



Programa Interdisciplinar de Pós-Graduação em

# **Computação Aplicada**

**Mestrado Acadêmico**

Joel da Silva Gonçalves Júnior

Abordagem metaheurística híbrida para otimização do  
planejamento de estiva de navios porta-contêineres

São Leopoldo, 2016

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

JOEL DA SILVA GONÇALVES JÚNIOR

Abordagem metaheurística híbrida para otimização do planejamento  
de estiva de navios porta-contêineres

São Leopoldo

2016

Joel da Silva Gonçalves Júnior

Abordagem metaheurística híbrida para otimização do planejamento  
de estiva de navios porta-contêineres

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

2016

G643a      Gonçalves Júnior, Joel da Silva.  
Abordagem metaheurística híbrida para otimização do planejamento de estiva de navios porta-contêineres / Joel da Silva Gonçalves Júnior. – 2016.  
102 f. : il. ; 30 cm.

Dissertação (mestrado) – Universidade do Vale do Rio dos Sinos, Programa Interdisciplinar de Pós-Graduação em Computação Aplicada, 2016.

“Orientador: Prof. Dr. Leonardo Dagnino Chiwiacowsky.”

1. Plano de estiva 2. Navio porta-contêineres. 3. Algoritmo genético. 4. Busca tabu. 5. Abordagem híbrida. 6. Otimização combinatória. I. Título.

CDU 004

Joel da Silva Gonçalves Júnior

## Abordagem metaheurística híbrida para otimização do planejamento de estiva de navios porta-contêineres

Dissertação apresentada à Universidade do Vale do Rio dos Sinos – Unisinos, como requisito parcial para obtenção do título de Mestre em Computação Aplicada.

Aprovado em 07 de março de 2016.

### BANCA EXAMINADORA

---

Prof. Dr. Anibal Tavares de Azevedo – UNICAMP

---

Prof. Dr. Arthur Tórgo Gómez – UNISINOS

---

Prof. Dr. Leonardo Dagnino Chiwiacowsky – UNISINOS

Prof. Dr. Leonardo Dagnino Chiwiacowsky (Orientador)

Visto e permitida a impressão

São Leopoldo,

Prof. Dr. Sandro José Rigo

Coordenador PPG em Computação Aplicada

*Aos meus pais Joel e Tânia, pelos ensinamentos, valores e incentivos.*

*A minha querida noiva Dafni, pelo apoio, amor e companheirismo.*

*“Viver é aventurar-se no novo, alçar voos sobre  
os próprios temores, pôr-se à prova... Um  
eterno recomeço.”*

*(Eduardo Martins)*

## **AGRADECIMENTOS**

Agradeço primeiramente aos meus pais, Tânia e Joel, por todos os ensinamentos e valores que recebi.

A Deus, pela vida e pela oportunidade.

Agradeço às minhas irmãs, Taiene e Nicole, pela parceria desta vida.

Aos meus queridos afilhados, Vitor e Tiago, pelo tempo que não estive presente.

À minha companheira, Dáfini, pelo suporte e pelo carinho que tornaram este caminho alegre e prazeroso.

Ao Prof. Dr. Leonardo, pelo apoio e orientação na pesquisa.

Aos professores, pelos ensinamentos no decorrer deste programa.

Aos amigos e colegas, pela troca de experiências, pelas boas conversas e pelas relações que não de se eternizar.

À CAPES, pelo apoio financeiro.

## RESUMO

O transporte marítimo mercante desempenha um papel fundamental para a economia de uma nação, ligando a produção ao consumo. No cenário de expansão do transporte marítimo, a utilização de contêineres para organização das cargas confere maior facilidade, segurança e rapidez ao transporte, aumentando, assim, a produtividade dos terminais e dos navios. No entanto, a operação de navios porta-contêineres possui limitações de movimentação e de estabilidade que impactam no custo operacional de um terminal portuário. Como os guindastes só podem acessar as pilhas de contêineres a partir do topo, a realização de remoções desnecessárias de contêineres bloqueantes gera um custo adicional de movimentação e de tempo nas operações de carga e descarga. Desta forma, faz-se necessária a elaboração de um plano de estiva eficiente para estas atividades, minimizando tanto os remanejamentos quanto a instabilidade da embarcação. Este estudo propõe uma abordagem híbrida, elaborada através da combinação das metaheurísticas Algoritmo Genético e Busca Tabu, utilizando a codificação da solução baseada em regras, a fim de elaborar uma ferramenta computacional que faça a gestão do número de remanejamentos e da instabilidade da embarcação, que são objetivos conflitantes. Nos experimentos, as metaheurísticas puras foram comparadas ao algoritmo híbrido e os resultados comprovaram que a aplicação hibridizada apresenta uma eficiência maior do que as metaheurísticas puras. As diferentes configurações de regras assumidas mostraram que a proposta de um número maior de regras, em complemento àquelas propostas na literatura, implica em melhores resultados. Através da aplicação da abordagem com múltiplos objetivos, foi possível observar a importância de considerar a movimentação e a estabilidade no plano de estiva. Com os resultados obtidos, demonstrou-se que o uso da abordagem proposta gera soluções melhores que as encontradas até o momento na literatura.

**Palavras-Chave:** Plano de Estiva. Navio Porta-contêineres. Algoritmo Genético. Busca Tabu. Abordagem Híbrida. Otimização Combinatória.

## ABSTRACT

The merchant shipping perform a fundamental role in the economy of a nation, by linking production to consumption. In shipping expansion scenario, the use of containers for cargo organizing provides greater facility, safety and velocity, thus increasing the productivity of terminals and ships. However, the use of container ships has handling and stability limitations that affect the operating cost of a port terminal. As the cranes can only access the container stacks from the top, carrying out unnecessary removals of blocking containers generates an additional cost of handling and time in loading and unloading operations. Thus, it is necessary to elaborate an efficient stowage plan for loading and unloading operations, minimizing both the shifting and the instability of the vessel. This study proposes an hybrid approach developed by the combination of Genetic Algorithms and Tabu Search metaheuristics, using a rules-based encoding for solution representation, in order to create a computational tool that manage both the rehandling and instability, which are conflicting. In the experiments, pure metaheuristics were compared to the hybrid algorithm and the results demonstrate that the hybridization presents greater efficiency than the pure metaheuristics. The different rules configuration have proven that the proposal of a greater number of rules, in addition to those proposed in the literature, implies better results. The application of a multiple objectives approach has proven the importance of considering the handling and stability in the stowage plan. With the results, it was showed that the use of the proposed approach produces better solutions than those found in the literature.

**Keywords:** Stowage Plan. Container Ship. Genetic Algorithms. Tabu Search. Hybrid Approach. Combinatorial Optimization.

## LISTA DE FIGURAS

Figura 1 - Navio porta-contêineres Triple-E .....	16
Figura 2 - Terminal portuário .....	21
Figura 3 - Estrutura celular de um navio porta-contêiner.....	23
Figura 4 – Centro de Gravidade (G), Centro de Carena (B) e Metacentro (M) .....	25
Figura 5 – Braço de endireitamento (GZ) .....	26
Figura 6 - GM e list .....	28
Figura 7 - <i>Trim</i> de uma embarcação .....	29
Figura 8 - Fluxograma do AG .....	37
Figura 9 - Ciclagem da BT .....	38
Figura 10 - Pseudocódigo da BT .....	39
Figura 11 - Estratégias de migração para Modelos <i>Island</i> e <i>Stepping-Stone</i> .....	47
Figura 12 - Matriz de Transporte T .....	60
Figura 13 – Matriz de ocupação B após aplicada regra de carregamento Rc1 .....	60
Figura 14 - Matriz de ocupação B após aplicada regra de carregamento Rc2 .....	60
Figura 15 - Matriz de ocupação B após aplicada regra de carregamento Rc3 .....	61
Figura 16 - Matriz de ocupação B após aplicada regra de carregamento Rc4 .....	61
Figura 17 - Matriz de ocupação B após aplicada regra de carregamento Rc5 .....	62
Figura 18 - Matriz de ocupação B após aplicada regra de carregamento Rc6 .....	62
Figura 19 – Matriz de ocupação B após aplicada regra de carregamento Rc7.....	63
Figura 20 - Matriz de ocupação B após aplicada regra de carregamento Rc8 .....	63
Figura 21 - Matriz de ocupação B após aplicada regra de carregamento Rc9 .....	63
Figura 22 - Matriz de ocupação B após aplicada regra de carregamento Rc10 .....	64
Figura 23 - Matriz de ocupação B após aplicada regra de carregamento Rc11 .....	64
Figura 24 - Matriz de ocupação B após aplicada regra de carregamento Rc12 .....	65
Figura 25 - Matriz de ocupação B após aplicar regra de descarregamento Rd1 .....	65
Figura 26 - Matriz de ocupação B após aplicar regra de descarregamento Rd2 .....	66
Figura 27 - Codificação do Cromossomo.....	67
Figura 28 - Operador Um Ponto de Corte .....	69
Figura 29 - Fluxograma do algoritmo híbrido AG+BT .....	72

