluções ótimas no valor de 6380,00 movimentos em todas as configurações. Para este problema, tempo computacional necessário para gerar cada solução é entre 5 e 6,20 minutos.

Analisando as simulações realizadas no Cenário 1, nota-se uma concentração das melhores médias de resultados quando adotada uma taxa de mutação é 15%. Já em relação à melhor taxa de cruzamento, há variações entre 100% e 90% tendo sido verificado apenas um caso com melhor valor médio quando a taxa de cruzamento foi de 80%. Na maioria dos casos, os melhores valores médios foram obtidos com 90% de probabilidade de cruzamento.

Com isso, para o Cenário 1, o emprego de maiores valores para as taxas de cruzamento e de mutação produziu soluções de qualidade superior. Por esta razão, foi

necessário realizar alguns experimentos adicionais assumindo uma probabilidade maior para a mutação, sendo fixada em 20%, com o intuito de analisar se o algoritmo continua obtendo melhora nas soluções. Após a realização de alguns experimentos, notou-se que, com a probabilidade de 20% os resultados obtidos apresentaram um leve declínio de qualidade, ou seja, o algoritmo retornou um valor médio maior da FO, isto é, um maior número de movimentos de contêineres. Sendo assim, admite-se a probabilidade de 15% como o melhor valor neste cenário.

Quando realizadas as simulações assumindo a Matriz tipo 3, percebeu-se que se trata de um caso especial e diferente dos outros dois tipos de matrizes. Foi possível alcançar o ótimo global rapidamente em todos os casos simulados. Este tipo de matriz baseia-se no carregamento e descarregamento de contêineres com curta permanência a bordo. Por este motivo, acredita-se que o planejamento seja mais simples sem a necessidade do emprego de uma quantidade grande de regras.

O tempo computacional de cada experimento foi apresentado apenas para conhecimento, não sendo considerado como parâmetro de análise, levando em conta que foram utilizadas diferentes máquinas na etapa de simulação.

#### *6.1.1.2 Cenário 2*

Neste cenário, são apresentados os resultados médios obtidos a partir das simulações realizadas para o problema, assumindo percursos no intervalo de 10 até 30 portos, representados em cinco tabelas. Em cada uma das tabelas são destacadas as médias de menor valor, por ser um problema de minimização. Os operadores escolhidos para tal cenário foram o cruzamento um ponto de corte e a mutação uniforme, conforme especificado anteriormente.

Na análise dos experimentos deste cenário, serão apresentadas comparações entre os valores assumidos para as taxas de cruzamento e mutação e ainda serão comparados os resultados obtidos com o algoritmo AG quando aplicadas 4 combinações de regras.

Na tabela 13 são apresentados os valores médios da FO para o problema com 10 portos e seus respectivos valores de desvio padrão.

Quando assumida a Matriz tipo 1, foram verificados bons resultados com taxa de cruzamento de 90% e taxa de mutação de 15%. Se efetuada uma comparação com os resultados obtidos no Cenário 1, para a mesma instância obteve-se como resultado o valor médio de 1335,03 movimentos, o que corresponde a um menor número de movimentos do que o encontrado no Cenário 2, onde obteve-se um valor médio de 1335,87 movimentos.

Em ambos os Cenários de teste, para a Matriz tipo 1, as menores médias foram obtidas quando empregados os mesmos valores de probabilidade de cruzamento e probabilidade de mutação, diferenciadas apenas pelo operador de cruzamento. Logo, o operador de Cruzamento Uniforme mostrou-se mais eficiente.

Tabela 13 - Cenário 2- AG 10 Portos

Problema com 10 Portos - Cenário 2					
Probabilidade de Mutação	Tipo da Matriz		Probabilidade de Cruzamento		
			100	90	80
5	1	FO	1337,73	1337,73	1339,13
		$\sigma$	3,92	3,38	4,70
	2	FO	760,5	760,03	761,33
		$\sigma$	1,90	0,18	2,98
	3	FO	2282,00	2282,00	2282,00
		$\sigma$	0,00	0,00	0,00
10	1	FO	1337,4	1336,53	1337,21
		$\sigma$	3,17	2,19	3,81
	2	FO	760,5	760,00	760,23
		$\sigma$	1,77	0,00	1,27
	3	FO	2282,00	2282,00	2282,00
		$\sigma$	0,00	0,00	0,00
15	1	FO	1336,46	1335,87	1336,43
		$\sigma$	2,23	2,19	2,29
	2	FO	760,00	760,26	760,46
		$\sigma$	0,00	1,46	1,77
	3	FO	2282,00	2282,00	2282,00
		$\sigma$	0,00	0,00	0,00

Quando empregada a Matriz tipo 2, o menor resultado médio obtido é idêntico ao obtido no Cenário 1, porém no Cenário 2 o resultado ótimo de 760 movimentos foi encontrado apenas em 2 casos, diferente do Cenário 1 onde esse mesmo resultado foi encontrado em seis configurações. Já quando empregada a Matriz tipo 3, foi possível encontrar rapidamente o resultado ótimo, da mesma forma que no Cenário 1. O tempo computacional foi equivalente ao problema com 10 portos avaliado no Cenário 1.

Tabela 14 - Cenário 2- AG 15 Portos

Problema com 15 Portos – Cenário 2					
Probabilidade de Mutação	Tipo da Matriz		Probabilidade de Cruzamento		
			100	90	80
5	1	FO	1620,76	1621,73	1619,90
		$\sigma$	6,30	6,74	5,93
	2	FO	790,16	790,80	790,96
		$\sigma$	2,40	2,84	3,41
	3	FO	3024,00	3024,00	3024,00
		$\sigma$	0,00	0,00	0,00
10	1	FO	1617,43	1618,73	1617,20
		$\sigma$	5,36	5,86	5,22
	2	FO	789,10	790,76	790,03
		$\sigma$	1,76	2,87	2,41
	3	FO	3024,00	3024,00	3024,00
		$\sigma$	0,00	0,00	0,00
15	1	FO	1617,60	1617,53	1617,66
		$\sigma$	4,93	4,43	5,4
	2	FO	789,60	789,70	790,10
		$\sigma$	2,60	2,07	2,57
	3	FO	3024,00	3024,00	3024,00
		$\sigma$	0,00	0,00	0,00

Analisando a Tabela 14, quando assumida a Matriz tipo 1, a média da FO destacada foi de 1617,20 movimentos e desvio padrão de 5,22, tendo sido assumidas as probabilidades de cruzamento e mutação de 80% e 10% respectivamente. Comparando o mesmo problema, porém no Cenário 1, foi obtido um valor médio de 1615,40 movimentos com desvio padrão de 5,28, mostrando um desempenho melhor que no Cenário 2.

Para o caso da Matriz tipo 2, os valores médios da FO ficaram muito próximos, sobressaindo-se a média de 789,10 movimentos com 100% de taxa de cruzamento e 10% de taxa de mutação. Esta média mostrou-se ligeiramente melhor que a média encontrada no Cenário 1 de 789,16 movimentos.

Para o caso da Matriz tipo 3, encontrou-se, da mesma forma que no Cenário 1, o resultado desejado de 3024,00 movimentos. O tempo computacional necessário para se obter a solução deste problema foi idêntico ao mencionado no Cenário 1.

Tabela 15 - Cenário 2- AG 20 Portos

Problema com 20 Portos – Cenário 2					
Probabilidade de Mutação	Tipo da Matriz		Probabilidade de Cruzamento		
			100	90	80
5	1	FO	2049,36	2048,83	2050,23
		$\sigma$	11,98	16,62	22,46
	2	FO	820,16	819,36	821,33
		$\sigma$	1,34	9,03	9,20
	3	FO	4880,00	4880,00	4880,00
		$\sigma$	0,00	0,00	0,00
10	1	FO	2043,08	2041,97	2044,10
		$\sigma$	10,54	9,86	10,77
	2	FO	816,66	818,70	821,43
		$\sigma$	9,27	9,98	9,33
	3	FO	4880,00	4880,00	4880,00
		$\sigma$	0,00	0,00	0,00
15	1	FO	2046,26	2041,70	2042,90
		$\sigma$	9,91	11,98	9,38
	2	FO	816,50	817,43	818,00
		$\sigma$	10,04	9,30	8,43
	3	FO	4880,00	4880,00	4880,00
		$\sigma$	0,00	0,00	0,00

A partir dos resultados apresentados na Tabela 15, verificou-se que o melhor valor médio de FO, para o caso da Matriz tipo 1 foi de 2041,7 com desvio padrão de 11,98. Este resultado foi obtido assumindo-se taxas de 90% de cruzamento e 15% de mutação. Ao comparar as médias do mesmo problema no Cenário 1, se constata que no Cenário 1 foram obtidas soluções de melhor qualidade e de menor desvio padrão, com o valor médio de 2037,70 movimentos e desvio padrão de 10,07.

Para o caso da Matriz tipo 2, o resultado de 816,50 movimentos foi o menor valor médio encontrado para a FO. Ao confrontá-lo com o valor médio encontrado no Cenário 1, que foi de 813,00, constata-se uma piora nos resultados quando empregado o operador de cruzamento um ponto de corte, correspondente ao Cenário2.

Para o caso da Matriz tipo 3, tem-se uma homogeneidade nos resultados, encontrando de forma adequada a solução ótima da mesma forma que no Cenário 1. A média de tempo gasto para se obter a solução deste problema foi semelhante ao tempo gasto no Cenário 1.

**Tabela 16 - Cenário 2- AG 25 Portos**

<b>Problema com 25 Portos – Cenário 2</b>					
<b>Probabilidade de Mutação</b>	<b>Tipo da Matriz</b>		<b>Probabilidade de Cruzamento</b>		
			<b>100</b>	<b>90</b>	<b>80</b>
<b>5</b>	<b>1</b>	<b>FO</b>	1703,20	1705,03	1708,06
		<b><math>\sigma</math></b>	10,74	11,46	14,66
	<b>2</b>	<b>FO</b>	995,3	993,56	993,83
		<b><math>\sigma</math></b>	11,63	13,51	11,54
	<b>3</b>	<b>FO</b>	5492,00	5492,00	5492,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00
<b>10</b>	<b>1</b>	<b>FO</b>	1698,46	1697,53	1706,80
		<b><math>\sigma</math></b>	10,58	9,14	13,98
	<b>2</b>	<b>FO</b>	994,53	995,26	992,90
		<b><math>\sigma</math></b>	11,36	12,00	15,22
	<b>3</b>	<b>FO</b>	5492,00	5492,00	5492,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00
<b>15</b>	<b>1</b>	<b>FO</b>	1702,36	1702,10	1704,00
		<b><math>\sigma</math></b>	12,08	11,98	11,72
	<b>2</b>	<b>FO</b>	990,70	992,10	991,26
		<b><math>\sigma</math></b>	10,04	13,24	10,89
	<b>3</b>	<b>FO</b>	5492,00	5492,00	5492,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00

Conforme os dados apresentados na Tabela 16, os resultados obtidos para o caso da Matriz tipo 1 se mostraram muito semelhantes, destacando-se o valor médio de 1697,53 movimentos. Este resultado foi obtido quando assumido 90% de taxa de cruzamento e 10% de taxa de mutação, com desvio padrão de 9,14, sendo o menor valor obtido dentre todas as configurações. Recordando os valores médios encontrados para o mesmo problema no primeiro Cenário, obteve-se 1695,06 movimentos, verificando-se a sua qualidade superior frente ao Cenário 2.

Para o caso da Matriz tipo 2, a melhor solução encontrada foi de 990,70 movimentos, tendo sido obtida com 100% de taxa de cruzamento e 15% de taxa de mutação. Observa-se

que o valor de desvio padrão encontrado para esta configuração também foi menor que os demais resultados apresentados nesta tabela. Ao ser realizada uma comparação com o Cenário 1, nota-se que os resultados do cenário anterior foram mais promissores, tendo sido obtido o valor médio de 985,76 movimentos, o que equivale em média a 5 movimentos a menos em todo o percurso.

Para o caso da matriz tipo 3, percebe-se a uniformidade de seus valores médios, independente dos parâmetros aplicados, e até mesmo do tipo de operador de mutação ou cruzamento, se realizada uma comparação dos resultados obtidos nos Cenários 1 e 2. O tempo gasto, neste cenário, para se obter a solução do problema com 25 portos, é idêntico ao encontrado no Cenário 1.

**Tabela 17 - Cenário 2- AG 30 Portos**

<b>Problema com 30 Portos – Cenário 2</b>					
<b>Probabilidade de Mutação</b>	<b>Tipo da Matriz</b>		<b>Probabilidade de Cruzamento</b>		
			<b>100</b>	<b>90</b>	<b>80</b>
<b>5</b>	<b>1</b>	<b>FO</b>	2321,93	2336,23	2326,03
		<b><math>\sigma</math></b>	10,15	22,43	15,68
	<b>2</b>	<b>FO</b>	1101,96	1108,80	1108,20
		<b><math>\sigma</math></b>	14,57	14,41	18,55
	<b>3</b>	<b>FO</b>	6380,00	6380,00	6380,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00
<b>10</b>	<b>1</b>	<b>FO</b>	2322,46	2322,03	2325,83
		<b><math>\sigma</math></b>	16,62	16,80	12,77
	<b>2</b>	<b>FO</b>	1104,96	1104,20	1104,90
		<b><math>\sigma</math></b>	15,83	13,28	14,06
	<b>3</b>	<b>FO</b>	6380,00	6380,00	6380,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00
<b>15</b>	<b>1</b>	<b>FO</b>	2320,80	2318,77	2323,23
		<b><math>\sigma</math></b>	15,97	11,85	13,94
	<b>2</b>	<b>FO</b>	1100,96	1104,90	1101,76
		<b><math>\sigma</math></b>	13,69	22,62	13,27
	<b>3</b>	<b>FO</b>	6380,00	6380,00	6380,00
		<b><math>\sigma</math></b>	0,00	0,00	0,00

Analisando o último problema teste do Cenário 2, observa-se que os melhores resultados médios para os casos das Matrizes tipo 1 e 2 se concentram na configuração com taxa de mutação de 15%, e taxa de cruzamento de 90% e 100%, respectivamente. Comparando as soluções obtidas para os casos das Matrizes tipo 1 e 2 com as soluções para o mesmo problema no Cenário 1, destacam-se os melhores resultados obtidos no Cenário 1, com base nas médias de 2311,43 e 1090,80, respectivamente. Já para o caso da Matriz de Transporte tipo 3, igualmente ao Cenário 1, encontrou-se o resultado ótimo em todos os casos, e com desvio padrão nulo, ou seja, em nenhuma das 30 execuções, de cada configuração deste problema teste, foi encontrado outro valor.

Após detalhar as simulações dos Cenários 1 e 2, que diferenciam-se pelo operador de cruzamento, é possível fazer uma análise a respeito dos parâmetros adotados nestes cenários. Quando a probabilidade de mutação é de 5%, com taxas de cruzamento entre 80% e 100%, somente são encontrados resultados médios mínimos para o caso da Matriz tipo 3, lembrando que em todas as simulações realizadas para este tipo de Matriz o resultado ótimo foi encontrado com desvio padrão nulo.

Considerando que o emprego de uma taxa de mutação de 5% não gera os melhores resultados, deve-se realizar a análise para o caso da taxa de mutação de 10%. Foram identificados alguns casos que apresentam este parâmetro como sendo o preferível. Se desconsideradas as simulações com resultados idênticos, a taxa de mutação de 10% se destaca no Cenário 1 para o caso com 30 portos e Matriz tipo 1, Cenário 2 para o caso com 15 Portos e Matriz tipos 1 e 2, e para o caso com 25 Portos e Matriz tipo 1. Desta forma, com o emprego da taxa de mutação de 15%, foi possível encontrar soluções melhores na maioria dos casos estudados até o momento em ambos os Cenário de teste.

### *6.1.1.3 Cenário 3*

Neste cenário, são apresentados os resultados médios das simulações realizadas para o intervalo de 10 a 30 portos. Conforme explicado anteriormente, os operadores escolhidos para tal cenário são o cruzamento uniforme e a mutação adaptativa. A Mutação Adaptativa consiste na adequação da probabilidade de mutação conforme a qualidade do indivíduo, por este motivo não possui valores de probabilidade fixos.

Observa-se na Tabela 18 que, para o caso da Matriz tipo 1 obteve-se a menor média de 1335,96 movimentos com resultados de 1,35 de desvio padrão, sendo empregado o valor de 80% para a taxa de cruzamento. Este valor médio não é melhor que o menor valor médio obtido no Cenário 1, de 1335,03 movimentos, e também em relação ao Cenário 2, onde foi obtido o valor de 1335,87 movimentos. Este cenário diferencia-se dos anteriores pelo operador de mutação, que para este problema mostrou-se menos eficiente que o operador de mutação uniforme aplicada nos Cenários 1 e 2.

Para os casos das Matrizes tipo 2 e 3, os valores de 760,00 e 2282,00 movimentos foram os mesmos em todos os casos avaliados e idênticos aos menores valores médios

encontrados nos Cenários 1 e 2. O tempo gasto com as simulações para este problema teste foram, em média, de 1 a 2 minutos.

**Tabela 18 - Cenário 3- AG 10 Portos**

<b>Problema com 10 Portos – Cenário 3</b>					
<b>Mutação Adaptativa</b>	<b>Tipo da Matriz</b>		<b>Probabilidade de Cruzamento</b>		
			<b>100</b>	<b>90</b>	<b>80</b>
	<b>1</b>	<b>FO</b>		1336,03	1336,33
	<b><math>\sigma</math></b>		1,18	1,49	1,35
<b>2</b>	<b>FO</b>		760,00	760,00	760,00
	<b><math>\sigma</math></b>		0,00	0,00	0,00
<b>3</b>	<b>FO</b>		2282,00	2282,00	2282,00
	<b><math>\sigma</math></b>		0,00	0,00	0,00

Para o problema teste com 15 portos, conforme a Tabela 19, a melhor solução para o caso da Matriz tipo 1 é 1620,23 movimentos com desvio padrão 4,62, tendo sido assumido o valor de 80% para a taxa de cruzamento. Verifica-se que este valor médio é de pior qualidade se comparado com as médias encontradas nos Cenário 1 e 2.

Para o caso da Matriz tipo 2, as soluções são muito semelhantes, mas a melhor é de 791,63 movimentos com desvio padrão de 4,62, tendo sido obtida para uma probabilidade de cruzamento de 90%. Nos Cenários 1 e 2, alcançaram-se melhores médias para o caso da Matriz tipo 2. Já para o caso da Matriz tipo 3, os resultados são idênticos aos encontrados nos Cenários anteriores, tendo sido obtida a solução esperada. O tempo gasto para simular o problema com 15 portos foi de 2 a 3 minutos.

**Tabela 19 - Cenário 3- AG 15 Portos**

<b>Problema com 15 Portos – Cenário 3</b>					
<b>Mutação Adaptativa</b>	<b>Tipo da Matriz</b>		<b>Probabilidade de Cruzamento</b>		
			<b>100</b>	<b>90</b>	<b>80</b>
	<b>1</b>	<b>FO</b>		1624,5	1623,36
	<b><math>\sigma</math></b>		4,96	5,32	4,62
<b>2</b>	<b>FO</b>		791,83	791,63	791,90
	<b><math>\sigma</math></b>		2,53	2,85	2,52
<b>3</b>	<b>FO</b>		3024,00	3024,00	3024,00
	<b><math>\sigma</math></b>		0,00	0,00	0,00

Observa-se na Tabela 20 que, para a taxa de 80% de cruzamento, estão destacados os melhores valores médios obtidos para os três tipos de Matrizes de Transporte, significando uma melhora nos resultados, ou seja, à medida que se diminui a taxa de cruzamento, melhores resultados são obtidos. Para o caso da Matriz 1, obteve-se 2085,36 movimentos e desvio padrão de 14,29. Para o caso da Matriz 2, foi possível atingir a média de 832,56 movimentos e desvio padrão de 6,91. Porém, para ambos os casos, não foi possível encontrar médias melhores que as encontradas nos Cenários anteriores. Em relação ao caso da Matriz tipo 3, a média de 4880,00 movimentos foi atingida em todos os casos com desvio padrão nulo. O tempo computacional deste problema foi, em média, 3,5 a 4 minutos.

**Tabela 20 - Cenário 3- AG 20 Portos**

<b>Problema com 20 Portos – Cenário 3</b>					
<b>Mutação Adaptativa</b>	<b>Tipo da Matriz</b>		<b>Probabilidade de Cruzamento</b>		
			<b>100</b>	<b>90</b>	<b>80</b>
	<b>1</b>	<b>FO</b>	2098,23	2094,80	2085,36
	<b><math>\sigma</math></b>	11,57	16,02	14,29	
<b>2</b>	<b>FO</b>	837,33	835,56	832,56	
	<b><math>\sigma</math></b>	7,67	7,83	6,91	
<b>3</b>	<b>FO</b>	4880,00	4880,00	4880,00	
	<b><math>\sigma</math></b>	0,00	0,00	0,00	

Analisando a Tabela 21, nota-se uma superioridade dos resultados quando empregado o valor de 80% para a taxa de cruzamento. Para os casos das Matrizes tipo 1 e 2, são alcançadas as médias de 1728,26 e 1005,30 movimentos. Apesar de serem bons resultados, ambas as médias não conseguem atingir os menores valores médios encontrados nos cenários anteriores.

**Tabela 21 - Cenário 3- AG 25 Portos**

<b>Problema com 25 Portos – Cenário 3</b>					
<b>Mutação Adaptativa</b>	<b>Tipo da Matriz</b>		<b>Probabilidade de Cruzamento</b>		
			<b>100</b>	<b>90</b>	<b>80</b>
	<b>1</b>	<b>FO</b>	1730,80	1731,23	1728,26
	<b><math>\sigma</math></b>	11,90	11,83	11,62	
<b>2</b>	<b>FO</b>	1008,36	1007,83	1005,30	
	<b><math>\sigma</math></b>	8,53	7,70	9,98	
<b>3</b>	<b>FO</b>	5492,00	5492,00	5492,00	
	<b><math>\sigma</math></b>	0,00	0,00	0,00	

Para o caso da Matriz tipo 3, o menor valor médio da FO é encontrado em 5492,00 movimentos, que representa o ótimo esperado. O tempo gasto para essas simulações é 4 a 5 minutos.

**Tabela 22 - Cenário 3- AG 30 Portos**

<b>Problema com 30 Portos – Cenário 3</b>					
<b>Mutação Adaptativa</b>	<b>Tipo da Matriz</b>		<b>Probabilidade de Cruzamento</b>		
			<b>100</b>	<b>90</b>	<b>80</b>
	<b>1</b>	<b>FO</b>	2393,06	2388,40	2385,76
	<b><math>\sigma</math></b>	14,68	18,76	13,19	
<b>2</b>	<b>FO</b>	1159,16	1156,40	1156,36	
	<b><math>\sigma</math></b>	13,71	11,87	10,88	
<b>3</b>	<b>FO</b>	6380,00	6380,00	6380,00	
	<b><math>\sigma</math></b>	0,00	0,00	0,00	

Para o problema com 30 portos, os melhores resultados são obtidos quando a taxa de cruzamento é de 80%, conforme representado na Tabela 22. Para os casos das Matrizes tipo 1 e 2, as médias mostradas foram 2385,76 e 1156,36 movimentos. Já para o caso da Matriz tipo 3, o resultado ótimo é encontrado em todos os casos. Para este problema foi necessário 5 minutos para simular cada solução.

Da mesma forma que verificado no problema com percurso de 20 portos, os problemas com 25 Portos (Tabela 21) e com 30 Portos (Tabela 22) apresentam os melhores resultados para uma configuração com 80% de probabilidade de cruzamento para os casos das Matrizes tipo 1 e 2. Em relação ao caso da Matriz tipo 3, em todas as simulações foi encontrado o valor de FO mínimo.

À medida que a probabilidade de cruzamento diminui, os resultados médios apresentam uma melhora. Por este motivo, foi necessário realizar alguns testes assumindo-se a probabilidade de cruzamento de 75%. Nos testes realizados, verificou-se que os resultados médios da FO aumentaram, ou seja, a taxa de cruzamento de 75% não gera resultados tão bons quanto os encontrados quando a taxa de cruzamento é de 80%.

Ao analisar cada problema teste do Cenário 3, foi feita uma breve relação dos resultados com os resultados obtidos nos demais cenários já apresentados. De forma geral, o Cenário 3 não apresentou resultados promissores. O que difere este cenário dos Cenários 1 e 2 é o operador de mutação adaptativa, que por sua vez não apresentou um bom desempenho nos testes realizados.

#### 6.1.1.4 Cenário 4

A seguir serão mostradas as Tabelas de soluções do Cenário 4. Neste cenário, as simulações são feitas com base nos operadores de mutação adaptativa e cruzamento um ponto de corte.

**Tabela 23 - Cenário 4- AG 10 Portos**

<b>Problema com 10 Portos – Cenário 4</b>					
<b>Mutação Adaptativa</b>	<b>Tipo da Matriz</b>		<b>Probabilidade de Cruzamento</b>		
			<b>100</b>	<b>90</b>	<b>80</b>
	<b>1</b>	<b>FO</b>		1335,76	1336,26
	<b><math>\sigma</math></b>		1,04	1,17	1,97
<b>2</b>	<b>FO</b>		760,00	760,00	760,00
	<b><math>\sigma</math></b>		0,00	0,00	0,00
<b>3</b>	<b>FO</b>		2282,00	2282,00	2282,00
	<b><math>\sigma</math></b>		0,00	0,00	0,00

Para o caso do percurso com 10 portos, Tabela 23, o caso da Matriz tipo 1 apresenta o resultado de 1335,76 movimentos quando assumida a taxa de cruzamento de 100%. Ao comparar este resultado com o do Cenário 3, onde obteve-se 1335,96 movimentos, percebe-se que a configuração empregada no Cenário 4 forneceu uma média melhor que a do Cenário 3.

Para os casos das Matrizes tipo 2 e 3, os resultados em todos os casos são idênticos aos resultados mínimos encontrados nos demais cenários. Embora a média para o caso da Matriz tipo 2 não alcance o ótimo esperado, os resultados se mostram muito próximos ao valor mínimo conhecido. O tempo gasto para simular cada experimento deste problema é, em média, o mesmo que no Cenário 3.

Na Tabela 24, são apresentados os resultados referentes ao percurso com 15 portos. O tempo computacional para este problema mostra a mesma média verificada no Cenário 3.

Nas simulações do problema com 15 Portos, os resultados encontrados foram melhores que os encontrados no Cenário 3, para o mesmo problema, porém não melhores que os resultados dos Cenários 1 e 2. Conferindo as médias obtidas para os casos das Matrizes tipo 1 e 2, nota-se que os melhores resultados estão concentradas para o caso da taxa de cruzamento de 80%, com valores de 1618,26 e 790,53 movimentos, na mesma ordem.

Tabela 24 - Cenário 4- AG 15 Portos

Problema com 15 Portos					
Mutaç�o Adaptativa	Tipo da Matriz		Probabilidade de Cruzamento		
			100	90	80
	1	FO		1620,03	1620,00
$\sigma$			3,80	3,92	3,60
2	FO		790,93	790,63	790,53
		$\sigma$	2,49	2,18	2,35
3	FO		3024	3024	3024
		$\sigma$	0	0	0

Da mesma forma que nos cen rios anteriores, para o caso da Matriz tipo 3, foi encontrado o valor m dio de 3024,00 movimentos em todos os casos. Tal solu o   equivalente ao resultado m nimo esperado.

Tabela 25 - Cen rio 4- AG 20 Portos

Problema com 20 Portos – Cen�rio 4					
Mutaç�o Adaptativa	Tipo da Matriz		Probabilidade de Cruzamento		
			100	90	80
	1	FO		2074,83	2076,53
$\sigma$			13,13	12,14	12,20
2	FO		830,60	830,23	824,56
		$\sigma$	8,03	7,62	8,57
3	FO		4880,00	4880,00	4880,00
		$\sigma$	0,00	0,00	0,00

A Tabela 25 corresponde aos resultados obtidos para o problema com 20 portos. Atrav s dos valores m dios obtidos para o caso da Matriz tipo 1, observa-se que a configura o com 100% de taxa de cruzamento forneceu o melhor resultado, que   menor que a m dia obtida no Cen rio 3 de 2085,33 movimentos. Para o caso da Matriz tipo 2, o menor valor m dio de FO obtido foi de 824,56 movimentos, para a configura o com 80% de taxa de cruzamento, que por sua vez   menor que a m dia encontrada no Cen rio 3.