## LISTA DE TABELAS

Tabela 1 - Lista da Movimentação de Contêineres dos principais Portos do Mundo .....	20
Tabela 2 - Explosão combinatória gerada pelo PCV.....	32
Tabela 3 - Regras utilizadas em cada porto .....	66
Tabela 4 – Resultados dos cenários de testes para sintonia do AG.....	75
Tabela 5 – Resultados dos cenários de testes para sintonia da BT .....	76
Tabela 6 – Resultados dos cenários de testes para sintonia do AG+BT .....	77
Tabela 7 – Resultados dos testes de validação dos algoritmos com 12 regras.....	78
Tabela 8 – Resultados médios dos testes com o AG para 12, 24 e 36 regras.....	80
Tabela 9 – Resultados mínimos dos testes com o AG para 12, 24 e 36 regras .....	81
Tabela 10 – Resultados médios dos testes com a BT para 12, 24 e 36 regras.....	82
Tabela 11 – Resultados mínimos dos testes com a BT para 12, 24 e 36 regras .....	83
Tabela 12 – Resultados médios dos testes com o AG+BT para 12, 24 e 36 regras .....	85
Tabela 13 – Resultados mínimos dos testes com a o AG+BT para 12, 24 e 36 regras .....	86
Tabela 14 – Comparação AG+BT com minimização de movimentos <i>versus</i> instabilidade ....	88
Tabela 15 – AG+BT com otimização de múltiplos objetivos .....	89

## LISTA DE GRÁFICOS

Gráfico 1 - <i>Speed-up</i> da metaheurística híbrida paralela.....	90
Gráfico 2 - Eficiência da metaheurística híbrida paralela .....	90
Gráfico 3 - Análise dos GAP's (%) por algoritmo .....	93
Gráfico 4 - Análise dos GAP's (%) por configuração de regras .....	93

## LISTA DE SIGLAS

AAPA	<i>American Association of Port Authorities</i>
AG	Algoritmo Genético
AGP	Algoritmo Genético Paralelo
BT	Busca Tabu
BTP	Busca Tabu Paralela
CG	Centro de Gravidade
CNT	Confederação Nacional de Transportes
FO	Função objetivo
HPC	<i>High Performance Computing</i>
IPTV	<i>Internet Protocol Television</i>
ISO	<i>International Organization for Standardization</i>
LT	Lista Tabu
NP	Classe de problemas de tempo polinomial não-determinísticos
P	Classe de problemas de tempo polinomial
PAB	Problema de Alocação de Berços
PAG	Problema de Alocação de Guindastes
PCV	Probabilidade do Caixeiro Viajante
PEG	Problema de Escalonamento de Guindastes
PM	Probabilidade de Mutação
PO	Pesquisa Operacional
PPE	Problema de Planejamento da Estiva
SA	<i>Simulated Annealing</i>
TEU	<i>Twenty-foot Equivalent Units</i>

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>16</b>
1.1 JUSTIFICATIVA .....	17
1.2 OBJETIVO.....	18
1.3 A ESTRUTURA DO TRABALHO.....	19
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>20</b>
2.1 O TRANSPORTE MARÍTIMO .....	20
2.2 O PROBLEMA DE PLANEJAMENTO DE ESTIVA (PPE).....	22
2.3 CRITÉRIO DE ESTABILIDADE .....	24
2.3.1 <i>Centro de gravidade (G)</i> .....	24
2.3.2 <i>Centro de carena (B)</i> .....	24
2.3.3 <i>Braço de endireitamento</i> .....	25
2.3.4 <i>Metacentro</i> .....	26
2.3.5 <i>Equilíbrio da embarcação</i> .....	27
2.3.6 <i>Ângulo de list</i> .....	28
2.3.7 <i>Medida de trim</i> .....	28
2.4 OTIMIZAÇÃO COMBINATÓRIA .....	29
2.4.1 <i>Teoria da Complexidade Computacional</i> .....	30
2.4.2 <i>Categorias de problemas</i> .....	31
2.4.3 <i>Técnicas de solução</i> .....	32
2.5 TÉCNICAS METAHEURÍSTICAS .....	33
2.5.1 <i>Algoritmo Genético (AG)</i> .....	34
2.5.1.1 <i>Número de gerações</i> .....	35
2.5.1.2 <i>Tamanho da população</i> .....	35
2.5.1.3 <i>Taxa de cruzamento</i> .....	36
2.5.1.4 <i>Taxa de mutação</i> .....	36
2.5.1.5 <i>Taxa de substituição</i> .....	36
2.5.1.6 <i>Critério de parada</i> .....	37
2.5.2 <i>Busca Tabu (BT)</i> .....	38
2.5.2.1 <i>Memória de curta duração</i> .....	39
2.5.2.2 <i>Memória de longa duração</i> .....	40
2.5.2.3 <i>Função de aspiração</i> .....	40
2.5.2.4 <i>Critério de parada</i> .....	41
2.5.3 <i>Algoritmos híbridos</i> .....	41
2.5.3.1 <i>Estratégias de hibridização</i> .....	42
2.5.3.2 <i>Nível de combinação</i> .....	42
2.5.3.3 <i>Ordem de execução</i> .....	43
2.5.3.4 <i>Estratégia de controle</i> .....	43

2.6	PROCESSAMENTO DE ALTO DESEMPENHO .....	43
2.6.1	<i>Paradigmas de programação paralela</i> .....	44
2.6.1.1	Mestre-Escravo .....	45
2.6.1.2	Pipeline.....	45
2.6.1.3	Divisão e Conquista .....	45
2.6.1.4	Fases Paralelas .....	46
2.6.2	<i>Métricas de desempenho</i> .....	46
2.6.3	<i>Algoritmo Genético Paralelo (AGP)</i> .....	47
2.6.3.1	Modelos de migração stepping-stone e island (ilha).....	47
2.6.4	<i>Busca Tabu Paralela (BTP)</i> .....	48
<b>3</b>	<b>REVISÃO BIBLIOGRÁFICA.....</b>	<b>49</b>
3.1	PROBLEMAS RELACIONADOS A OPERAÇÕES EM TERMINAIS PORTUÁRIOS .....	49
3.2	O PROBLEMA DE PLANEJAMENTO DE ESTIVA DE NAVIOS PORTA-CONTÊINERES.....	50
3.3	A ESTABILIDADE NO CARREGAMENTO DE NAVIOS .....	53
<b>4</b>	<b>METODOLOGIA .....</b>	<b>57</b>
4.1	MODELO MATEMÁTICO .....	57
4.2	FERRAMENTAS COMPUTACIONAIS.....	58
4.2.1	<i>Codificação por regras</i> .....	58
4.2.1.1	Regras de carregamento e de descarregamento .....	59
4.2.2	<i>A metaheurística Algoritmo Genético</i> .....	67
4.2.2.1	Codificação da solução .....	67
4.2.2.2	Operador de Seleção .....	67
4.2.2.3	Operador de Cruzamento .....	68
4.2.2.4	Operador de Mutação .....	69
4.2.2.5	Critério de Parada.....	69
4.2.3	<i>A metaheurística Busca Tabu</i> .....	70
4.2.3.1	Estrutura de vizinhança .....	70
4.2.3.2	Lista Tabu.....	70
4.2.3.3	Critério de aspiração .....	70
4.2.3.4	Critério de Parada.....	71
4.2.4	<i>Abordagem híbrida</i> .....	71
<b>5</b>	<b>EXPERIMENTOS E RESULTADOS.....</b>	<b>73</b>
5.1	TIPOS DE MATRIZES DE TRANSPORTE .....	73
5.2	PROBLEMAS-TESTE UTILIZADOS .....	73
5.3	SINTONIA DE PARÂMETROS .....	74
5.3.1	<i>Sintonia dos parâmetros do Algoritmo Genético</i> .....	74
5.3.2	<i>Sintonia dos parâmetros da Busca Tabu</i> .....	75
5.3.3	<i>Sintonia dos parâmetros do AG+BT</i> .....	76
5.4	VALIDAÇÃO DOS ALGORITMOS AG, BT E AG+BT .....	77
5.5	EXPERIMENTOS COM AG, BT E AG+BT.....	79
5.5.1	<i>Minimização da movimentação com o AG</i> .....	79
5.5.2	<i>Minimização da movimentação com a BT</i> .....	82

5.5.3 <i>Minimização da movimentação com o AG+BT</i> .....	84
5.6 MINIMIZAÇÃO DA MOVIMENTAÇÃO E DA ESTABILIDADE .....	87
5.7 ABORDAGEM DE ALTO DESEMPENHO .....	89
<b>6 CONSIDERAÇÕES FINAIS .....</b>	<b>92</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>96</b>

## 1 INTRODUÇÃO

A história do transporte marítimo mercante data dos séculos XV e XVI, quando os europeus, em virtude do domínio italiano no Mediterrâneo e do grande interesse comercial em encontrar novas rotas marítimas para as Índias, dedicaram-se às navegações nos oceanos Atlântico, Pacífico e Índico (PAZZINATO e SENISE, 1992). Outro fator que estimulou as navegações, na época, foi a necessidade de obter matéria-prima, metais preciosos e outras especiarias. Este período ficou conhecido como a Era das Grandes Navegações e Descobrimentos Marítimos (BARROS, 2007; PAZZINATO e SENISE, 1992).

O transporte marítimo sempre contribuiu para o crescimento econômico das nações. Os serviços de transporte marítimo são essenciais para que a economia e as empresas possam competir em nível mundial (SILVA, G., 1999). Em meados dos anos 1960, o processo logístico passou por uma importante mudança com o surgimento de contêineres para organização das cargas. A containerização de cargas (WILSON e ROACH, 2000) modificou a operação das empresas de navegação e aumentou a produtividade, tanto dos terminais quanto dos navios. Segundo Lacerda (2004), um navio típico de transporte de carga geral gastava 149 dias por ano atracado nos portos. Já os navios porta-contêineres gastam apenas 64 dias por ano nos portos. A Figura 1 ilustra um navio porta-contêineres da companhia de navegação Maersk.

**Figura 1 - Navio porta-contêineres Triple-E**



Fonte: <http://economia.estadao.com.br/>.

O desenvolvimento da economia mundial e do comércio internacional de mercadorias aumentou a procura de serviços pelo transporte marítimo. Com isso, o transporte de carga alocada em contêineres, desde o início da década passada (2000), subiu cerca de 150% (UNCTAD, 2009) e, segundo a Comissão das Comunidades Europeias (CCE, 2009), o número de navios deverá aumentar cerca de 29% até 2018. Devido à grande importância do transporte marítimo para a economia, o transporte de cargas tornou-se um nicho de mercado

cada vez mais competitivo, levando as companhias a voltarem sua atenção à minimização dos custos relacionados ao transporte.

Com a expansão das operações portuárias, alguns problemas ligados à produtividade dos portos tornaram-se evidentes, como a alocação de berços, atribuição e escalonamento de guindastes, entre outros. Dentre os problemas relacionados à operação portuária e marítima, destaca-se a necessidade de um planejamento avançado para o carregamento e descarregamento de navios porta-contêineres, que consiste na elaboração otimizada do plano de estiva. Considerando que existe um custo inerente à movimentação unitária de contêineres em cada porto, a minimização do número de movimentos realizados nas operações de carga e descarga reflete, diretamente, na redução dos custos relacionados a cada um dos terminais portuários que constituem a rota do navio.

O navio porta-contêineres e os portos destinados ao carregamento deste tipo de embarcação possuem características que impõem algumas restrições às operações de carga e descarga. Uma das restrições existentes é o acesso aos contêineres que se encontram abaixo do topo da pilha. Como os guindastes só podem descarregar os contêineres que se encontram no topo, quando se deseja acessar um contêiner com destino no porto atual e que se encontra abaixo de contêineres com destino nos portos seguintes, é necessário remover temporariamente estes contêineres bloqueantes para, então, descarregar o contêiner desejado. Esta movimentação é chamada de remanejamento e impacta tanto no tempo quanto no custo da operação, justificando assim a aplicação de estudos capazes de propor um plano de estiva otimizado que minimize estas ocorrências.

Além da redução das movimentações, outros critérios podem ser considerados na elaboração de um plano estiva do navio, tais como o uso da água de lastro, utilização dos equipamentos específicos de operação do terminal, o uso efetivo das escotilhas ou ainda a estabilidade do navio (Giemsch e Jellinghaus, 2003).

## 1.1 Justificativa

Conforme Pedreira (2006), o transporte marítimo tem grande parcela de contribuição para a economia e para o transporte. Este protagonismo no cenário logístico é que denota a grande importância de pesquisas relacionadas ao tema. Existem diversos trabalhos na literatura que propõem o emprego de técnicas distintas para solução do problema de planejamento de estiva de navios porta-contêineres (PPE), considerando diferentes restrições e propondo modelagens distintas (MONACO *et al.*, 2014; CARRARO *et al.*, 2013; AZEVEDO *et al.*, 2009, 2011; DUBROVSKY *et al.*, 2002; WILSON e ROACH, 2000).

Em virtude do grande volume de variáveis envolvidas e do tempo gasto no processamento para obtenção de uma solução, a literatura propõe métodos aproximados e evidencia a alternativa de abrir mão dos resultados ótimos, aplicando técnicas metaheurísticas na busca de uma solução próxima da ideal ou aceitável dentro de uma faixa de tolerância (MONACO *et al.*, 2014; CARRARO *et al.*, 2013; AZEVEDO *et al.*, 2009, 2011). No entanto,

as soluções propostas para os problemas envolvendo as operações portuárias ainda possuem margem para evolução dos resultados.

Em Azevedo *et al.* (2009), a proposta de emprego de regras de carga e descarga para a representação de um plano de estiva permitiu a geração de soluções factíveis e comprovou que a utilização das mesmas pode conferir facilidade e agilidade ao processo de solução, tendo sido obtidos resultados que foram aprimorados em Carraro *et al.* (2013), através da aplicação de novas regras de carregamento. Seguindo nesta linha, o estudo proposto por Azevedo *et al.* (2011) propôs regras de carregamento para o problema 3D, considerando, assim, a estabilidade da embarcação. Contudo, a quantidade de regras propostas no estudo citado é pequena, o que possibilita a contribuição do presente trabalho no sentido de propor novas regras de carga e descarga para o problema 3D.

O Problema do Plano de Estiva (PPE) pode ser abordado sob vários aspectos no que diz respeito às suas restrições, sendo um deles a restrição de estabilidade. A preocupação com a distribuição adequada dos pesos carregados a bordo deve ser levada em consideração na geração do plano de estiva quando se deseja evitar o adernamento do mesmo, incrementando, assim, a segurança da embarcação. Neste contexto, este estudo propõe a geração do plano de estiva com a minimização da instabilidade, através do cálculo da altura metacêntrica GM (distância geométrica entre o centro de gravidade e o metacentro), do *List* e do *Trim*, sendo este cálculo destacado como uma contribuição, já que o cálculo de estabilidade tradicional ainda não foi aplicado em conjunto com a codificação por regras de carregamento para solução do problema de estiva.

## 1.2 Objetivo

Com base no exposto, o objetivo geral deste trabalho é representado pela aplicação de uma técnica metaheurística híbrida na otimização do problema de planejamento de estiva, considerando a minimização da movimentação e da instabilidade. Os objetivos específicos desta pesquisa são:

- a) Desenvolver um Algoritmo Genético, baseado na codificação por regras, que resolva o PPE através da otimização da movimentação e da instabilidade, avaliando o mesmo quantitativamente;
- b) Desenvolver um algoritmo Busca Tabu, baseado na codificação por regras, que resolva o PPE através da otimização da movimentação e da instabilidade, avaliando o mesmo quantitativamente;
- c) Desenvolver um algoritmo híbrido baseado na codificação por regras, combinando o Algoritmo Genético com a Busca Tabu, avaliando e comparando seus resultados de otimização com os resultados das técnicas que o compõem, a fim de verificar sua maior eficiência, isto é, a obtenção de resultados mais próximo do ótimo;

- d) Avaliar os resultados obtidos por cada um dos algoritmos propostos com o emprego de um número maior de regras utilizadas na codificação da solução, comprovando a eficiência das regras propostas neste trabalho;
- e) Propor um modelo de minimização da instabilidade em conjunto com a minimização da movimentação de contêineres e comprovar a importância de atender ambos os aspectos em conjunto, elaborando, desta forma, uma ferramenta computacional capaz de fazer a gestão do número de remanejamentos e da instabilidade, que são objetivos conflitantes.

### 1.3 A Estrutura do Trabalho

Este trabalho está dividido e organizado da seguinte forma:

O Capítulo 1 consiste da introdução, onde é brevemente apresentado o problema objeto desta pesquisa, as justificativas e os objetivos do trabalho, destacando as suas contribuições.

No Capítulo 2, é apresentada a fundamentação teórica, onde é contextualizado o transporte marítimo e o problema a ser tratado por este estudo, bem como são apresentados e explanados os métodos utilizados no modelo de solução desenvolvido.

No Capítulo 3, encontra-se a revisão bibliográfica, que discute os problemas relacionados a operações portuárias, dando evidência ao problema de planejamento de estiva de navios porta-contêineres e aos esforços da comunidade científica em resolver o mesmo através da análise de alguns trabalhos publicados na literatura e as técnicas utilizadas pelos autores.

O Capítulo 4 aborda o modelo matemático adotado para o problema, bem como a codificação da solução através do emprego de regras. Em adendo, são apresentadas as ferramentas computacionais empregadas na implementação do algoritmo.

No Capítulo 5, são apresentados os problemas-teste utilizados na validação dos algoritmos, bem como os experimentos realizados e a análise de quantitativa dos resultados individuais de cada algoritmo.

Por fim, o Capítulo 6 trata das considerações finais e conclusões geradas a partir das comparações entre os resultados obtidos, comprovando hipóteses e demonstrando o cumprimento dos objetivos propostos. Também são abordadas as possibilidades relacionadas ao futuro desta pesquisa.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos necessários para o desenvolvimento desta pesquisa. Inicialmente, são contextualizados o problema e as restrições envolvidas e, depois, é realizado um estudo dos métodos de solução empregados na implementação da solução proposta.

### 2.1 O transporte marítimo

Segundo Peixoto (1977), o sistema de transportes de um país tem a função de promover a circulação de produtos, movimentação de cargas e de pessoas. Neste contexto, as operações portuárias possuem relevância estratégica para a constituição de negócios, pois criam um amplo campo de oportunidades comerciais. Na Tabela 1, é apresentada a lista de movimentação de contêineres dos principais portos do mundo, na qual se percebe que o volume de carga transportada por contêineres vem aumentando a cada ano. A unidade de medida utilizada na contagem de contêineres, conforme o padrão ISO (*International Organization for Standardization*), é o *Twenty-foot Equivalent Units* (TEU), ou Unidade Equivalente de Vinte Pés, que corresponde a um contêiner de comprimento de 20' (AMBROSINO *et al.*, 2006).