Para o caso da Matriz tipo 3, foi atingido o valor de 4880,00 movimentos, que   id ntico ao resultado  timo. Conforme os demais cen rios apresentados, em todos os casos foi poss vel encontrar tal solu o.

Tabela 26 - Cenário 4- AG 25 Portos

Problema com 25 Portos – Cenário 4					
Mutaçã Adaptativa	Tipo da Matriz		Probabilidade de Cruzamento		
			100	90	80
	1	FO		1720,00	1720,46
$\sigma$			12,30	10,55	10,06
2	FO		1000,16	996,75	996,60
		$\sigma$	7,25	9,71	9,05
3	FO		5492,00	5492,00	5492,00
		$\sigma$	0,00	0,00	0,00

Da mesma forma que verificado para o percurso com 20 portos, os resultados obtidos para o problema com 25 portos, Tabela 26, para os casos das Matrizes tipo 1 e 2, os seus menores resultados foram encontrados quando assumido os valores 100% e 80% para a taxa de cruzamento, respectivamente. Ao se efetuar uma comparação com os resultados do Cenário 3, para este mesmo problema, verifica-se novamente uma superioridade dos resultados obtidos no Cenário 4. Para o caso da Matriz tipo 3, foi obtido o valor médio de 5492,00 movimentos com desvio-padrão nulo. O tempo gasto para este problema é, em média, de 4 a 5 minutos.

Tabela 27 - Cenário 4- AG 30 Portos

Problema com 30 Portos – Cenário 4					
Mutaçã Adaptativa	Tipo da Matriz		Probabilidade de Cruzamento		
			100	90	80
	1	FO		2369,20	2370,53
$\sigma$			18,58	12,29	14,45
2	FO		1145,20	1146,36	1148,60
		$\sigma$	13,84	11,44	14,29
3	FO		6380,00	6380,00	6380,00
		$\sigma$	0	0	0

Para o problema com 30 portos, Tabela 27, para o caso da Matriz tipo 1, foi encontrado o valor médio da FO de 2367,36 movimentos com desvio padrão de 14,45, quando empregado o valor de 80% para a taxa de cruzamento. Para o caso da Matriz tipo 2, a menor média de FO foi de 1145,20, encontrada quando o cruzamento é realizado em 100% das vezes.

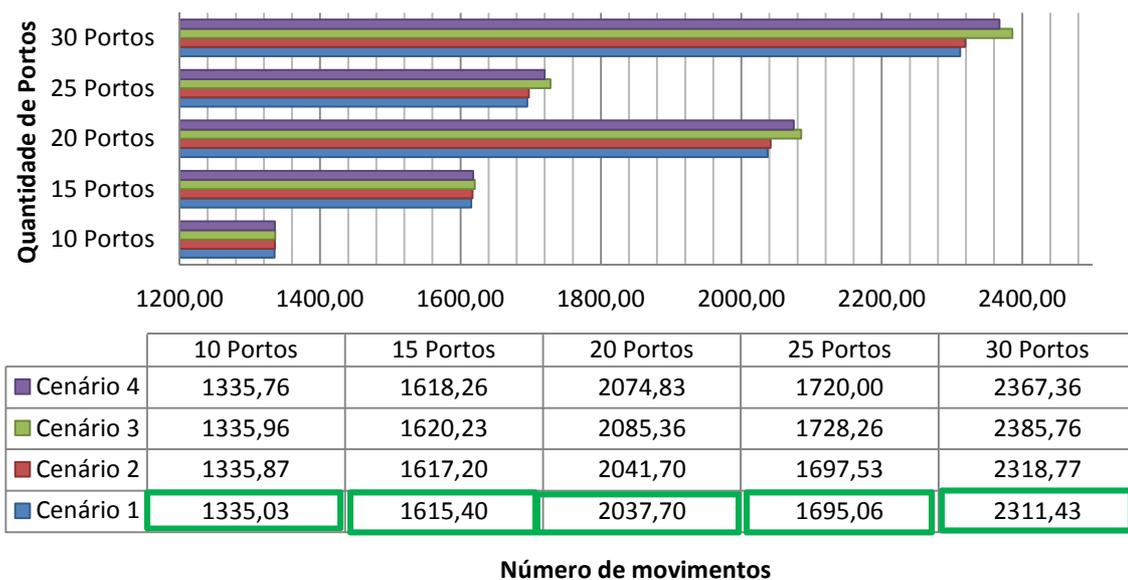
Os resultados médios obtidos para os casos das Matrizes tipo 1 e 2, também são classificados como melhores se comparados com os valores médios obtidos, para a mesma instância, no Cenário 3. Já os resultados obtidos para o caso da Matriz tipo 3 não diferem dos resultados encontrados anteriormente. O tempo gasto para este problema é, em média, de 5 minutos.

Ao longo das análises feitas neste cenário de teste, pode-se perceber melhores resultados que aqueles obtidos no Cenário 3, isso se deve ao emprego de um operador de cruzamento diferente. Com base nos dois cenários, conclui-se que, quando aplicada mutação adaptativa, o operador de cruzamento um ponto de corte apresenta desempenho melhor que o cruzamento uniforme. Além disso, na maioria dos casos, as menores médias foram encontradas quando a operação de cruzamento acontecia em todos os casos (100%), ou quando o cruzamento acontecia em 80% dos casos.

#### 6.1.1.5 Comparativo entre cenários do AG

Nesta seção, efetua-se um comparativo entre os quatro cenários de teste apresentados, com o intuito de identificar a melhor configuração para os operadores de mutação e cruzamento e, em consequência, os melhores valores para as taxas de uso destes operadores. Cada gráfico mostrado a seguir se refere a um dos três tipos de Matriz de Transporte e os resultados apresentados se referem às melhores médias encontradas em cada cenário, ou seja, as médias que foram destacadas com sombreado nas tabelas das seções anteriores.

**Figura 29 - Comparativo das melhores médias de resultados de cada cenário referente à Matriz Tipo 1**

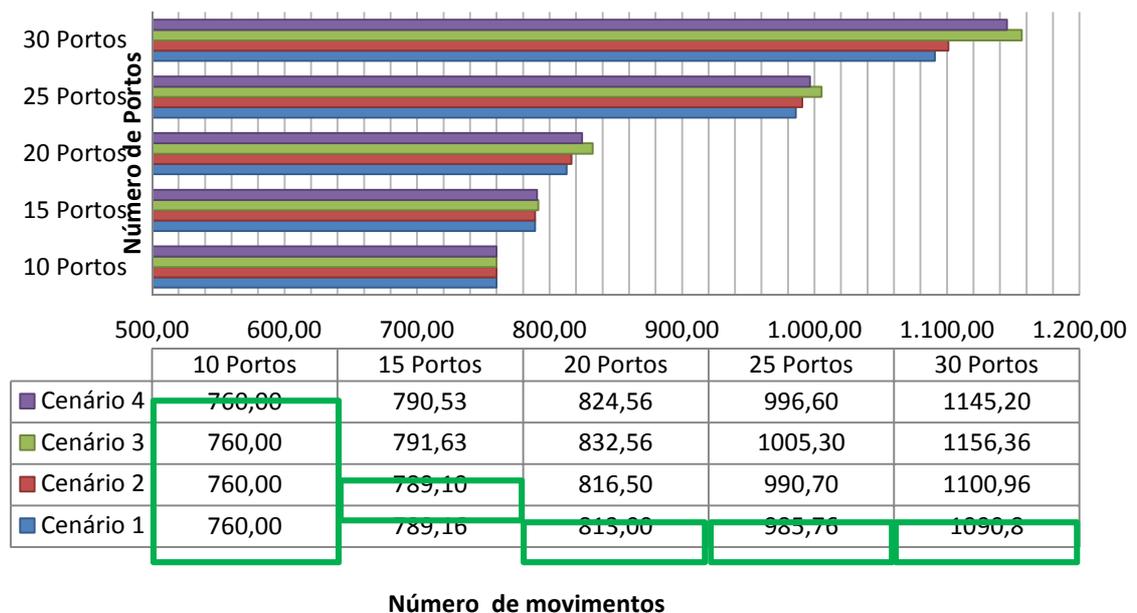


Conforme a Figura 29, o Cenário 3, representado pela cor verde, obteve os maiores resultados em todas as simulações. Por ser um problema de minimização, o Cenário 3 teve o

pior desempenho. Nele foram aplicados o cruzamento uniforme e a mutação adaptativa. O Cenário 4, que possui semelhança com o Cenário 3 por ser simulado com o mesmo operador de mutação, apresentou melhora nos resultados ao substituir o cruzamento uniforme pelo cruzamento um ponto de corte. Apesar do Cenário 4 obter melhores resultados que o Cenário 3, não conseguiu alcançar a qualidade dos resultados dos Cenários 1 e 2.

Por este motivo, a mutação mais adequada para o problema de carregamento de navios-contêineres é a Mutação Uniforme, a qual foi aplicada nas simulações dos Cenários 1 e 2. Em relação ao cruzamento mais eficiente, basta analisar entre os Cenários 1 e 2, qual atingiu um menor número de movimentos. O Cenário 1 obteve um melhor desempenho e nele foi aplicado o Cruzamento Uniforme.

**Figura 30 - Comparativo das melhores médias de resultados de cada cenário referente à Matriz Tipo 2**



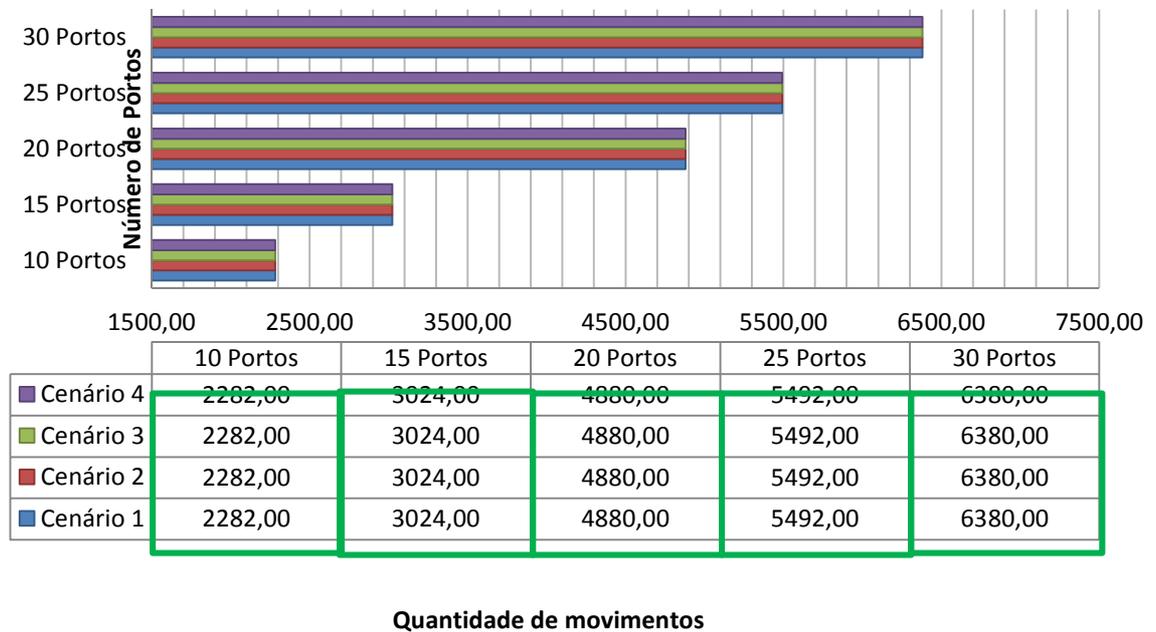
Na Figura 30, para o caso da Matriz tipo 2, novamente o Cenário 3 apresenta o pior desempenho comparado aos outros cenários. No percurso com 10 portos, todos os cenários encontraram o mesmo resultado. Apesar de não ser o valor ótimo, está muito próximo aos 750,00 movimentos esperados. Desta foram, acredita-se que o algoritmo tenha ficado preso neste mínimo local de 760,00 movimentos. Em relação ao caso com 15 Portos, os Cenários de melhor desempenho foram os Cenários 1 e 2, com uma pequena vantagem para o Cenário 2, que atingiu o menor resultado.

Para os casos com 20, 25 e 30 portos, o Cenário 1 se sobressaiu. Em suma, o Cenário 1 continua sendo a melhor configuração para os testes deste problema, quando aplicadas as dezoito combinações de regras. O tempo computacional despendido para o caso da Matriz

tipo 2 é muito semelhante ao do caso da Matriz tipo 1. O que determina um maior gasto de tempo é a quantidade de portos, ou seja, resolver o mesmo problema num percurso de 30 portos despende muito mais tempo do que se comparado a resolver um problema num percurso com 25 portos, que por sua vez exige um esforço computacional maior que o exigido pelo caso com 20 portos, e assim por diante. Diferentes parâmetros como troca do operador, ou diferentes valores de probabilidade, não interferiram no tempo do algoritmo.

Na Figura 31 são expostos os resultados médios para o caso da Matriz Tipo 3.