**Tabela 1 - Lista da Movimentação de Contêineres dos principais Portos do Mundo**

Ranking	Porto	País	TEUs – Milhões		
			2011	2012	2013
1º	Shanghai	China	31.739.000	32.528.200	33.617.000
2º	Singapore	Singapore	29.937.700	31.649.400	32.578.700
3º	Shenzhen	China	22.570.800	22.959.800	23.278.000
4º	Hong Kong	China	24.384.000	23.118.000	22.352.000
5º	Busan	South Korea	16.163.842	17.022.969	17.611.882
6º	Ningbo	China	14.719.200	16.782.700	17.326.800
7º	Qingdao	China	13.020.100	14.609.470	15.520.000
8º	Guangzhou	China	14.260.400	14.514.200	15.309.200
9º	Dubai Ports	United Arab Emirates	12.617.595	13.280.000	13.600.000
10º	Tianjin	China	11.587.600	12.298.300	12.996.510
11º	Rotterdam	Netherlands	11.876.920	11.865.916	11.664.195
12º	Port Kelang	Malaysia	9.435.408	10.001.496	10.350.410
13º	Kaohsiung	Taiwan	9.636.289	9.781.221	9.978.857
14º	Dalian	China	6.400.300	8.064.000	9.912.000
15º	Hamburg	Germany	9.014.165	8.889.477	9.257.358
20º	Long Beach	United States	6.061.091	6.045.662	6.730.573
38º	Santos	Brazil	2.985.922	2.961.426	3.448.879

Fonte: Elaborado pelo autor. Informações consultadas de AAPA - *American Association of Port Authorities* (<http://www.aapa-ports.org/>).

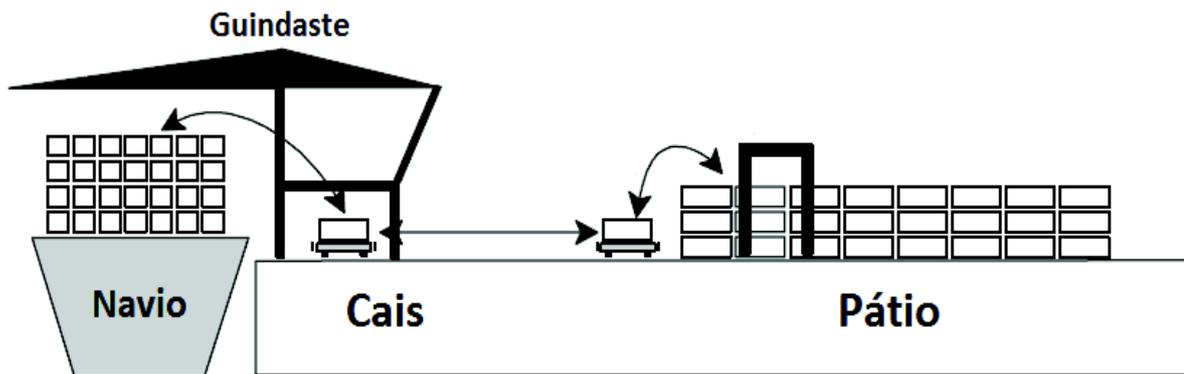
Segundo Ferreira e Neto (2011), o Instituto de Pesquisa Econômica Aplicada estudou a estrutura tarifária praticada pelos portos brasileiros e verificou que há uma grande divergência entre os mesmos na aplicação das tarifas. Além disso, o estudo realizado pelo IPEA constatou a prática de valores bastante elevados, prejudicando a competitividade dos produtos exportados. Segundo afirmam Ferreira e Neto (2011),

As tarifas pagas pela prestação desses serviços são estabelecidas pela administração de cada porto, que impõem não apenas os valores, mas sua própria estrutura tarifária, prejudicando a comparação entre os preços, pelo usuário, e compromete o potencial competitivo do setor.

Em virtude da importância da operação portuária para o sistema de transporte, e em decorrência do alto custo das operações em terminais de contêineres, é de suma importância que a operação de carregamento do navio ocorra de maneira eficiente e com baixo custo (ÁLVAREZ, 2006). Por isso, nos últimos anos, as operações dos terminais portuários foram alvo de diversos estudos na área de Pesquisa Operacional (PO), com enfoque nas operações realizadas tanto em terra quanto no cais (BIERWIRTH e MEISEL, 2010).

A estrutura de um terminal portuário pode ser dividida em partes, conforme a Figura 2. No pátio, os contêineres são armazenados para que possam ser posteriormente acessados por caminhões e levados ao cais, que é a região mais próxima do navio que irá receber os contêineres. Na região do cais, existem guindastes que transferem os contêineres do cais para o navio e do navio para o cais, realizando as operações chamadas de carga e descarga.

**Figura 2 - Terminal portuário**



Fonte: Adaptado de Choo (2005).

A eficiência de um terminal portuário de contêineres está centrada no planejamento adequado da movimentação dos contêineres, uma vez que a programação poderá refletir em redução de tempo e, conseqüentemente, em redução de custo. (WILSON e ROACH, 2000). Este planejamento envolve desde as operações realizadas no pátio do terminal até as operações realizadas no navio, sendo resultado da prévia organização e da distribuição coerente dos recursos utilizados para a operação (BIERWIRTH e MEISEL, 2010).

## 2.2 O Problema de planejamento de estiva (PPE)

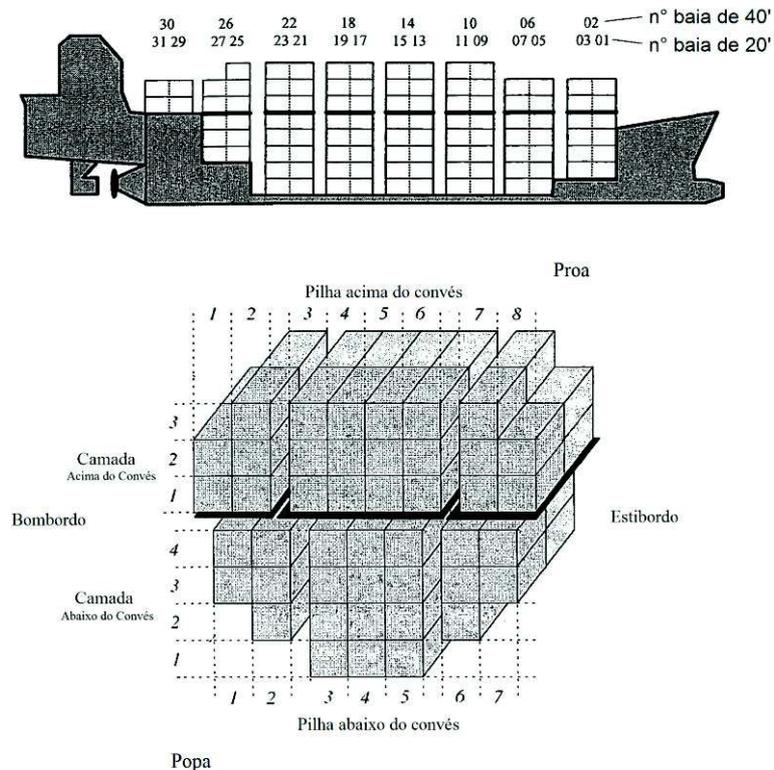
O problema do plano de estiva consiste na elaboração de um plano de carregamento que visa determinar a maneira mais adequada de organizar os contêineres a bordo do navio, com o objetivo de minimizar o tempo de carga e descarga (AZEVEDO *et al.*, 2009). Para tal, deve-se considerar as restrições de operação envolvidas no problema, podendo ser de diferentes tipos (SCIOMACHEN e TANFANI, 2007; WILSON e ROACH, 2000; AMBROSINO *et al.*, 2006; GIEMSCH e JELLINGHAUS, 2003):

- a) Minimização das movimentações de contêineres;
- b) Estrutura do navio em relação ao tipo, peso e tamanho dos contêineres;
- c) Estabilidade do navio e garantia de segurança na navegação;
- d) Minimização dos esforços estruturais;
- e) Localização de contêineres com relação ao tipo de carga transportada, como cargas refrigeradas ou perigosas;
- f) Utilização dos equipamentos do terminal;
- g) Uso efetivo das escotilhas;
- h) Uso de tanques de água de lastro para controle da estabilidade;
- i) Outros.

A primeira restrição (minimização de movimentação de contêineres) está diretamente ligada à quantidade de remanejamentos realizados ao longo da rota percorrida pelos navios porta-contêineres e depende, fundamentalmente, do planejamento da alocação destes contêineres, chamado de plano de estiva (SHIELDS, 1984; SPINRAD, 1994). Esta restrição estará em foco neste trabalho, uma vez que esta é uma das restrições que impactam no tempo e no custo de operação de um navio de carga e que existe um custo financeiro real relacionado a cada movimentação de contêiner no porto (DUBROVSKY *et al.*, 2002).

Para Sciomachen e Tanfani (2007), as dimensões exteriores de contêineres respeitam as normas ISO, sendo elas de 10', 20', 30' e 40' de comprimento. A dimensão padrão utilizada para identificação dos contêineres nos navios é o TEU, referente ao contêiner de 20' (1 TEU). Contêineres de 40' ocupam dois espaços contíguos. Segundo Choo (2005), um navio porta-contêiner é dividido, ao longo de seu comprimento, em segmentos conhecidos como baias, contadas de proa a popa. Cada baia é dividida em linhas, que são contadas do fundo para o convés do navio; e colunas, que informam a localização transversal de bombordo a boreste da embarcação. A Figura 3 mostra a estrutura celular de um navio de porta-contêineres.