**Figura 31 - Comparativo das melhores médias de resultados de cada cenário referente à Matriz Tipo 3**



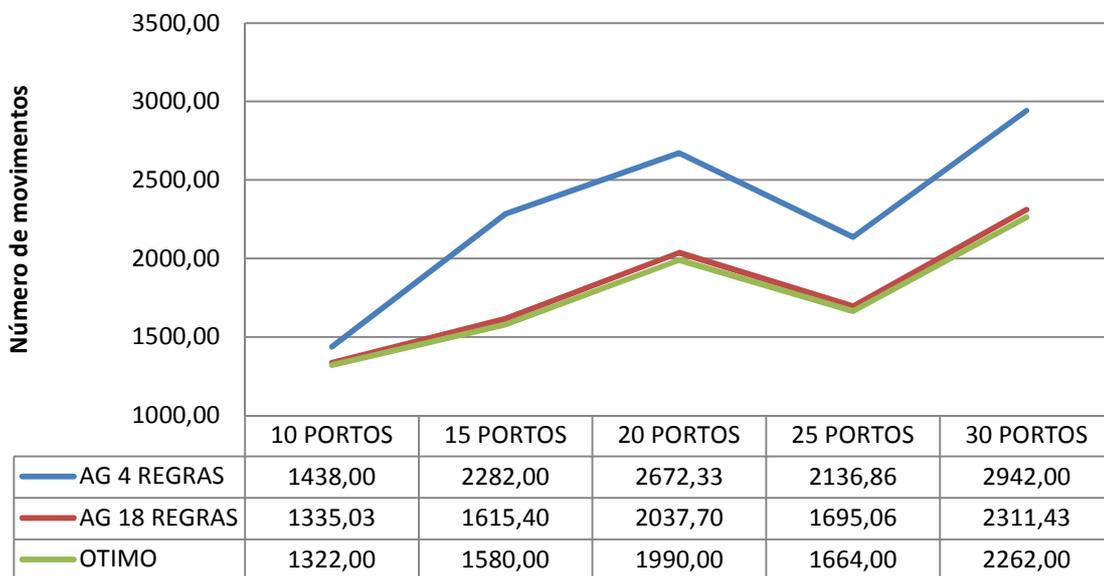
Analisando os resultados médios de cada cenário, verifica-se uma homogeneidade nas soluções. Os resultados independem dos Cenários, pois em todos os casos foi possível encontrar o resultado ótimo. Acredita-se que, por ser uma matriz de curta distância, é possível encontrar os resultados com maior facilidade, pois os contêineres permanecem um período curto de tempo a bordo do navio. Logo, o algoritmo AG, aplicado com 18 combinações de regras, mostrou-se muito eficiente na solução dos casos da Matriz tipo 3 e os diferentes operadores empregados foram adequados para encontrar tais soluções.

Além disso, cabe salientar aqui que para o caso da Matriz tipo 3, quando aplicado o algoritmo AG com quatro combinações de regras, percebeu-se que as médias ficaram muito próximas do valor mínimo esperado, mas sem alcançá-lo em qualquer simulação. Já com o emprego do AG com dezoito combinações de regras, foi possível chegar à solução ótima com desvio-padrão nulo, em todos os casos.

### 6.1.1.6 Comparativo entre AG's

Após verificar todos os resultados e configurações do AG com dezoito combinações de regras, faz-se necessário um comparativo do AG aplicado com quatro e dezoito combinações de regras. Na Figura 32, fica claro o ganho na qualidade dos resultados que o algoritmo teve quando empregado um número maior de combinações de regras. O algoritmo com quatro combinações de regras refere-se ao desenvolvido neste estudo, que por sua vez mostrou-se melhor que o AG com quatro combinações de regras, proposto por Azevedo (2010).

**Figura 32 - Comparativo entre AG's Matriz 1**



Neste gráfico têm-se as melhores médias de resultados do Algoritmo Genético aplicado com quatro e dezoito combinações de regras e, ainda, a solução ótima do problema como base de comparação. Os resultados mostrados para cada porto são referentes ao caso da Matriz tipo 1.

De forma evidente, nota-se que, ao serem aplicadas dezoito combinações de regras, os resultados ficam mais próximos aos esperados (ótimos globais) do que quando aplicadas apenas quatro combinações de regras. Através deste gráfico, é perceptível a importância da aplicação de mais regras para a obtenção de melhores soluções.

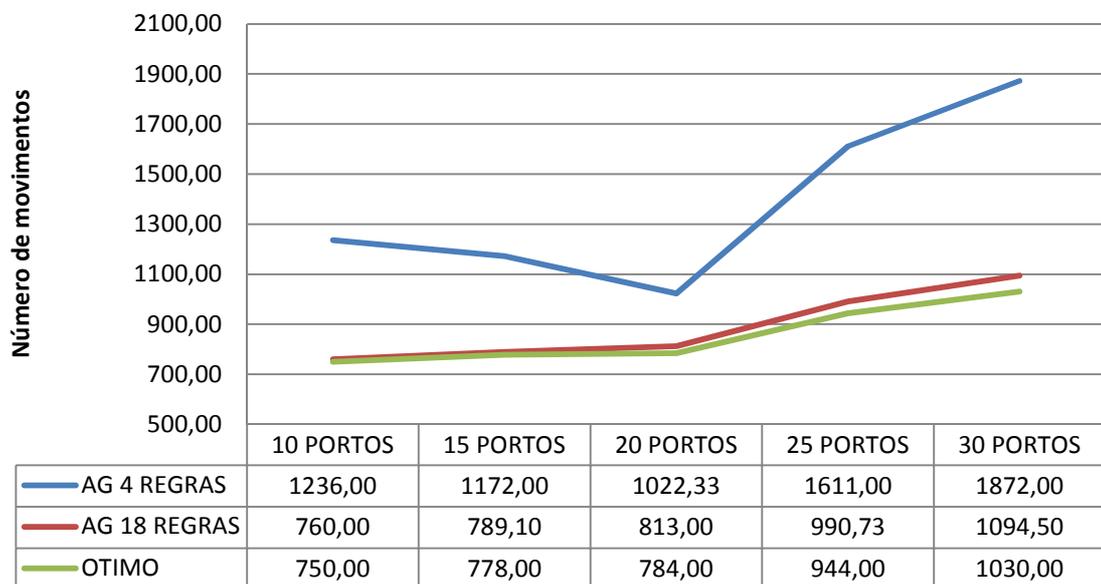
Na representação da Figura 33, têm-se as melhores médias para o problema com matriz de longa distância (tipo 2), resolvido pelo AG com quatro e dezoito combinações de regras. Observa-se, novamente, um bom desempenho do AG depois de aplicadas as dezoito combinações de regras.

Ao aplicar as 4 combinações de regras, o algoritmo apresenta bons resultados para o caso com 10 portos, porém afastados do esperado. Ao resolver o problema com 15 portos a qualidade dos resultados começa a melhorar, com pico de melhora no problema com 20

portos. À medida que a quantidade de portos aumenta, para 25 e 30 portos, os resultados obtidos com quatro combinações de regras se afastam do ótimo.

Já para o AG com dezoito combinações de regras, as soluções são muito próximas dos valores ótimos. Partindo do percurso com 10 portos e à medida que o número de portos aumenta, a dificuldade também aumenta, percebendo-se uma leve piora na qualidade das soluções. Apesar disso, os resultados médios com dezoito combinações de regras, mais uma vez, é consideravelmente melhor do que com quatro combinações de regras.

**Figura 33 - Comparativo entre AG's Matriz 2**

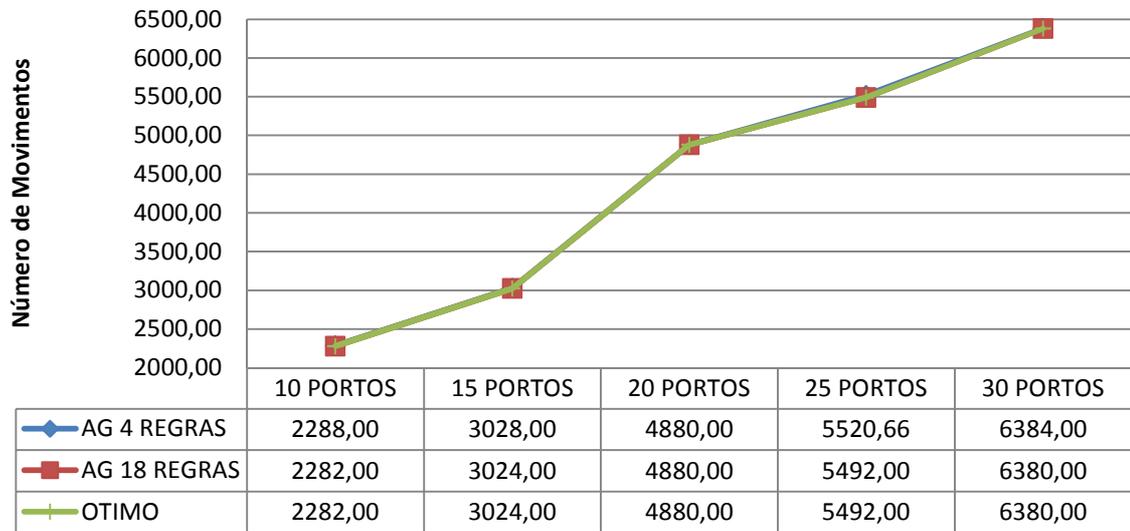


No gráfico da Figura 34, as médias apresentadas correspondem ao caso da Matriz de curta distância (tipo 3). Por este motivo não mostra evidente diferença entre os resultados, como visto nos gráficos anteriores para os demais tipos de Matriz de Transporte. Os resultados estão tão próximos do ótimo que, muitas vezes, fica difícil de visualizar a diferença entre eles.

Logo, de acordo com as médias obtidas quando empregado o AG aplicado com dezoito combinações de regras, o caso da Matriz tipo 3 obteve desempenho ótimo.

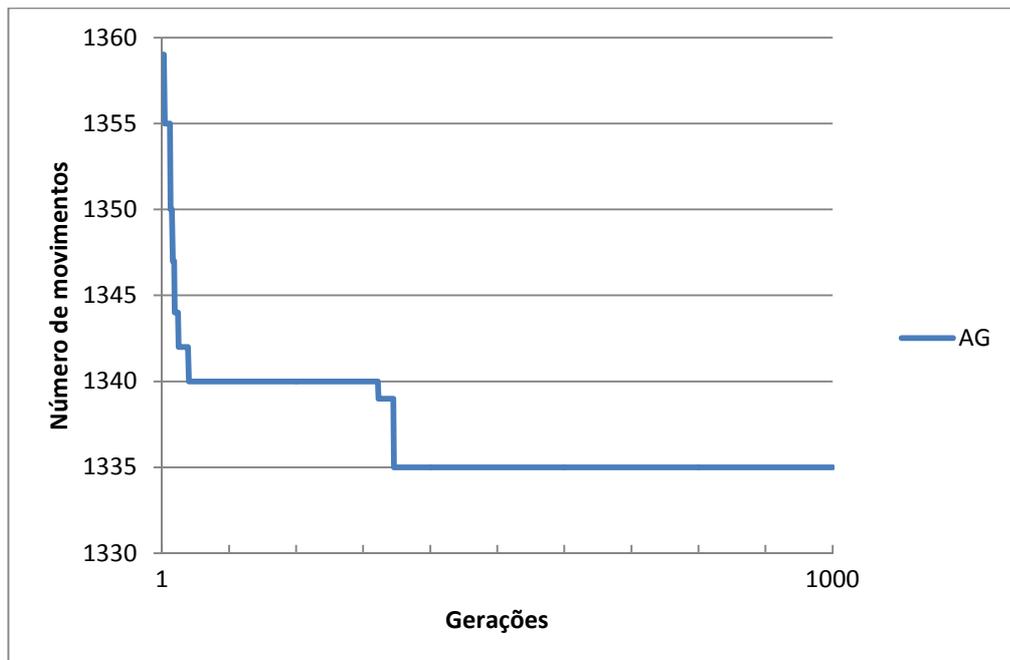
Depois de comentados todos os resultados e mencionadas as diferenças entre a aplicação do AG com quatro e dezoito combinações de regras, fica visível a influência das regras no comportamento do algoritmo para obtenção de bons resultados.

**Figura 34 - Comparativo entre AG's Matriz 3**



Em complemento às análises realizadas, é apresentado, na Figura 35, uma evolução típica da FO do AG para o problema com 10 portos e Matriz tipo 1. Para este resultado assumiu-se o AG com 18 combinações de regras, probabilidade de cruzamento de 90% e probabilidade de mutação de 15%, operador de cruzamento uniforme e operador de mutação uniforme.

**Figura 35 - Evolução das soluções do AG**



Conforme pode-se observar, para este problema a evolução das soluções se concentra nas primeiras gerações. Existem alguns períodos em que o algoritmo não obtém melhora na melhor solução, mas em outros períodos o AG é capaz de aprimorá-la. No último período, o algoritmo permanece estável até o final das 1000 gerações.

### 6.1.1.7 Experimentos com o ABC (Enxame de Abelhas)

Nesta seção, são exibidos os experimentos realizados com o algoritmo ABC, baseado na metaheurística Enxame de Abelhas. A partir dos resultados são realizadas comparações entre as duas metaheurísticas aplicadas neste estudo.

O código computacional do método ABC foi desenvolvido na ferramenta Matlab, e executado nas mesmas máquinas onde o Algoritmo Genético foi executado. Da mesma forma que no AG, aplicaram-se as dezoito combinações de regras, e as quinze instâncias divididas em três tipos de Matrizes de Transporte, distância mista (1), longa distância (2) e curta distância (3). Para cada quantidade de portos presente no percurso do navio (10, 15, 20, 25, e, 30) é apresentada uma tabela, sendo que cada tabela é dividida em dois cenários variando o número de fontes de alimento em 50 e 100. Os resultados médios mostrados se referem ao valor da FO e logo abaixo é apresentado o desvio-padrão ( $\sigma$ ) correspondente. Os resultados médios em negrito correspondem aos melhores de cada cenário, e os resultados médios em negrito e sombreados são os resultados que se destacaram entre os dois cenários. Na Tabela 28 são exibidos os resultados médios e seus respectivos desvios-padrão para o problema com 10 Portos.

Tabela 28 - ABC 10 Portos

Problema com 10 Portos					
Tipo da Matriz		Cenário 1: Vizinhaça Inversão		Cenário 2: Vizinhaça Troca	
		Fontes de Alimento		Fontes de Alimento	
		100	50	100	50
1	FO	<b>1356,45</b>	1359,60	<b>1354,75</b>	1355,75
	$\sigma$	<b>6,91</b>	10,79	<b>7,11</b>	8,60
2	FO	<b>773,07</b>	778,86	<b>771,77</b>	772,04
	$\sigma$	<b>8,48</b>	9,43	<b>4,33</b>	7,02
3	FO	<b>2282,00</b>	<b>2282,00</b>	<b>2282,00</b>	<b>2282,00</b>
	$\sigma$	<b>0,00</b>	<b>0,00</b>	<b>0,00</b>	<b>0,00</b>

Observando o Cenário 1, verifica-se que os resultados para os casos das Matrizes tipo 1 e 2 melhoraram quando o número de fontes de alimento foi aumentado, tendo acontecido da mesma forma no Cenário 2. Apesar disso, o Cenário 2 forneceu um valor médio de 1355,75

movimentos, quando empregadas 50 fontes de alimento, enquanto o melhor resultado do Cenário 1 é de 1356,45 movimentos, com o dobro de fontes de alimentos. O mesmo comportamento também foi verificado para o caso da Matriz tipo 2. Em resumo, a melhor FO média para o caso da Matriz tipo 1 foi de 1354,75 movimentos com desvio-padrão de 7,11, e para o caso da Matriz tipo 2 foi de 771,77 movimentos com desvio-padrão de 4,33. Se comparadas estas soluções aos resultados encontrados pelo algoritmo ABC com 4 combinações de regras, que obteve 1461,63 e 1259,27 movimentos respectivamente, pode-se notar que o emprego de 18 combinações de regras proporcionou uma melhora significativa nas soluções. Já para o caso da Matriz tipo 3, o resultado ótimo foi alcançado em todos os testes, diferente do ABC com 4 combinações de regras, que não conseguiu chegar na solução mínima esperada.

Logo, para o problema com 10 portos, os melhores resultados se concentraram no Cenário 2 que possui o operador de vizinhança por troca e com 100 fontes de alimento. Em contrapartida aos bons resultados, quanto maior a quantidade de fontes de alimento, mais elevado é o tempo computacional exigido. O tempo computacional para resolver este problema pelo ABC é de aproximadamente 85 a 90 minutos para cada solução de 1000 gerações.

Na Tabela 29 são apresentadas as soluções do Problema com 15 Portos.

**Tabela 29 - ABC 15 Portos**

<b>Problema com 15 Portos</b>					
<b>Tipo da Matriz</b>		<b>Cenário 1: Vizinhança Inversão</b>		<b>Cenário 2: Vizinhança Troca</b>	
		<b>Fontes de Alimento</b>		<b>Fontes de Alimento</b>	
		<b>100</b>	<b>50</b>	<b>100</b>	<b>50</b>
<b>1</b>	<b>FO</b>	<b>1679,50</b>	1689,33	<b>1657,81</b>	1662,00
	<b><math>\sigma</math></b>	<b>14,99</b>	41,37	<b>21,71</b>	19,54
<b>2</b>	<b>FO</b>	<b>832,80</b>	833,83	813,66	<b>809,60</b>
	<b><math>\sigma</math></b>	<b>13,30</b>	17,02	12,53	<b>11,12</b>
<b>3</b>	<b>FO</b>	<b>3024,00</b>	<b>3024,00</b>	<b>3024,00</b>	<b>3024,00</b>
	<b><math>\sigma</math></b>	<b>0,00</b>	<b>0,00</b>	<b>0,00</b>	<b>0,00</b>

Nesta Tabela são apresentados os resultados médios do Problema com 15 Portos. Para o caso da Matriz tipo 1 e Cenário 1, bons resultados são encontrados quando utilizadas 100 fontes de alimento, obtendo-se uma média de 1679,50 movimentos e desvio padrão de 14,99. O mesmo acontece no Cenário 2, onde o melhor resultado é encontrado com 100 fontes de alimento, contabilizando 1657,81 movimentos e 21,71 de desvio padrão, resultando na FO média deste cenário melhor que a do Cenário 1. Já quando empregado o algoritmo ABC com

4 combinações de regras, a média da FO foi de 2446,50 movimentos, tendo se distanciado da solução esperada, mas com a aplicação de 18 combinações de regras esta situação mudou.

Para o caso da Matriz tipo 2, a média com 100 fontes de alimento é melhor no Cenário 1, mas as médias são muito próximas. Já no Cenário 2, tanto empregando-se 50 quanto 100 fontes de alimento, os resultados são melhores que os resultados do Cenário 1. Entretanto, no Cenário 2, o destaque está para a configuração com 50 fontes de alimento, onde foi obtida uma média de 809,60 movimentos, diferenciando-se dos demais resultados que encontraram seu ótimo com 100 fontes de alimento. Comparando com os resultados do algoritmo ABC com 4 combinações de regras, novamente a média atual da FO foi melhor que a média de 1273,33 movimentos encontrada anteriormente.

Para o caso da Matriz tipo 3, os resultados alcançaram o ótimo em todos os casos, causando uma uniformidade dos resultados médios e do desvio padrão zero. O tempo gasto com este problema teste foi em média de 2 a 2,4 horas.

O problema com 20 portos é representado na Tabela 30.

**Tabela 30 - ABC 20 Portos**

<b>Problema com 20 Portos</b>					
<b>Tipo da Matriz</b>		<b>Cenário 1: Vizinhança Inversão</b>		<b>Cenário 2: Vizinhança Troca</b>	
		<b>Fontes de Alimento</b>		<b>Fontes de Alimento</b>	
		<b>100</b>	<b>50</b>	<b>100</b>	<b>50</b>
<b>1</b>	<b>FO</b>	2314,87	<b>2291,06</b>	2211,00	<b>2209,33</b>
	<b><math>\sigma</math></b>	58,42	<b>113,20</b>	63,06	<b>88,41</b>
<b>2</b>	<b>FO</b>	<b>889,57</b>	899,34	872,30	<b>863,70</b>
	<b><math>\sigma</math></b>	<b>13,27</b>	23,01	19,46	<b>22,64</b>
<b>3</b>	<b>FO</b>	<b>4880,00</b>	<b>4880,00</b>	<b>4880,00</b>	<b>4880,00</b>
	<b><math>\sigma</math></b>	<b>0,00</b>	<b>0,00</b>	<b>0,00</b>	<b>0,00</b>

Conforme os resultados apresentados na Tabela 30, a média da FO encontrada no Problema com 20 portos e no Cenário 1, se destacou quando havia apenas 50 fontes de alimento e da mesma forma aconteceu no Cenário 2. A melhor média está localizada no Cenário 2, com 2209,33 movimentos e com um desvio-padrão elevado de 88,41. Apesar do desvio-padrão ser alto, mesmo assim foi possível encontrar um valor médio da FO menor que aquele obtido quando empregado o algoritmo ABC com 4 combinações de regras, que contabilizou 3121,14 movimentos.

Para o caso da Matriz tipo 2, no Cenário 1 a menor FO média foi obtida quando aplicadas 100 fontes de alimento, diferente do Cenário 2 que obteve este resultado com 50 fontes de alimento. No entanto, no Cenário 2, que aplica a vizinhança por troca, foi possível alcançar uma menor média com 863,70 movimentos e desvio-padrão de 22,64. Novamente

observa-se um melhor desempenho que o mesmo algoritmo aplicado com 4 combinações de regras.

Para o caso da matriz de distância curta, foi possível obter a melhor FO média independente do cenário e da quantidade de fontes de alimento, diferentemente do resultado encontrado com o uso do algoritmo ABC com menos regras. Em resumo, as melhores médias se encontram no Cenário 2, quando utilizadas 100 fontes de alimento. O tempo gasto para simular este problema foi, em média, de 3 a 3,5 horas.

Com relação ao problema com 25 portos, os resultados são apresentados na Tabela 31. Para o caso da Matriz tipo 1, os melhores resultados foram encontrados com 100 fontes de alimento em ambos os Cenários avaliados. Porém, a média que ganhou destaque está no Cenário 2, com 1781,75 movimentos e 13,18 de desvio-padrão. Ao comparar este resultado ao resultado obtido com o mesmo programa, porém com o uso de 4 combinações de regras, nota-se que o algoritmo com mais regras novamente obteve um desempenho superior que o algoritmo anterior, que forneceu 2345,14 movimentos como resultado médio, ou seja, o algoritmo melhorou 31% quando aplicadas as 18 combinações de regras.

**Tabela 31 - ABC 25 Portos**

<b>Problema com 25 Portos</b>					
<b>Tipo da Matriz</b>		<b>Cenário 1: Vizinhança Inversão</b>		<b>Cenário 2: Vizinhança Troca</b>	
		<b>Fontes de Alimento</b>		<b>Fontes de Alimento</b>	
		<b>100</b>	<b>50</b>	<b>100</b>	<b>50</b>
<b>1</b>	<b>FO</b>	<b>1801,87</b>	1819,96	<b>1781,75</b>	1786,70
	<b><math>\sigma</math></b>	<b>43,39</b>	31,47	<b>13,18</b>	41,81
<b>2</b>	<b>FO</b>	1108,00	<b>1100,29</b>	<b>1062,00</b>	1067,20
	<b><math>\sigma</math></b>	30,52	<b>26,59</b>	<b>11,18</b>	16,26
<b>3</b>	<b>FO</b>	<b>5492,00</b>	5494,30	<b>5492,60</b>	5493,55
	<b><math>\sigma</math></b>	<b>0,00</b>	2,45	<b>0,89</b>	2,40

O problema considerando a Matriz tipo 2 foi resolvido pelo ABC obtendo como média da FO 1062,00 movimentos e 11,18 de desvio-padrão, melhores que os 1706,66 movimentos encontrados pelo ABC com 4 combinações de regras. Esta solução corresponde às simulações do Cenário 2, quando foram consideradas 100 fontes de alimento.

Para o caso da Matriz tipo 3, o resultado ótimo foi alcançado em apenas um caso, no Cenário 1, com 100 fontes de alimento. Distinto dos demais exemplos, não foi possível chegar à solução ótima em todos os casos. Porém, mesmo não alcançando o ótimo, foi possível melhorar a média encontrada com a aplicação do algoritmo ABC com 4 combinações de regras. Para simular este problema, foi necessário um tempo de 3,5 a 4,5 horas para cada solução.

Tabela 32 - ABC 30 Portos

Problema com 30 Portos					
Tipo da Matriz		Cenário 1: Vizinhança Inversão		Cenário 2: Vizinhança Troca	
		Fontes de Alimento		Fontes de Alimento	
		100	50	100	50
1	FO	2545,66	2620,77	2543,90	2539,80
	$\sigma$	101,02	65,93	98,02	85,02
2	FO	1288,71	1332,1	1247,70	1280,71
	$\sigma$	76,45	116,69	94,42	72,32
3	FO	6380,00	6380,00	6380,00	6380,00
	$\sigma$	0,00	0,00	0,00	0,00

A Tabela 32 apresenta os resultados referentes ao problema com 30 portos. Este problema atingiu o maior tempo gasto com simulações neste trabalho, variando de 4,4 a 4,6 horas.

Para o Cenário 1, as melhores soluções foram obtidas quando empregadas 100 fontes de alimentos, diferente do Cenário 2 que dividiu igualmente as soluções entre os dois parâmetros de fontes de alimento avaliados. Para o caso da Matriz tipo 1, a menor FO média foi obtida no Cenário 2, com 2539,80 movimentos e desvio-padrão de 85,02. Este valor de FO média é melhor que o valor de 3539,14 movimentos, obtido quando aplicadas somente 4 combinações de regras.

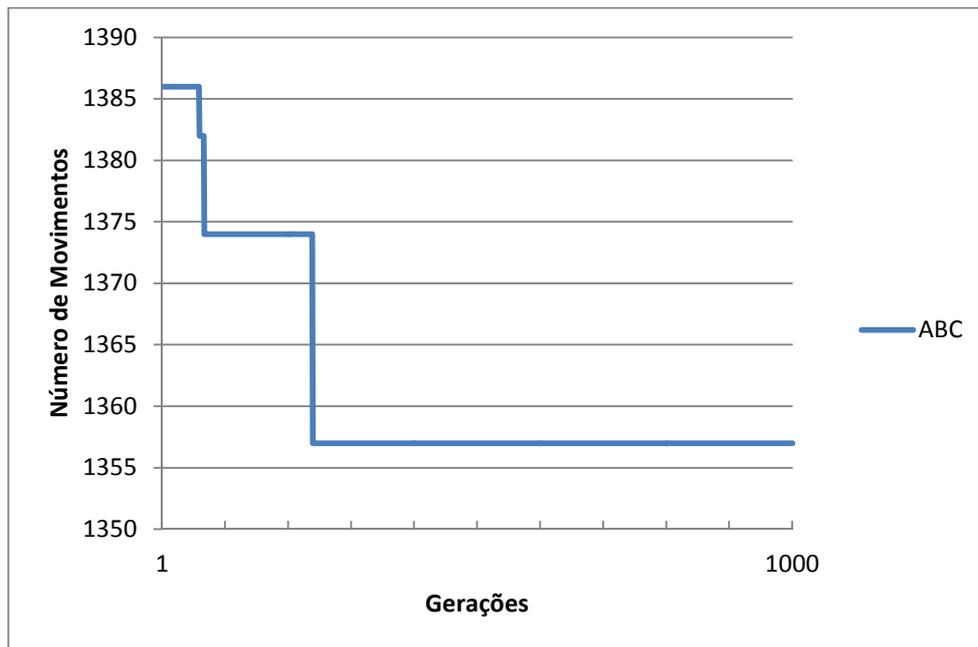
Ao analisar o caso da Matriz tipo 2, encontrou-se a média de 1247,70 movimentos no Cenário 2 com o emprego de 100 fontes de alimento. Este valor é melhor que a média encontrada com o mesmo algoritmo, porém utilizando-se 4 combinações de regras. Para a matriz tipo 3, as médias foram idênticas e obtiveram desempenho ótimo com desvio-padrão nulo.

Depois de avaliados os resultados obtidos a partir dos experimentos realizados com o algoritmo ABC, pode-se identificar o melhor Cenário. Em todas as tabelas, o Cenário 2 obteve um melhor desempenho que o Cenário 1, exceto para o problema com 25 portos para o caso da Matriz tipo 3. Logo, a estrutura de vizinhança por troca é a mais indicada para futuros testes. Em relação ao número de fontes de alimento, neste cenário foram encontradas 10 melhores soluções quando utilizadas 100 fontes de alimento e 9 melhores soluções quando utilizadas 50 fontes de alimentos. Então, na maioria das vezes, o uso de 100 fontes de alimento produziu soluções de melhor qualidade, embora a diferença tenha sido mínima, com o algoritmo com 50 fontes de alimento também se mostrando como uma boa opção para futuras simulações. Se avaliado o tempo computacional gasto nas simulações, quando consideradas 100 fontes de alimentos, que é consideravelmente superior ao tempo gasto nas

simulações quando consideradas 50 fontes de alimento, talvez isso também possa auxiliar na definição da quantidade de fontes de alimento mais adequada.