Figura 3 - Estrutura celular de um navio porta-contêiner



Fonte: Adaptado de Guilbert e Paquin (2007).

Dubrovsky *et al.* (2002) detalham o problema relacionado ao remanejamento dos contêineres bloqueantes, que surge devido à estrutura celular de alocação dos contêineres. Admitindo que um *contêiner-j* (contêiner com destino ao porto  $j$ ) esteja armazenado em um navio  $n$  abaixo de um *contêiner-k* (contêiner com destino ao porto  $k$ ) e que  $k > j$ , quando  $n$  aportar em  $j$ , o *contêiner-k* deve ser temporariamente removido para descarregar o *contêiner-j*. Neste caso, o *contêiner-k* é chamado de “contêiner bloqueante” e o movimento de remoção temporária e retorno deste contêiner bloqueante para a pilha é chamado de “remanejamento”. O objetivo, na resolução do PPE, é encontrar um arranjo adequado que minimize o número de remanejamentos.

Avriel e Penn (1993) propuseram um modelo matemático binário, baseado na estrutura celular do navio de carga, com o objetivo de minimizar o remanejamento de contêineres. No entanto, aplicando o modelo a problemas reais, cujas dimensões do navio, em geral, são de grande escala, o problema assume um espaço de solução cuja dimensão torna proibitiva a utilização deste modelo, podendo ser empregado apenas para problemas cujas instâncias são pequenas (AVRIEL *et al.*, 1998). Conforme Azevedo *et al.* (2009), usando a formulação proposta por Avriel e Penn (1993) para uma instância cujo navio possua capacidade de 2000 TEU's, o número de arrumações possíveis neste caso será de aproximadamente  $3,3 \times 10^{5735}$ . Dessa forma, justifica-se o emprego de métodos estocásticos para encontrar uma solução viável (SCIOMACHEN e TANFANI, 2007; WILSON e ROACH, 1999).

A fim de reduzir o espaço de busca para o problema, Dubrovsky *et al.* (2002) sugerem uma simplificação do esquema de codificação completa de Todd e Sen (1997), chamada de codificação compacta. A proposta da codificação compacta foi conservar apenas as mudanças na estrutura celular que resultavam em carga e descarga ao longo da rota dos navios, simplificando, dessa forma, a avaliação das soluções, reduzindo o espaço de busca e preservando a factibilidade da solução. Mais tarde, Chiwiacowsky e Monteiro (2012) utilizaram a mesma codificação na aplicação de um algoritmo genético, comparando os resultados do algoritmo clássico com uma abordagem utilizando o operador de epidemia, demonstrando, assim, a aplicabilidade da técnica.

Azevedo *et al.* (2009) propuseram a codificação da solução através do emprego de regras de carregamento, que apresentam algumas vantagens em relação ao modelo de Avriel e Penn (1993), tais como a facilidade de incorporar o conhecimento prévio sob a forma de regras, gerar somente soluções factíveis e, principalmente, representar a solução de forma mais compacta se comparada com outras abordagens da literatura.

## 2.3 Critério de estabilidade

O plano de estiva exerce efeito significativo na estabilidade de um navio, pois a movimentação ou adição de cargas a bordo alteram o centro de gravidade (G) da embarcação, o que pode fazê-la imergir ou adernar (BOGOSSIAN, 2011). Por isso, é importante considerar a estabilidade no plano de estiva, a fim de distribuir as cargas a bordo de forma adequada. Além de garantir segurança e estabilidade, a distribuição uniforme das cargas tende a evitar esforços estruturais na embarcação (SCIOMACHEN e TANFANI, 2007; WILSON e ROACH, 2000; AMBROSINO *et al.*, 2006).

A seguir, são detalhados alguns conceitos necessários à compreensão das medidas de estabilidade de embarcações.

### 2.3.1 Centro de gravidade (G)

De acordo com Barrass e Derrett (2006), o centro de gravidade de um corpo é o ponto no qual pode-se assumir estar concentrada toda a massa de um corpo e no qual assume-se que a força da gravidade atue para baixo, com módulo igual ao peso deste corpo. O centro de gravidade depende da geometria da embarcação e da distribuição dos pesos contidos na mesma, sendo representado pela letra G.

### 2.3.2 Centro de carena (B)

Carena é a parte do casco da embarcação abaixo da linha de flutuação. O centro de carena (B) obedece ao Princípio de Arquimedes, sendo o ponto no qual considera-se que a

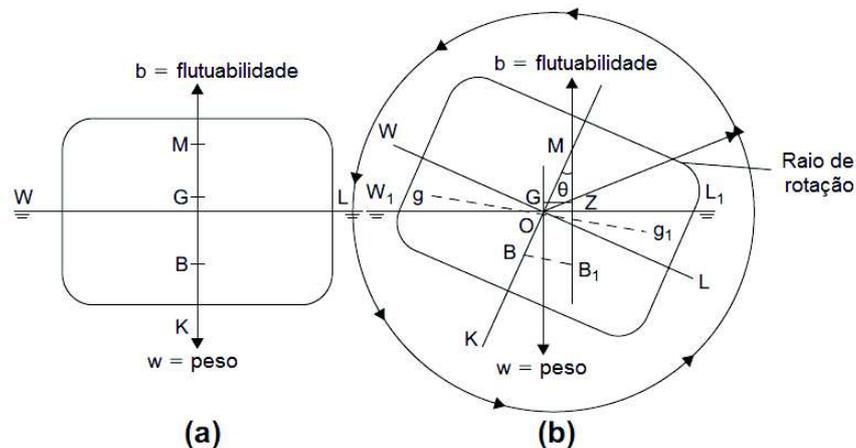
força de empuxo age verticalmente para cima com módulo igual ao peso relativo ao volume de água deslocada pelo navio. É o centro de gravidade do volume abaixo da linha de água (BARRASS e DERRETT, 2006), sendo o responsável pela flutuação de uma embarcação.

Para que o navio se encontre em condição de flutuação, o somatório de forças e de momentos deve ser igual a zero. Portanto, o peso da embarcação, que atua em  $G$ , deve ser igual à força de empuxo aplicada sobre o centro de carena ( $B$ ). A força de empuxo tem intensidade igual ao peso da massa de líquido deslocado pelo navio ( $W$ ) e depende da densidade ( $\rho$ ) do líquido, do volume de líquido ( $V$ ) deslocado pela embarcação e da gravidade ( $g$ ). Portanto, a massa de líquido deslocado é igual ao volume deslocado multiplicado pela densidade do mesmo, de acordo com a equação (1):

$$W = V \times \rho. \quad (1)$$

Na Figura 4(a), o navio se encontra em repouso sobre águas tranquilas, estando, portanto, o centro de gravidade ( $G$ ) na mesma linha vertical que o centro de empuxo ( $B$ ). Quando a embarcação sofre a ação de uma força externa (Figura 4(b)), uma porção da carena, antes submersa, é trazida para fora d'água. Assim, uma parte do empuxo, com centro de gravidade em  $g$ , é transportado para a posição com centro de gravidade  $g_1$ . Como o centro de carena é o centro de gravidade do volume de água deslocada, o centro de carena  $B$  é transferido para  $B_1$  (BARRASS e DERRETT, 2006).

**Figura 4 – Centro de Gravidade (G), Centro de Carena (B) e Metacentro (M)**



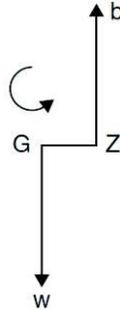
Fonte: Adaptado de Barrass e Derrett (2006).

### 2.3.3 Braço de endireitamento

Após o navio sofrer uma inclinação, é gerado um momento da força de empuxo em relação ao centro de gravidade, induzindo o navio a retornar à sua posição original. Este momento é chamado de momento de endireitamento (ou braço de endireitamento) e é dado pela distância entre o centro de gravidade  $G$  e o ponto  $Z$  ( $GZ$ ), mostrada na Figura 4(b). Este equilíbrio de momentos é explicado pela lei de Stevin (Lewis, 1988). Em resumo, o braço de

endireitamento, ilustrado pela Figura 5, gera uma força contrária ao movimento de rotação, induzindo o navio a retornar à sua posição de equilíbrio.

**Figura 5 – Braço de endireitamento (GZ)**



Fonte: Barrass e Derrett (2006).

O braço de endireitamento é fundamental para estimar a condição de estabilidade estática do navio.

### 2.3.4 Metacentro

Segundo Gillmer (1959), o conceito de metacentro foi introduzido por Pierre Bouguer, em 1746, que conceituou:

Para um corpo qualquer se admite uma linha perpendicular à linha d'água, onde há a aplicação da força de empuxo. Após a inclinação do corpo, cria-se outra linha onde é aplicada a nova força de empuxo, essa linha deve ser também perpendicular à linha d'água. O ponto de interseção dessas duas linhas é chamado de metacentro (M).