Em complemento às análises realizadas, é apresentado na Figura 36 a evolução da FO para uma simulação típica do ABC, para o problema com 10 portos e Matriz tipo 1. Assumiu-se o ABC com 18 combinações de regras, operador de vizinhança por troca e 100 fontes de alimento.

**Figura 36 - Evolução das soluções do ABC**



Analisando a Figura 36, pode-se observar que existe uma evolução maior das soluções nas primeiras gerações, e a tendência do algoritmo ao longo das gerações é de estabilização, obtendo pouca evolução.

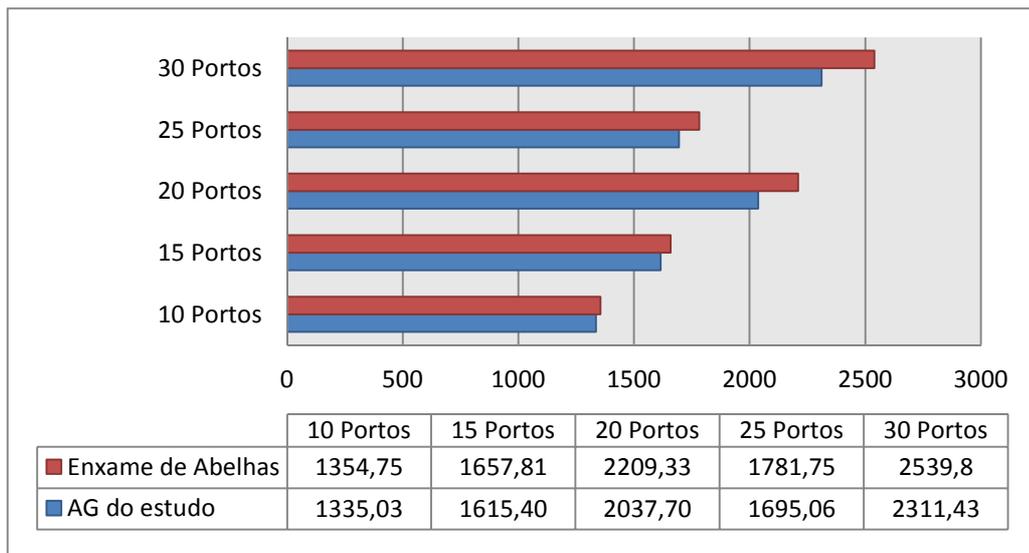
## 6.2 Comparativo entre os métodos de solução

Após analisados os experimentos realizados, é necessário estabelecer uma comparação entre os dois algoritmos aplicados neste estudo, o AG e o ABC. Cada gráfico mostrado nesta seção refere-se a um tipo de Matriz de Transporte (1, 2 ou 3).

As comparações serão realizadas para as cinco rotas diferentes, a partir das soluções destacadas em cada problema teste do algoritmo correspondente. Este comparativo tem o intuito de eleger o algoritmo mais eficiente para resolução do problema de carregamento e descarregamento de navios-contêineres.

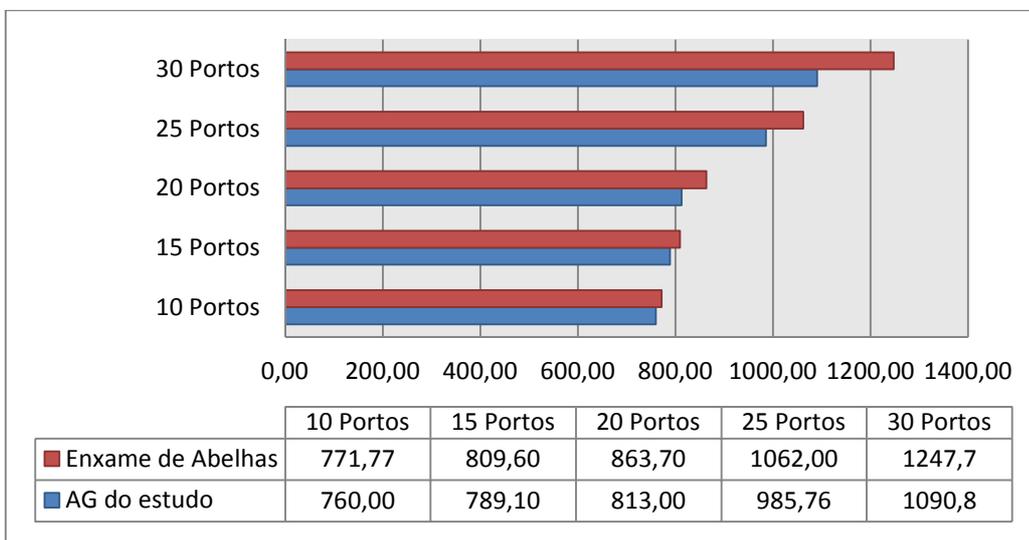
Na Figura 37, observa-se que o AG atingiu menores médias em todos os testes realizados para a Matriz tipo 1. Sem dúvida o AG possui um desempenho melhor que o ABC para este tipo de matriz de distância mista. Apesar de não existir diferença tão destacada quanto foram as diferenças constatadas nas aplicações dos algoritmos com 4 e 18 combinações de regras, o ABC não teve bom desempenho quando comparado ao AG deste estudo. Porém, se comparado ao AG proposto por Azevedo (2010), o ABC obteve resultados consideravelmente melhores.

**Figura 37 - AG x ABC (Matriz 1)**



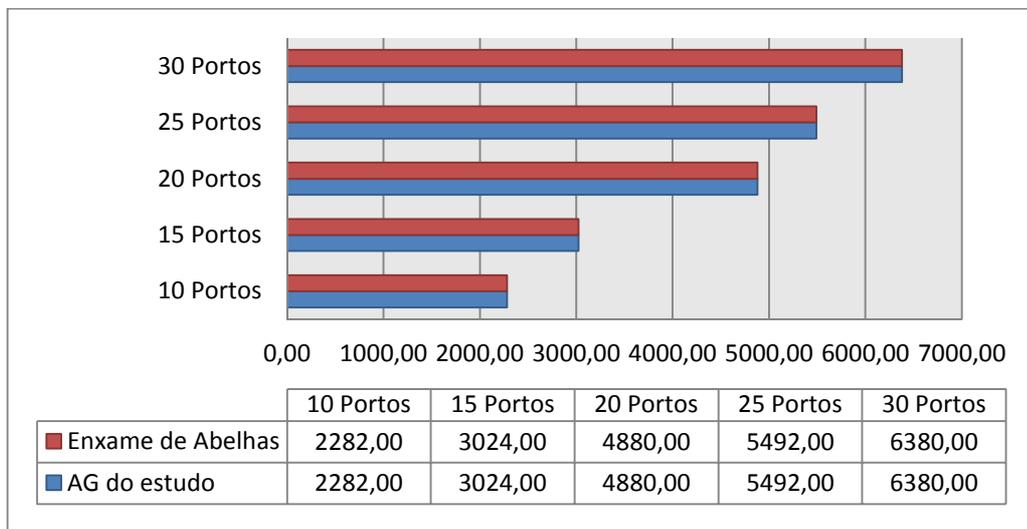
Seguindo a análise, agora são avaliados os resultados da Figura 38, referente ao problema para a matriz tipo 2. As médias do AG apresentadas são melhores que as médias do ABC para todos os cinco percursos, da mesma forma como verificado para o caso da matriz tipo 1, analisada anteriormente.

**Figura 38 - AG x ABC (Matriz 2)**



Na Figura 39, as médias referem-se ao problema considerando a matriz de curta distância (tipo 3). Nestas simulações, nota-se uma homogeneidade nas melhores médias, o que significa que não há distinção entre os algoritmos quando aplicados na solução do problema para o caso da matriz tipo 3.

**Figura 39 - AG x ABC (Matriz 3)**



Para auxiliar as análises entre os algoritmos, na seção 6.3 são realizadas outras avaliações como complemento.

### 6.3 Avaliações complementares

Nas avaliações complementares, é aplicado o GAP, medida que define a diferença relativa percentual de cada solução média em relação à melhor solução. Além disso, nesta seção são exploradas algumas simulações para compreender como se dá a aplicação das combinações de regras nos portos.

A Tabela 33 confronta os melhores resultados obtidos entre todas as simulações realizadas, com os dois algoritmos aplicados, utilizando 4 e 18 combinações de regras.

Analisando o GAP calculado para o AG com 4 combinações de regras, percebe-se valores elevados, o que corresponde a uma grande diferença percentual em relação à solução ótima. Já o AG com 18 combinações de regras, em geral, fornece baixas diferenças percentuais em relação ao resultado ótimo.

O algoritmo ABC com 4 combinações de regras também forneceu resultados contendo diferenças elevadas, em algumas vezes até iguais às encontradas pelo AG também com 4 combinações de regras, mas na maioria das vezes existe uma diferença maior no ABC de 4 regras. Consequentemente, o AG com 4 combinações de regras apresentou soluções mais próximas das soluções ótimas que o ABC.

Tabela 33 - GAP

Inst.	AG 4R	AG 18R	ABC 4R	ABC 18R	ÓTIMO	GAP (%)			
	FO	FO	FO	FO	FO	AG 4R	AG 18R	ABC 4R	ABC 18R
<b>10/1</b>	1438	1332	1438	1341	1322	8,77	<b>0,75</b>	8,77	1,43
<b>10/2</b>	1236	760	1236	760	750	64,8	<b>1,33</b>	64,8	1,33
<b>10/3</b>	2288	2282	2288	2282	2282	0,26	<b>0,00</b>	0,26	<b>0,00</b>
<b>15/1</b>	2280	1603	2366	1629	1580	44,30	<b>1,45</b>	49,74	3,10
<b>15/2</b>	1172	788	1182	792	778	50,64	<b>1,28</b>	51,92	1,79
<b>15/3</b>	3028	3024	3028	3024	3024	0,13	<b>0,00</b>	0,13	<b>0,00</b>
<b>20/1</b>	2670	2018	2998	2088	1990	34,17	<b>1,40</b>	50,65	4,92
<b>20/2</b>	1018	803	1072	825	784	29,80	<b>2,42</b>	36,73	5,22
<b>20/3</b>	4880	4880	4880	4880	4880	<b>0,00</b>	<b>0,00</b>	<b>0,00</b>	<b>0,00</b>
<b>25/1</b>	2108	1687	2300	1735	1664	26,68	<b>1,38</b>	38,22	4,26
<b>25/2</b>	1540	980	1668	1049	944	63,13	<b>3,81</b>	76,69	11,12
<b>25/3</b>	5520	5492	5536	5492	5492	0,50	<b>0,00</b>	0,80	<b>0,00</b>
<b>30/1</b>	2892	2290	3388	2412	2262	27,85	<b>1,23</b>	49,77	6,63
<b>30/2</b>	1870	1076	1930	1161	1030	81,55	<b>4,46</b>	87,37	12,72
<b>30/3</b>	6384	6380	6384	6380	6380	0,06	<b>0,00</b>	0,06	<b>0,00</b>

Para o ABC com 18 combinações de regras, o valor de GAP obtido constata uma diferença percentual baixa se comparada com o mesmo algoritmo aplicado com 4 combinações de regras. Em alguns casos, o ABC encontra diferença semelhante ao AG com 18 combinações de regras, mas na maioria dos casos há maior diferença percentual no ABC do que no AG com 18 combinações de regras. Por esta razão, e pelas análises anteriores, entende-se que o AG com 18 combinações de regras fornece os melhores resultados para o problema de carregamento de navios-contêineres.

A fim de exemplificar a solução obtida na forma de combinações de regras para cada problema teste, apresenta-se a sequência de combinações de regras que compõem a solução como informação complementar ao valor da FO, que contabiliza o número total de movimentos do plano de carga proposto. Os resultados contidos na Tabela 34 referem-se a apenas uma execução para cada problema teste. Nestes experimentos, as simulações foram realizadas sem interessar o número de movimentos, mas apenas a aplicação das combinações de regras nos portos. O algoritmo empregado foi o AG através do uso de 18 combinações de regras.

Tabela 34 - AG - 18 combinações de regras

Inst.	Combinações de regras aplicadas a cada porto
10/1	9 8 16 17 17 16 1 1 13
10/2	17 7 17 11 12 4 9 18 15
10/3	9 9 13 7 14 6 15 10 7
15/1	9 17 11 16 11 17 7 9 16 5 9 10 17 6
15/2	9 18 12 8 3 12 17 5 17 15 15 15 13 1
15/3	11 14 7 17 4 5 5 13 13 7 6 4 6 11
20/1	18 9 17 18 10 12 18 17 15 16 18 3 18 7 1 11 16 9 7
20/2	12 10 17 17 18 13 12 16 17 5 15 16 9 10 18 7 16 17 16
20/3	4 8 9 11 4 6 3 13 14 15 3 11 18 6 4 9 10 14 2
25/1	11 5 17 4 16 18 3 10 14 17 15 16 9 14 7 1 13 17 5 17 7 16 13 7
25/2	17 11 8 12 14 14 4 3 5 16 18 3 9 5 15 7 9 17 16 16 5 3 13 9
25/3	10 3 5 14 15 9 11 6 12 5 10 5 15 3 11 6 16 7 13 11 10 6 10 17
30/1	12 18 17 14 2 9 11 8 8 14 17 1 9 17 1 2 1 3 10 17 11 14 9 5 1 1 18 3 14
30/2	9 10 5 10 18 18 11 16 1 3 16 15 11 5 18 17 13 16 7 1 3 16 11 7 4 13 15 15 7
30/3	3 6 3 6 6 17 12 10 7 8 14 13 18 14 7 17 1 18 11 16 3 15 18 8 9 10 18 2 12

Cada uma das instâncias tem seu resultado com uma quantidade de combinações, a serem aplicadas nos respectivos portos da rota, uma unidade a menos que a quantidade de portos. Isto é devido ao fato de no último porto somente acontecer descarregamento, deixando o navio totalmente vazio e não sendo necessária nenhuma regra para o último porto. Desta forma, para todos os percursos com  $p$  portos sempre haverá  $p - 1$  regras.