Este conceito é reafirmado por Barrass e Derrett (2006), sendo visto na Figura 4(b). De modo simples, metacentro é o ponto de curvatura dos centros de carena para pequenos ângulos de inclinação. O vetor formado entre o centro de gravidade (G) e o metacentro (M) é chamado de altura metacêntrica ( $GM$ ). A altura do metacentro inicial acima da quilha (KM) depende da forma do volume de água deslocado pelo casco (BARRASS e DERRETT, 2006). A altura metacêntrica ( $GM$ ) resultante do carregamento de um contêiner  $i$ , segundo Imai *et al.* (2006), é um dos principais fatores de estabilidade do navio, expresso na equação (2).

$$GM = G_0M + \frac{\sum_i w_i \cdot lh_i}{\Delta_T}, \quad (2)$$

onde:

$\Delta_T$  é a massa de líquido deslocado pelo navio após carregado;

$G_0M$  é a distância entre o centro de gravidade do navio ( $G_0$ ) e o metacentro ( $M$ );

$w_i$  é a massa do contêiner  $i$ ;

$lh_i$  é a distância vertical entre o centro de gravidade do navio ( $G_0$ ) e o centro de gravidade do contêiner  $i$ .

### 2.3.5 Equilíbrio da embarcação

Para melhor expor o conceito de equilíbrio para embarcações, é necessário identificar a posição do centro de gravidade, do centro de empuxo, do metacentro e, posteriormente, do braço de endireitamento. Considerando os conceitos de estabilidade apresentados por Imai *et al.* (2006) e Barrass e Derrett (2006), existem três tipos de equilíbrio:

- a) Equilíbrio instável (ou negativo): quando um corpo continua se movimentando na mesma direção da força aplicada, mesmo após a mesma parar de atuar. Ocorre quando  $GM < 0$ ;
- b) Equilíbrio neutro: quando um corpo entra em equilíbrio em qualquer posição, não voltando a sua posição de equilíbrio original. Ocorre quando  $GM = 0$ ;
- c) Equilíbrio estável (ou positivo): quando um corpo em equilíbrio for perturbado por uma força externa e retornar à posição original quando a força deixa de atuar sobre ele. Ocorre quando  $GM > 0$ .

O equilíbrio desejado para embarcações é o equilíbrio positivo. Para que uma embarcação possa ser considerada estável, o centro de gravidade deve estar abaixo do metacentro, para que a força de empuxo possa atuar para cima na vertical do centro de carena, induzindo o navio a retornar à sua posição de origem. Este momento é chamado de momento de endireitamento (Lewis, 1988). Barrass e Derrett (2006) ratificam que, para um navio estar em equilíbrio estável, o centro de gravidade deve estar abaixo do metacentro, isto é, o navio deve ter altura metacêntrica positiva.

Para que uma embarcação que se encontra em equilíbrio neutro ou instável possa ser levada ao equilíbrio estável, o centro de gravidade deve ser deslocado para baixo (BARRASS e DERRETT, 2006). Segundo Bogossian (2011), o centro de gravidade de uma embarcação se move quando há:

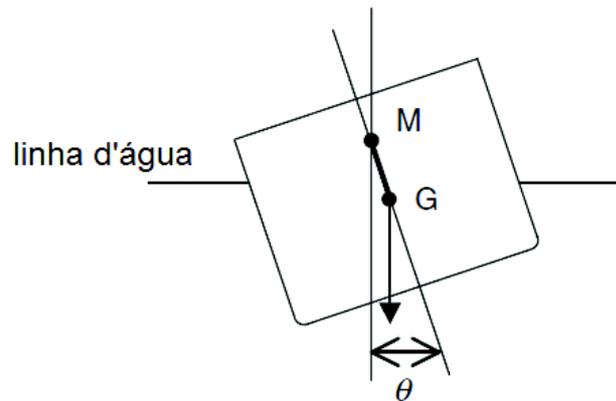
- a) Deslocamento de pesos na embarcação – neste caso, G se desloca paralelamente ao movimento do peso movido a bordo e com magnitude proporcional à distância pela qual o peso a bordo foi deslocado;
- b) Embarque de cargas – neste caso, G se movimenta em direção ao peso embarcado. Nesta situação, o navio tende a adernar na direção do embarque realizado;
- c) Desembarque de cargas – neste caso, G se movimenta em direção contrária ao peso desembarcado;
- d) Elevação de pesos – neste caso, G se desloca paralelamente à direção do movimento da carga.

Por fim, entende-se que a estabilidade é o resultado do dimensionamento adequado da altura metacêntrica (M) e do centro de gravidade (G), que corroboram para a atuação do braço de endireitamento (GZ) e contribuem para o equilíbrio estável da embarcação.

### 2.3.6 Ângulo de *list*

De acordo com os conceitos introduzidos por Barrass e Derrett (2006), um navio é dito estar em *list* quando o mesmo está inclinado por forças internas do navio, como por exemplo, pela má distribuição transversal das cargas que estão a bordo, sendo medido pela tangente de um ângulo  $\theta$ . A Figura 6 representa uma embarcação que está adernada e o ângulo de *list* formado.

Figura 6 - GM e list



Fonte: Adaptado de Imai *et al.* (2006).

A formulação do *list*, resultante do carregamento de um contêiner  $i$ , (IMAI *et al.*, 2006; BARRASS e DERRETT, 2006) é definida conforme equação (3).

$$\tan \theta = \frac{\sum_i w_i \cdot l w_i}{\Delta_T \cdot GM}, \quad (3)$$

onde:

$\tan \theta$  é a medida de inclinação transversal do navio;

$\Delta_T$  é a massa de líquido deslocado pelo navio após carregado;

$GM$  é a distância entre o centro de gravidade deslocado do navio em função da carga ( $G$ ) e o metacentro ( $M$ ), determinada pela equação (2);

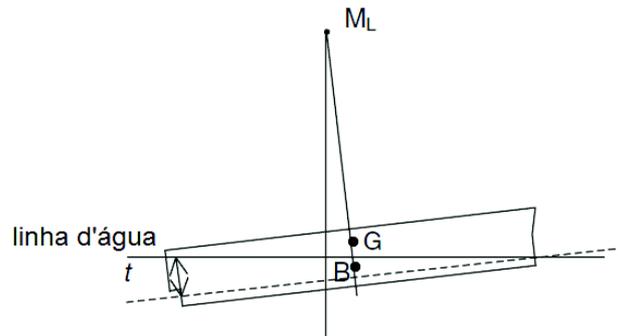
$w_i$  é a massa do contêiner  $i$ ;

$l w_i$  é a distância horizontal transversal entre o centro de gravidade deslocado do navio ( $G$ ) e o centro de gravidade do contêiner  $i$ .

### 2.3.7 Medida de *trim*

O *trim* (ou compasso) é também conhecido como estabilidade longitudinal, sendo a diferença de calado a vante e a ré. É uma medida de estabilidade semelhante ao *list*, porém se aplica à inclinação longitudinal de proa ou de popa da embarcação. A Figura 7 mostra a medida de *trim* de uma embarcação inclinada no sentido longitudinal.

Figura 7 - *Trim* de uma embarcação



Fonte: Imai *et al.* (2006).

Segundo Imai *et al.* (2006), a medida de *trim* é dada pela equação (4).

$$t = \frac{L \sum_i w_i \cdot l_i}{\Delta_T \cdot GM_L}, \quad (4)$$

onde:

$t$  é a medida *trim* do navio;

$\Delta_T$  é a massa de líquido deslocado pelo navio quando carregado;

$GM_L$  é a distância entre o centro de gravidade deslocado do navio em função da carga ( $G$ ) e o metacentro longitudinal ( $M_L$ );

$L$  é o comprimento longitudinal do navio;

$w_i$  é a massa do contêiner  $i$ ;

$l_i$  é a distância horizontal longitudinal entre o centro de gravidade deslocado do navio ( $G$ ) e centro de gravidade do contêiner  $i$ .

## 2.4 Otimização combinatória

A palavra “ótimo” é derivada do latim *optimus*, que significa “o melhor”. Otimizar, portanto, significa tentar levar o objetivo em foco ao seu melhor estado. Conforme citado por Goldberg (1989), a teoria da otimização estuda como descrever e alcançar o que é melhor, uma vez que se sabe como medir e modificar o que é bom ou ruim. A Otimização é uma área de conhecimento da Matemática que fornece métodos adequados para implementar melhorias dentro de determinados processos.

Problemas de otimização, em geral, têm como meta minimizar ou maximizar uma determinada medida de desempenho, denominada função objetivo (FO), sendo ela descrita em termos das variáveis de decisão e sujeita a condições limitantes às variáveis do problema, chamadas de restrições. O procedimento de identificar as variáveis de decisão, a FO e as restrições de um problema é chamado de modelagem (WRIGHT e NOCEDAL, 1999), e pode ser descrito no domínio contínuo ou discreto, sendo este último o alvo do estudo da otimização combinatória.

No estudo dos problemas de otimização combinatória, avalia-se o conjunto de soluções pertencente ao espaço de busca, cujos elementos representam soluções viáveis e que atendem às restrições impostas pelo modelo do problema, chamadas soluções factíveis. Dentre estas soluções, existe o ótimo global, que é a solução que apresenta o menor (ou maior) valor possível para a FO e cujos valores das variáveis não violem as restrições impostas pelo modelo.

Formalmente, um problema de minimização pode ser descrito como:

$$\text{Min } f(s) \quad (5)$$

$$\text{Sujeito a } s \in S. \quad (6)$$

A resposta para o problema é encontrar uma solução  $s^*$  tal que:

$$f(s^*) \leq f(s), \forall s \in S. \quad (7)$$

Após formular um modelo que represente o problema real, é possível aplicar as técnicas de otimização baseadas em algoritmos para chegar à solução ótima do problema (WRIGHT e NOCEDAL, 1999). Segundo Hertz e Widmer (2003), para problemas de otimização cujas instâncias são pequenas, a modelagem matemática e a aplicação de ferramentas de cálculo matemático costumam convergir para a solução ótima em um tempo considerado razoável. No entanto, a maioria dos problemas de interesse e estudados pela comunidade científica possuem um número significativo de variáveis envolvidas, o que impacta diretamente no tempo computacional dispendido ou no espaço de memória necessário para o processamento do algoritmo de solução. O estudo do esforço computacional envolvido em um algoritmo é chamado de Teoria de Análise de Algoritmos (SCHRIJVER, 2012).

#### 2.4.1 Teoria da Complexidade Computacional

Segundo Garey e Johnson (1979), em *Ciência da Computação*, a Teoria da Complexidade Computacional se concentra em classificar os problemas de acordo com o esforço computacional envolvido no algoritmo de solução dos mesmos, relacionando estas classes entre si. O esforço computacional pode ser quantificado de forma temporal ou espacial. A complexidade espacial tem relação com a memória de armazenamento, que relaciona o espaço de memória utilizada com os dados de entrada; a complexidade temporal, por sua vez, leva em conta o tempo de execução, que é resultado do número de instruções realizadas no cálculo da solução (GAREY e JOHNSON, 1979).

A análise da complexidade temporal avalia o tempo de execução de um algoritmo aplicado à instância de um dado problema, sendo este tempo dependente não apenas do tamanho dos dados de entrada, mas também da disposição dos mesmos, do *hardware* utilizado, da qualidade do compilador e, principalmente, da estrutura do algoritmo. A notação assintótica, denominada, “big-O” ( $O(f(n))$ ) determina a complexidade do algoritmo, representando uma medida assintótica do tempo necessário para a execução do pior caso da instância tratada. Este tempo de execução do algoritmo é a base para avaliar a complexidade computacional na classificação dos problemas de otimização.

A notação  $O(f(n))$  significa que o tempo de execução de um algoritmo é proporcional a  $f(n)$ , e pode ser de ordem polinomial ou exponencial. Algoritmos polinomiais são aqueles cuja complexidade é representada por uma função polinomial do tamanho da entrada, como, por exemplo, a pesquisa sequencial ( $O(n)$ ), ordenação por inserção ( $O(n^2)$ ), multiplicação de matrizes ( $O(n^3)$ ), entre outros. Algoritmos exponenciais são aqueles que possuem complexidade representada por uma função do tipo exponencial em relação à entrada, como por exemplo, o problema do caixeiro viajante, de complexidade ( $O(n!)$ ). Algoritmos de complexidade exponencial demandam tal quantidade de tempo de modo que instâncias de tamanho médio a grande porte não possam ser resolvidas em tempo hábil.

#### 2.4.2 Categorias de problemas

Existem três tipos de problemas quanto ao tipo de resposta que se espera (ROCHA, 2011): os problemas de decisão, que respondem quanto à existência de uma solução; os problemas de localização, que consistem em localizar e exibir uma solução que satisfaça uma dada restrição; e os problemas de otimização, cujo objetivo final é encontrar a melhor solução dentre todas as soluções possíveis e que satisfaça um critério de otimização previamente definido. Para elucidar os conceitos dos tipos de problemas, considere-se o problema do caixeiro viajante (PCV), que consiste na determinação do circuito de menor custo para um vendedor que deseja visitar um conjunto finito de cidades.

Para Goldbarg *et al.* (2015), problemas de decisão, também chamados de problemas de reconhecimento, são aqueles cuja resposta esperada é “sim” ou “não”. Assim, considerando o PCV, um problema de decisão pode consistir em responder se, para  $n$  cidades com distância  $c_{ij}$  entre as cidades  $i$  e  $j$  e um inteiro não negativo  $k$ , existe percurso de custo menor ou igual a  $k$ ? Se existir, a resposta será “sim”, caso contrário, “não”. Os problemas de localização têm como objetivo localizar e exibir o resultado, caso ele exista, ou seja, para  $n$  cidades com distância  $c_{ij}$  entre as cidades  $i$  e  $j$  e um inteiro não negativo  $k$ , um problema de localização deve localizar percurso de custo menor ou igual a  $k$  e exibir o resultado. Para problemas de otimização, o resultado consiste em encontrar, dentro do conjunto das possíveis soluções, a que apresenta o menor custo. Considerando o exemplo do PCV, o resultado esperado é encontrar dentre todos os circuitos possíveis, qual das deles apresenta o menor custo, exibindo este resultado.

Em resumo, a dificuldade de um problema de decisão não será maior que a dificuldade de um problema de localização, que por sua vez não terá dificuldade maior que a de um problema de otimização. Os problemas de otimização representam a categoria de problemas que possuem maior complexidade, sendo classificados de acordo com os algoritmos já conhecidos para a solução dos mesmos e agrupados em conjuntos cuja dificuldade de encontrar a solução seja semelhante (ROCHA, 2011).

A classificação dos problemas em P e NP é baseada na teoria desenvolvida por Cook (COOK, 1971), que afirma que problemas resolvidos em tempo polinomial por algoritmo determinístico constituem a classe P, enquanto problemas cujas soluções são geradas por

algoritmos não-determinísticos e podem ser verificada em tempo polinomial, constituem a classe NP. A classe de problemas NP-Completo foi conceituada a partir do problema da satisfabilidade (SAT), definido como o problema mais difícil da classe NP, sendo que todos os outros problemas da classe NP-Completo podem ser reduzidos a ele (COOK, 1971). A classe NP-Difícil compreende os problemas mais difíceis da classe NP-Completo. Um problema é NP-Difícil se existe um problema NP-Completo que pode ser reduzido ao problema NP-Difícil em tempo polinomial. A questão P *versus* NP é extensa e bastante estudada na Ciência da Computação (COOK, 1983).

### 2.4.3 Técnicas de solução

Para encontrar a solução de alguns problemas de otimização com instâncias de pequeno porte, é possível a aplicação de métodos exatos. Porém, quando há alguma descontinuidade na FO, algoritmos determinísticos não funcionam bem (YANG, 2010). Devido à complexidade dos problemas de otimização da classe NP e NP-Completo, a modelagem e a posterior aplicação de algoritmos determinísticos não possibilitam chegar à solução dos mesmos em tempo viável, devido à grande quantidade de elementos envolvidos e suas combinações, denominada de explosão combinatória (GOLDBERG, 1989; RAYWARD-SMITH *et al.*, 1996).

O problema da explosão combinatória, de acordo com Rayward-Smith *et al.* (1996), cria restrições relativas ao tempo de processamento para a solução de um dado problema. Tomando como exemplo o Problema do Caixeiro Viajante, já abordado anteriormente e, no qual, a quantidade de rotas possíveis é calculada por  $(n - 1)!$ , ao assumir  $n$  como o número de cidades que devem ser visitadas, é possível verificar, através dos dados apresentados na Tabela 2, o tempo necessário para se chegar à solução do problema utilizando um algoritmo exato, considerando o tempo de 1 ns para o processamento de uma rota, evidenciando, assim, o problema da explosão combinatória.

**Tabela 2 - Explosão combinatória gerada pelo PCV**

Cidades	Quantidade de circuitos (n-1)!	Tempo de processamento
7	720	720 ns
10	362.880	362.880 ns
20	$\sim 2,43 \times 10^{18}$	3857 anos
30	$\sim 8,84 \times 10^{30}$	$2,8 \times 10^{14}$ anos

Fonte: Elaborado pelo autor.

Especificamente com relação ao Problema do Plano de Estiva de navios porta-contêineres, o mesmo é classificado como um problema de otimização combinatória pertencente à classe NP-Completo (AVRIEL *et al.*, 2000). Existem trabalhos na literatura que aplicam métodos exatos para sua solução, como *Branch and Bound* (WILSON e ROACH, 1999) e programação inteira binária (AMBROSINO *et al.*, 2006), porém assumindo instâncias de pequeno porte. Contudo, alguns problemas teste são suficientemente grandes a ponto de inviabilizar o uso de métodos exatos devido à explosão combinatória gerada pelos possíveis arranjos de contêineres a bordo e, conseqüentemente, ao tempo de processamento

computacional imposto para a sua resolução, sendo uma alternativa o emprego de técnicas metaheurísticas.

## 2.5 Técnicas metaheurísticas

As heurísticas são procedimentos ou métodos aplicados a um dado problema a fim de se chegar a um resultado viável, geralmente baseados em regras de aplicação com um procedimento a ser seguido. A aplicação dos métodos heurísticos não assegura a obtenção da solução ótima do problema, e, por isso, em contraponto aos métodos exatos, são chamados de métodos aproximativos. Além disso, heurísticas simples não possuem uma reaplicação dos métodos baseada na memória da busca, o que implica em sempre obterem o mesmo resultado ou permanecerem presas a resultados considerados “ótimos locais”.

Glover (1986) define heurística como uma técnica utilizada para resolver dado problema em tempo computacional considerado aceitável, sem garantia de otimalidade do resultado. Ou seja, a aplicação de uma técnica heurística para solução de um problema implica na renúncia ao melhor resultado (ótimo global) devido ao custo do processamento computacional, buscando assim uma solução que se aproxime do ótimo global ou satisfaça um critério pré-estabelecido.