Como se pode observar, o valor da maioria das combinações de regras empregadas nas soluções obtidas nas simulações são combinações acima de 4, revelando a importância das novas regras aplicadas e de suas combinações. A partir dos resultados apresentados acima, é possível verificar a real aplicação das novas regras pelo algoritmo, elegendo tais regras após 1000 gerações do algoritmo. Efetuando-se uma análise mais aprofundada, através das combinações de regras que mais são selecionadas pelo algoritmo, é possível identificar as regras que mais auxiliam na busca dos melhores resultados, isso é importante para trabalhos futuros, a fim de aprimorar as regras criadas neste estudo.

## 6.4 Testes Adicionais

Os testes adicionais foram realizados com o intuito de testar o algoritmo para navios-contêineres de capacidade maior do que as testadas nos experimentos anteriores. Os experimentos foram realizados utilizando-se o AG com 18 combinações de regras, visto que apresentou desempenho melhor nos experimentos anteriores. Foi escolhida a mesma configuração do Cenário 1, com cruzamento uniforme e mutação uniforme, e com taxa de cruzamento de 90% e taxa de mutação de 15%, devido a esta configuração ter apresentado os melhores resultados nas simulações anteriores. Para estes experimentos, foi utilizado um computador com processador Dual Core 2.00GHz, memória RAM de 3GB e Sistema Operacional Windows 7.

### 6.4.1 Testes com 1000 contêineres

Nos testes anteriores, a capacidade do navio-contêiner foi de 300 contêineres, ou seja, um navio com seis linhas ( $R=6$ ) e cinquenta colunas ( $C=50$ ), conforme especificado anteriormente. Os testes, que são apresentados nesta seção, foram realizados para um navio com capacidade de 1000 contêineres, ou seja, um navio com dez linhas ( $R=10$ ) e cem colunas ( $C=100$ ).

Para realização destes testes, foi necessário criar novas instâncias. Foi utilizada a ferramenta Matlab, através de uma função criada especificamente, sendo geradas 15 novas instâncias de forma aleatória. Da mesma forma que as instâncias testadas anteriormente, as instâncias são classificadas de acordo com o tipo de Matriz de Transporte, Mista (1), Longa (2) e Curta (3) distância.

Após geradas as instâncias, foi calculado o número mínimo de movimentos (solução ótima) para cada instância, que corresponderá ao resultado esperado nas simulações. Na Tabela 35, são mostrados os valores calculados.

**Tabela 35 - Movimentos mínimos para 1000 contêineres**

<b>Navio-Contêiner com capacidade de 1000 contêineres</b>			
	<b>Tipo de Matriz de Transporte</b>		
<b>Portos</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>10</b>	4.494	2.408	7.508
<b>15</b>	4.318	2.516	11.336
<b>20</b>	5.770	3.106	15.228
<b>25</b>	6.296	3.368	18.446
<b>30</b>	7.588	1.384	22.344

Conforme se pode observar, as soluções ótimas, representadas na Tabela 28, possuem valores elevados de movimentos mínimos, o que reforça a complexidade do problema à medida que a capacidade do navio aumenta. Em consequência, o tempo gasto para resolver este problema também é maior comparado ao de 300 contêineres.

Em média, o tempo gasto para gerar cada resultado com critério de parada baseado no número máximo de 1000 gerações variou entre 4 a 11 minutos.

A Tabela 36 apresenta os valores médios de FO para cada instância avaliada, bem como seu respectivo desvio-padrão. No primeiro caso, o problema com 10 portos e matriz tipo 1, encontrou-se a média de 4511,16 e o resultado esperado era de 4494,00 movimentos, tendo sido obtido um GAP de 0,38%. Este valor comprova uma diferença percentual baixa, indicando a obtenção de um resultado de boa qualidade. No problema com 15 portos, para o mesmo tipo de matriz, tem-se 4338,96 movimentos contra 4318,00 movimentos esperados, apresentando um GAP de 0,48%.

**Tabela 36 - Resultados Médios do AG para 1000 contêineres**

	1		2		3	
	FO	$\sigma$	FO	$\sigma$	FO	$\sigma$
<b>10</b>	4511,16	1,55	2444,00	0,00	7508,73	1,55
<b>15</b>	4338,96	10,83	2533,00	0,00	11336,00	0,00
<b>20</b>	5863,7	20,85	3238,86	14,01	15228,00	0,00
<b>25</b>	6388,86	19,11	3639,11	34,85	18446,00	0,00
<b>30</b>	7755,93	23,83	2824,30	13,95	22344,00	0,00

Seguindo, no problema com 20 portos encontrou-se 5863,70 movimentos contra 5770,00 esperados, contabilizando um GAP de 1,62%. Para 25 Portos tem-se 6388,86 movimentos para um valor ótimo de 6296,00 e apresentando um GAP de 1,47%. Para 30 portos a média é de 7755,93 contra 7588,00 movimentos esperados e GAP de 2,21%. Observa-se através do cálculo do GAP que, à medida que o número de portos aumenta, existe maior dificuldade do algoritmo em encontrar resultados próximos aos resultados ótimos.

Analisando os resultados obtidos, quando considerada a matriz tipo 2, foram obtidos valores médios para os casos com 10 e 15 portos com desvio-padrão nulo, apesar de não ter sido encontrado o ótimo global. Para estes casos, acredita-se que o algoritmo tenha ficado preso em um mínimo local e seja necessária uma técnica de diversificação das soluções mais eficiente. O GAP para o resultado de 2440,00, para o caso com 10 portos, contra o valor esperado de 2408,00, é de 1,32%. Para o caso com 15 portos, o GAP possui valor menor que 0,67%, indicando que o algoritmo encontrou um resultado mais próximo do ótimo do que para 10 portos. Para o caso com 20 portos, têm-se 3238,86 e o esperado era de 3106,00 movimentos, resultando em um GAP mais elevado de 4,27%. Para o caso com 25 portos,

encontrou-se 3639,11 movimentos com um elevado desvio-padrão de 34,85 e GAP de 8,04% e com 30 portos o GAP é mais elevado ainda, contabilizando 104,06%.

Observando o caso da matriz tipo 3 para 10 portos, o GAP é de 0,009%, ou seja, muito próximo da solução esperada. Para os problemas com 15, 20, 25 e 30 portos é possível encontrar a solução esperada com desvio-padrão nulo. Isto comprova que, mesmo para problemas de dimensão maior, o Algoritmo Genético aplicado com dezoito combinações de regras se mostra muito eficaz.

A fim de complementar os resultados médios, comentados acima, a Tabela 37 apresenta uma comparação entre os melhores resultados obtidos com a metaheurística AG e a solução ótima, para cada problema teste avaliado. Para esta comparação é empregada a medida GAP, já utilizada anteriormente.

**Tabela 37 - GAP para AG 1000 contêineres**

Inst.	AG 18R	ÓTIMO	GAP (%)
	FO	FO	
<b>10/1</b>	4510	4494	0,35
<b>10/2</b>	2444	2408	1,47
<b>10/3</b>	7508	7508	0,00
<b>15/1</b>	4329	4318	0,25
<b>15/2</b>	2533	2516	0,67
<b>15/3</b>	11336	11336	0,00
<b>20/1</b>	5830	5770	1,03
<b>20/2</b>	3221	3106	3,70
<b>20/3</b>	15228	15228	<b>0,00</b>
<b>25/1</b>	6350	6296	0,85
<b>25/2</b>	3590	3368	6,59
<b>25/3</b>	18446	18446	0,00
<b>30/1</b>	7712	7588	1,63
<b>30/2</b>	2809	1384	102,96
<b>30/3</b>	22344	22344	0,00

Através de uma análise dos resultados apresentados na Tabela 37, observa-se que o AG forneceu resultados de qualidade satisfatória, tendo alcançado valores de GAP inferiores a 1%, com exceção para os problemas teste onde emprega-se a Matriz tipo 2.

#### 6.4.2 O caso com 5000 contêineres

Esta seção é destinada a apresentar os resultados obtidos para um problema teste considerando um navio com capacidade de 5000 contêineres. Visto que o AG teve bom desempenho nos experimentos com 1000 contêineres, foi criada uma instância para testar o mesmo algoritmo num navio-contêiner de capacidade maior, mais próximo da dimensão dos

navios que comumente operam em terminais portuários de contêineres. A instância foi gerada da mesma forma que descrito no caso com 1000 contêineres e o computador utilizado para teste também manteve-se o mesmo. Quanto aos parâmetros, estes foram mantidos os mesmos do problema anterior.

A instância gerada foi para um percurso de 10 portos e tipo de Matriz de distância mista. Foram realizadas 30 execuções deste algoritmo, que obteve a média de 13597,35 movimentos e desvio padrão de 21,76. O resultado ótimo para esta instância contabilizou 13120,00 movimentos e o GAP correspondente é de 3,63%. Logo, a média encontrada se mostrou muito próxima do resultado esperado, considerando que a complexidade do problema é maior que nos demais problemas com capacidades menores. Entre as 30 execuções, o melhor resultado encontrado foi de 13566,00 movimentos, apresentando um GAP de 3,39%.

O AG mostrou-se apropriado para resolução deste problema, não perdendo eficiência quando empregado em problemas envolvendo navios de grande porte. Apesar de sua eficiência, quanto maior a capacidade, e conseqüentemente a complexidade do problema, maior também o seu tempo de execução. Para este teste de um navio com capacidade para 5000 contêineres, foi gasto em média 27 minutos.

## 7 CONSIDERAÇÕES FINAIS

O problema de carregamento de navios-contêineres é de fato importante e merece atenção. Este problema influencia na eficiência de um porto, que é um fator essencial para a melhora de sua atratividade e operacionalidade. Quando realizada a revisão bibliográfica, percebeu-se que o problema ainda carecia de técnicas de solução satisfatórias.

Neste sentido, este trabalho buscou aprimorar a codificação por regras aplicada no trabalho de Azevedo (2010) para a minimização do remanejamento de contêineres, em operações de carga e descarga em terminais portuários. Para isso, foram criadas novas regras de carregamento e descarregamento de contêineres, que foram aplicadas às metaheurísticas Algoritmo Genético e Algoritmo Enxame de Abelhas. A criação de novas regras e o emprego da metaheurística Enxame de Abelhas são destacadas como as principais contribuições desta pesquisa. As novas regras possibilitaram uma melhora significativa nos resultados e o Enxame de Abelhas mostrou-se como uma nova técnica possível de ser aplicada para solução do problema de carregamento de navios-contêineres, o que ainda não havia sido encontrado na literatura.

O objetivo geral deste trabalho foi de aplicar as duas metaheurísticas (AG e ABC) na resolução do problema de minimização de movimentos de contêineres em navios, com o intuito de encontrar resultados semelhantes aos encontrados na literatura. A partir da análise dos resultados, foi possível encontrar resultados semelhantes, idênticos e melhores que os resultados da literatura.

Analisando o Algoritmo Genético com 4 combinações de regras, análogo ao algoritmo de Azevedo (2010), observou-se que a aplicação de operadores genéticos diferentes aprimoraram os resultados do AG deste estudo em relação ao AG comparado. Já o ABC aplicado com 4 combinações de regras, se comparado ao AG de Azevedo (2010), teve um desempenho pior, mas o algoritmo não foi aprimorado neste estudo, uma vez que o objetivo principal foi apenas o seu desenvolvimento para aplicação ao problema de navios-contêineres.

A partir das dificuldades encontradas nos testes iniciais, foram criadas quatro novas regras de carregamento e uma nova regra de descarregamento de contêineres, a fim de avaliar a sua influência quando empregadas em ambos os algoritmos propostos. Através da realização de experimentos de sintonia para o AG, os melhores resultados foram obtidos para o Cenário 1, definido pelo emprego dos operadores de cruzamento uniforme e mutação uniforme, tendo sido identificado os valores de 90% para a taxa de cruzamento e 15% para a taxa de mutação como ideais.

Quando realizados os experimentos de sintonia para o ABC, o Cenário 2 foi identificado como a melhor configuração, sendo definido pelo uso da estrutura de vizinhança por troca e o valor de 100 para a quantidade de fontes de alimento.

Os melhores resultados obtidos com a melhor configuração de cada algoritmo proposto são mostrados na Tabela 37.

Tabela 38 - Melhor Cenário de cada Metaheurística

Portos	Tipo de matriz	Resultado Ótimo	Resultado médio do AG (melhor cenário)	Resultado Médio do ABC (melhor cenário)
10	1	1322,00	1335,03	1354,75
	2	750,00	760,00	771,77
	3	2282,00	2282,00	2282,00
15	1	1580,00	1615,40	1657,81
	2	778,00	789,16	813,66
	3	3024,00	3024,00	3024,00
20	1	1990,00	2040,76	2211,00
	2	784,00	813,00	872,30
	3	4880,00	4880,00	4880,00
25	1	1664,00	1696,60	1781,75
	2	944,00	990,73	1062,00
	3	5492,00	5492,00	5492,60
30	1	2262,00	2311,43	2543,90
	2	1030,00	1094,50	1247,70
	3	63805,00	6380,00	6380,00

Através dos resultados apresentados na Tabela 37, observa-se que o AG, empregado com 18 combinações de regras, mostrou-se melhor que o ABC, também empregado com 18 combinações de regras. Além disso, o uso de 18 combinações de regras proporcionou a obtenção de melhores soluções se comparadas com aquelas obtidas quando do uso de apenas 4 combinações de regra, para ambas as metaheurísticas propostas neste trabalho. Desta forma, pode-se concluir que um número maior de regras produziu melhores resultados.

Em uma análise superficial a respeito das regras mais utilizadas pelos algoritmos, nota-se um uso mais intenso das novas regras criadas, em especial a regra de descarregamento criada nesta pesquisa. Nos experimentos realizados com as novas regras, cujos resultados são apresentados na Tabela 37, nota-se de forma clara uma melhora relevante, proporcionando a

obtenção de resultados muito próximos dos ótimos globais esperados, e em alguns casos chegando a esta solução ótima.

Constatados os bons resultados encontrados através do uso das novas regras, foram realizados testes com navios de maior capacidade, a fim de avaliar os limites de eficiência e eficácia dos métodos propostos. Foram observados resultados satisfatórios mesmo considerando um problema mais complexo. Foram avaliados problemas teste com as capacidades do navio de 1000 e 5000 contêineres. Em ambos os casos, empregou-se o AG tendo sido obtidos resultados de qualidade e muito próximos do ótimo global, mesmo operando em problemas de grande dimensão.

Em relação ao tempo computacional, o AG despendeu um tempo aceitável de simulação, mesmo quando aplicado a problemas de capacidade maior, já o ABC exigiu um tempo computacional elevado se comparado ao tempo exigido pelo AG. Por esta razão, e pelos resultados encontrados, o algoritmo que obteve melhor desempenho foi o AG.

Após a análise dos experimentos realizados e os resultados obtidos, pode-se afirmar que ainda existem vários aspectos que devem ser avaliados e aprimorados, visando a obtenção de resultados ainda melhores para o problema de carregamento e descarregamento de navios-contêineres. A seguir são elencados alguns dos aspectos que podem ser explorados em trabalhos futuros:

- Criação de mais regras com a finalidade de obter melhores resultados, a partir de um estudo minucioso em relação às arrumações possíveis e que contribuiriam para o aprimoramento das soluções. Este estudo pode ser realizado através de uma análise profunda em relação às regras mais utilizadas pelo algoritmo;
- Avaliação das metaheurísticas considerando navios com capacidades ainda maiores, uma vez que atualmente existem navios capazes de transportar um pouco mais de 15000 contêineres;
- Avaliar o desempenho da metaheurística ABC, quando empregados os mesmos operadores utilizados no AG, ou ainda com outros operadores de vizinhança ou uma combinação deles;
- Aplicar a estratégia de codificação por regras para solução de problemas de carregamento de navios-contêineres sujeitos a outras restrições, como por exemplo, restrições relativas à estabilidade do navio.

## REFERÊNCIAS BIBLIOGRÁFICAS

AKAY B, KARABOGA D. Artificial bee colony algorithm for large-scale problems and engineering design optimization. *J Intell Manuf*, 2012. Disponível em: <http://www.springerlink.com/content/h110150117074362/>.

AKBARI R, Hedayatzadeh R, Ziarati K, Hassanizadeh B. A multi-objective artificial bee colony algorithm. *Swarm Evol Comput*, v. 2, p. 39-52, 2011

AMBROSINO, D.; SCIOMACHEN A.; TANFANI, E. A decomposition heuristics for the container ship stowage Problem. *J. Heuristics*, v. 12, p. 211- 233, 2006.

AVRIEL, M.; PENN, M. Exact and approximate solutions of the container ship stowage problem. *Computers and Industrial Engineering*, v. 25, p. 271– 274, 1993

AVRIEL, M., M. PENN, N.; SHPIRER, N.; WITTEBOON, S. Stowage Planning for Container Ships to Reduce the Number of Shifts. *Annals of Operations Research*, v.76, p. 55– 71, 1998.

AVRIEL, M., M. PENN, and N. Shpirer. Container Ship Stowage Problem: Complexity and Connection to the Coloring of Circle Graphs. *Discrete Applied Mathematics*, v.103, p.271– 279, 2000.

AZEVEDO, Anibal T. Instâncias empregadas nos artigos do SIMPEP, 2009. Disponível em: <https://sites.google.com/site/projetonavio/programas-dados>.

AZEVEDO, Anibal T.; SOBRAL, C.; LIMA, F. Resolução do Problema de Carregamento e Descarregamento de Contêineres em Terminais Portuários via Beam Search. XVI SIMPEP, 2006.

AZEVEDO, Anibal T; RIBEIRO, C. M.; DEUS, N. M. R. Resolução do Problema de Carregamento e Descarregamento de Contêineres em Terminais Portuários via Algoritmo Genético. *Revista INGEPRO – Inovação, Gestão e Produção*, v. 2, p. 38-51, 2010.

BOTTER, R.C; BRINATI, M.A. Stowage Container Planning: A Model for Getting an Optimal Solution. *IFIP Transactions B (Applications in Techn.)* v. 5, p. 217–229, 1992.

CATARINA, Adair Santa. Um Algoritmo Genético Com Representação Explícita De Relacionamentos Espaciais Para Modelagem Sócio-Ambiental (Tese de Doutorado). INPE, 2006.

CHAN, K. L. Mak, Y. S. Wong and E T. S. A Genetic Algorithm for facility layout problems. The University of Hong Kong, 1998.

CHAVES, Antonio Augusto. Metaheurísticas. Encontro Regional de Pesquisa Operacional da Região Sudeste, UNESP, 2010.

CHEN, C.Y.; HSIEH, T.W. A Time-Space network model for the berth allocation problem. 19th IFIP TC7 Conference on System Modeling and Optimization, Cambridge, UK, 1999.

CHIWIACOWSKY, Leonardo D. Método variacional e algoritmo genético em identificação de danos estruturais (Tese de doutorado). INPE, 2005.

CHIWIACOWSKY, Leonardo D; MONTEIRO, Rodrigo dos Santos. Uma aplicação da metaheurística algoritmo genético ao Problema de carregamento de navios-contêineres. Pesquisa Operacional para o Desenvolvimento, v.4, n.1, p.52-64, 2011.

CHOO, Shawn. The Crane Split and Sequencing Problem with Clearance and Yard Congestion Constraints in Container Terminal Ports (Dissertação de Mestrado em Computação). Massachusetts Institute of Technology, 2006.

CORDENONSI, Andre Zanki. Ambientes, Objetos e Dialogicidade: Uma estratégia de Ensino Superior em Heurísticas e Metaheurísticas (Tese de doutorado), 2008.

DIAS, João da Sila; GROSKO, Ana Paula; GORSKI, José Roberto. Algoritmo Genético: Revisão Histórica E Exemplificação, 2006.

DRIAS, H.; Sadeg, S.; Yahi, S. Cooperative bees swarm for solving the maximum weighted satisfiability problem, computational intelligence and bioinspired Systems. Proceedings of the 8th International Workshop on Artificial Neural Networks, 2005.

DUBROVSKY, O; LEVITIN, G; PENN, M. A Genetic Algorithm with a compact solution encoding for the container ship stowage Problem. Journal of Heuristics, v.8 p. 585-589, 2002.

FILITTO, Danilo. Algoritmos Genéticos: Uma visão explanatória. Revista Saber Acadêmico, v.6, 136-143, 2008.

GLOVER, F. Future paths for integer programming and links to artificial intelligence. Computers and Operations Research, Oxford, UK, v. 13, p. 533–549, 1986.

GOLDBERG, David E. Genetic Algorithms: in Search, Optimization & Machine Learning, Addison Wesley, 1999.

GÜMUS, Mehmet; KAMINSKY, P; TIEMROTH, E.; AYIK, M. A Multi-stage Decomposition Heuristic for the Container Stowage Problem, University Montreal, University of California, 2008.

HÜE, Xavier. Genetic Algorithms for Optimisation: Background and Applications. Version 1.0, University of Edinburgh, 1997.

IMAI, A., Sasaki, K., Nishimura, E. and Papadimitriou, S. Multi-objective simultaneous stowage and load planning for a container ship with container rehandle in yard stacks, *European Journal of Operational Research* 171: 373–389, 2006.

KAISAR, Evangelos I. A stowage planning model for multiport container transportation (Doctor of Philosophy), Univ. of Maryland, Maryland, USA, 2006.

KARABOGA, D. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

KARABOGA D, Gorkemli B (2011) A combinatorial artificial bee colony algorithm for traveling salesman problem. In: 2011 international symposium on innovations in intelligent systems and applications (INISTA), p. 50–53, 2011.

KARABOGA, D; AKAY, B. Artificial Bee Colony (ABC), Harmony Search and Bees Algorithms on Numerical Optimization, Erciyes University, Dept of Computer Engineering, 2010.

KIM, K. H. and Park, Y. M. A crane scheduling method for port container terminals, *European Journal of Operational Research* 156: 752–768, 2004.

KIM, K. H; MOON. K. H. Berth scheduling by simulated annealing. *Transportation Research-B* 37, p.541-560, 2003.

LEE, D. H., Cao, Z. and Meng, Q. Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm, *International Journal of Production Economics* 107: 115–124, 2007.

LEE, L. H., Chew, E. P., Tan, K. C. and Han, Y. An optimization model for storage yard management in transshipment hubs, *OR Spectrum* 28: 539–561, 2006.

LI WH, Li WJ, Yang Y, Liao HQ, Li JL, Zheng XP. Artificial bee colony algorithm for traveling salesman problem. *Adv Mater Res* 314(316):2191–2196, 2011.

LORENA, Luiz A.N.; MAURI, G.; OLIVEIRA, R. Abordagem da Metaheurística Clustering Search com Simulated Annealing para o Problema de Alocação de Berços de Navios – INPE, X WORCAP, 2010.

LUCIC, P.; Teodorovic, D. Bee System: Modeling Combinatorial Optimization Transportation Engineering Problems by Swarm Intelligence. Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis, p. 441–445, 2001.

MALAQUIAS, Neli Gomes Lisboa. Uso dos Algoritmos Genéticos para Otimização de Rotas de Distribuição (Dissertação de Mestrado). 2006.

MARINHO, Euler Horta. Heurísticas Busca Tabu para o Problema de Programação de Tripulações de Ônibus Urbano (Dissertação de Mestrado), 2005.

MICHALEWICZ, Z. Genetic Algorithms + Data Structures = Evolution Programs, 3<sup>rd</sup> edition, Berlin: Springer-Verlag, 1996.

MITCHELL, M. An introduction to genetic algorithms. Cambridge, Massachusetts: MIT Press, 1996.

MOCCIA, L., Cordeau, J. F., Gaudioso, M. and Laporte, G. A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal, *Naval Research Logistics* 53: 45–59, 2006.

MORAIS, V.; VIEIRA, C. Matlab – Curso Completo. FCA – Editora de Informática, 2012.

NAKRANI, S.; Tovey, C. On Honey Bees and Dynamic Allocation in an Internet Server Colony. Proceedings of 2nd International Workshop on The Mathematics and Algorithms of Social Insects, Atlanta, Georgia, USA, 2003.

NG, W. C. and Mak, K. L. Yard crane scheduling in port container terminals, *Applied Mathematical Modelling* 29: 263–276, 2005.

ÖZBAKIR, Lale; BAYKASOĞLU, Adil; TAPKAN, Pinar. Bees algorithm for generalized assignment problem. Turkey, 2009. Disponível em:  
<<http://www.sciencedirect.com/science/article/pii/S0096300309010078>>.

PEDREIRA, Adriana F. Os Recentes Avanços da Multimodalidade no Brasil (Dissertação de Mestrado)- PUC RIO, 2006.

PHAM, D.T.; Darwish, A.H.; Eldukhr, E.; Otri, S. Using the bees algorithm to tune a fuzzy logic controller for a robot gymnasta. Proceedings of the Innovative Production Machines and Systems Virtual Conference, 2007.

PHAM, D.T.; Muhamad, Z.; Mahmuddin, M.; Ghanbarzadeh, A.; Koç, E.; Otri, S. Using the bees algorithm to optimize a support vector machine for wood defect classification. Proceedings of the Innovative Production Machines and Systems Virtual Conference, 2007.

PHAM, D.T; GHANBARZADEH, A; E. Koç, S. Otri, S. Rahim, M. Zaidi. The Bees Algorithm – A Novel Tool for Complex Optimisation Problems. Manufacturing Engineering Centre, Cardiff University, Cardiff CF24 3AA, UK, 2006.

RUBIO-LARGO I, Vega-Rodríguez M, Gmez-Pulido J, Snchez-Prez J. Tackling the static rwa problem by using a multiobjective artificial bee colony algorithm. In: Cabestany J, Rojas I, Joya G (eds) *Advances in computational intelligence. Lecture notes in computer science*, vol 6692. Springer, Berlin, p. 364–371, 2011.

SAMMARRA, M., Cordeau, J. F., Laporte, G. and Monaco, M. F. A tabu search heuristic for the quay crane scheduling problem, *Journal of Scheduling* 10: 327–336, 2007.

SCHRIJVER, Alexander. *A Course in Combinatorial Optimization*. University of Amsterdam, 2012.

SCIOMACHEN, Anna; TANFANI, Elena. A 3D-BPP approach for optimizing stowage plans and terminal productivity. *European Journal of Operational Research*, 2007.

SCIOMACHEN, Anna; TANFANI, Elena. The master bay plan problem: a solution method based on its connection to the three-dimensional bin packing problem, 2004.

SERAPIÃO, Adriane Beatriz de Souza . *Fundamentos de otimização por inteligência de enxames: Uma visão geral*. Unesp, *Revista Controle & Automação*, v.20, n.3, p.271-304, 2009.

SIKORA, Gustavo; G, Eduardo. *Colônia Artificial de Abelhas – Um Algoritmo de Otimização Baseado em Enxame de Abelhas e Uma Aplicação no Treinamento de Redes Neurais para o Problema XOR*, 2010.

SZETO, W.Y.; Yongzhong Wub, Sin C. Ho. An artificial bee colony algorithm for the capacitated vehicle routing problem, *European Journal of Operational Research*, p.126-135, 2010.

TEODOROVIC, D.; Dell’Orco, M. Bee colony optimization – a cooperative learning approach to complex transportation problems. *Proceedings of the 10th EWGT Meeting and 16th Mini-EURO Conference*, 2005.

VACCA, Illaria, BIERLAIRE, Michel; SALANI, Matteo. *Optimization at Container Terminals: Status, Trends and Perspectives*. Swiss Transport Research Conference, 2007.

VIS, I. F. A., de Koster, R. and Savelsbergh, M. Minimum vehicle fleet size under time-window constraints at a container terminal, *Transportation Science* v.39, 249–260, 2005.

WILSON, I.D; ROACH, P.A. *Principles of Combinatorial Optimization Applied to Container-Ship Stowage Planning*. *Journal of Heuristics*, v.5 p.403-418, 1999.

WILSON, I.D;ROACH,P.A. Container stowage planning: a methodology for generating computerized solutions, *Journal of the Operational Research Society*, v.51, p.1248-1255, 2000.

YANG, Xin-She. *Nature-Inspired Metaheuristic Algorithms Second Edition*, University of Cambridge, 2010.

ZHANG Y,Wu L,Wang S. Magnetic resonance brain image classification by an improved artificial bee colony algorithm. *Prog Electromagn Res-PIER* 116:65–79, 2011.