Em avanço ao desenvolvimento das heurísticas, surgiu o conceito de metaheurística. O termo metaheurística apareceu pela primeira vez em um artigo sobre Busca Tabu de Fred Glover (1986) e é derivado de duas palavras gregas: *meta* que significa “mais além” ou “alto nível”, e *heuristic*, que significa “encontrar” (GLOVER, 1986; YANG, 2010), o que sugere que métodos metaheurísticos são mais eficazes que os métodos heurísticos. Por definição, metaheurísticas são métodos de solução que promovem uma interação entre procedimentos de busca local e estratégias de alto nível, a fim de criar processos capazes de escapar de ótimos locais e realizar uma busca mais robusta em um determinado espaço de soluções (GLOVER e LAGUNA, 1993). Em outras palavras, a metaheurística consiste em um processo capaz de direcionar ou guiar uma dada heurística na exploração do espaço de soluções, sempre de forma não-determinística.

A utilização de técnicas de solução baseadas em metaheurísticas, diferentemente das técnicas por métodos exatos, não garante a obtenção do melhor resultado para o problema. No entanto, a metaheurística possibilita a obtenção de soluções que se aproximam do resultado ótimo em um tempo computacional viável, o que, geralmente, não pode ser alcançado para problemas de grande porte com o uso de modelagem e abordagens exatas, que impõem um alto custo computacional para a solução de problemas desta magnitude.

A metaheurística possui particularidades que a caracterizam, como as propriedades de diversificação e de intensificação. A diversificação tem a capacidade de realizar a exploração do espaço de busca de forma eficiente, geralmente através da randomização, ao passo que a intensificação leva a busca para regiões na vizinhança das melhores soluções atuais, permitindo identificar uma melhor solução candidata (GLOVER e LAGUNA, 1997). O ajuste fino entre estas duas propriedades é muito importante para a eficiência e desempenho

de um algoritmo. Pouca diversificação e muita intensificação podem fazer com que o algoritmo de busca fique aprisionado em um ótimo local. Do mesmo modo, pouca intensificação e muita diversificação pode fazer com que o algoritmo tenha dificuldade em convergir para resultados ótimos, finalizando antes de se aproximar do ótimo global (YANG, 2010).

Para Melián *et al.* (2003), a metaheurística é uma estratégia para projetar procedimentos heurísticos e se classificam pelo tipo de procedimento utilizado: metaheurística de métodos de relaxação, metaheurística de busca em vizinhança, metaheurística para processos construtivos, metaheurística para métodos evolutivos e metaheurística híbrida.

Neste trabalho, serão adotadas três abordagens metaheurísticas, sendo a primeira um Algoritmo Genético, classificado como metaheurística de método evolutivo populacional; a segunda uma metaheurística Busca Tabu de busca em vizinhança; e a terceira um algoritmo híbrido, combinando o Algoritmo Genético e a Busca Tabu. A seguir, são apresentadas as características proeminentes de cada algoritmo e suas particularidades.

### 2.5.1 Algoritmo Genético (AG)

O Algoritmo Genético (AG) é uma técnica de busca baseada na teoria da seleção natural e na genética natural (Goldberg, 1989). De acordo com Silva, A. J. M. (2005), os AG's são amplamente difundidos, pois possuem flexibilidade e simplicidade de aplicação, além de serem eficazes em buscas globais.

O AG é baseado na teoria da evolução natural proposta por Charles Darwin, e foi desenvolvido por John Holland e outros pesquisadores da Universidade de Michigan, na década de 60. Com base nas pesquisas realizadas sobre os AG's, Holland publicou um livro intitulado "*Adaptation in Natural and Artificial Systems*" (HOLLAND, 1975), o que deu a ele o título de "pai" do algoritmo genético (MITCHEL, 1997). Após este período inicial de desenvolvimento, os AG's vêm sendo analisados e aprimorados por diversos pesquisadores, sendo aplicados a diversos tipos de problemas.

No AG, o processo de exploração do espaço de busca ocorre através de um mecanismo de reprodução baseado em um processo evolutivo, aplicado sobre uma população de soluções e com o objetivo de encontrar uma solução adequada para um dado problema. Na biologia, cada indivíduo é composto por um conjunto de informações denominado genótipo, que contém a codificação das características do mesmo, representado no AG pelo cromossomo. Quando AG's são empregados para solucionar problemas do mundo real, cada indivíduo que compõe a população, denominado cromossomo, corresponde a uma possível solução para o problema (SANCHES, 2008).

Para explorar as possíveis soluções, é criada uma população de cromossomos (ou indivíduos), com tamanho determinado pela quantidade de cromossomos presentes na mesma (DIAS *et al.*, 2006). Uma função calcula a qualidade, com relação ao problema, de cada

cromossomo da população inicial e associa o valor calculado ao mesmo. Esta função é denominada função de aptidão e é o critério de avaliação utilizado para classificar os indivíduos da população, sendo o ponto de partida para a seleção. Ocorre então a etapa de seleção, onde é simulada a teoria da seleção natural dos seres vivos. Os indivíduos que apresentam o melhor material genético e, conseqüentemente, os melhores valores de aptidão, têm maior probabilidade de ser selecionados para se reproduzir, gerando novos indivíduos na etapa de reprodução.

Na etapa de reprodução, a recombinação genética faz com que os novos indivíduos, denominados filhos, herdem o código genético de seus pais. Os novos indivíduos são, assim, gerados e, a cada geração, há uma maior probabilidade de aproveitamento dos indivíduos mais aptos. Isto força a evolução de uma população, seguindo as regras da seleção natural e, assim, perpetuando as características que aproximam a solução do ótimo global. Estas etapas são realizadas iterativamente até que uma condição de parada seja satisfeita (CANTÚ-PAZ e GOLDBERG, 1999). A execução do AG é finalizada quando o mesmo atinge um estado que satisfaz determinada condição, chamada de “critério de parada”.

Nas próximas seções, serão apresentados os parâmetros genéticos e suas particularidades para os procedimentos que compõem o AG. É importante analisar a maneira como a definição destes parâmetros influencia o desempenho dos AG's, para que se possa estabelecê-los conforme as necessidades do problema e dos recursos disponíveis (CANTÚ-PAZ e GOLDBERG, 1999).

### ***2.5.1.1 Número de gerações***

No AG, uma geração é formada através da realização iterativa de uma série de etapas até a composição completa de uma população. Estas etapas são: cálculo de aptidão, ranqueamento, seleção, cruzamento e mutação. O número máximo de gerações indica a quantidade máxima de gerações que serão criadas antes de interromper a execução do algoritmo.

### ***2.5.1.2 Tamanho da população***

O tamanho da população é um parâmetro do AG que afeta diretamente o desempenho e a eficiência do algoritmo. Quanto maior o número de elementos na população, maior é a probabilidade de convergência, pois há maior possibilidade de que a solução desejada esteja entre os elementos que compõem esta população. Em contrapartida, o tempo de processamento demandado para a solução do problema aumenta. Já no caso de a população ser pequena, é possível que ocorra perda de diversidade da população (URSEM, 2002), o que pode acarretar em uma cobertura reduzida do espaço de busca, comprometendo a característica de exploração da metaheurística.

A correta definição do parâmetro de tamanho da população pode conferir à mesma uma boa diversidade, cobrindo de forma eficiente o espaço de busca sem consumir demasiadamente os recursos computacionais.

#### 2.5.1.3 Taxa de cruzamento

O operador de cruzamento é necessário para a intensificação de regiões promissoras do espaço de busca, e consiste na troca de genes entre um par de cromossomos selecionados, originando novos indivíduos. A taxa de cruzamento é definida no início do algoritmo e tem sua aplicação definida através de um valor gerado aleatoriamente. Caso ocorrer a opção de não cruzamento, poderão ser copiados os pais tais como são (CHAN *et al.*, 1998).

A taxa de cruzamento do AG representa a probabilidade de que os pais selecionados gerem filhos ou somente componham a nova população. Quanto maior for a taxa de cruzamento, mais rapidamente serão introduzidas na nova população as características que conferem maior aptidão aos indivíduos. Em contraponto, se a taxa de cruzamento for muito baixa, a evolução do algoritmo será muito lenta, fazendo com que o critério de parada seja atingido sem intensificar adequadamente regiões promissoras do espaço de busca ou aproximar-se do ótimo global.

#### 2.5.1.4 Taxa de mutação

A mutação consiste na alteração de um ou mais genes escolhidos de forma aleatória, tendo por objetivo diversificar a população e acrescentar características que não haviam sido exploradas até o momento (DIAS *et al.*, 2006). O papel da mutação é conferir diversidade à população e assegurar que qualquer ponto do espaço de busca possa ser explorado.

Apesar de o operador de mutação ser de fundamental importância para a eficiência do AG, a aplicação do mesmo não deve ser feita de forma deliberada. Uma taxa de mutação alta tornaria a exploração pelo espaço de busca essencialmente aleatória. Além disso, indivíduos promissores poderiam ser perdidos. Por esse motivo, a taxa de mutação deve ser pequena, se comparada à taxa de cruzamento.

#### 2.5.1.5 Taxa de substituição

O parâmetro de taxa de substituição define a porcentagem de indivíduos que serão substituídos na próxima geração. A definição de um valor baixo para o parâmetro pode tornar a convergência muito lenta pela baixa diversidade inserida a cada geração. Por outro lado, um valor alto para a taxa de substituição pode acarretar em perda de indivíduos promissores.

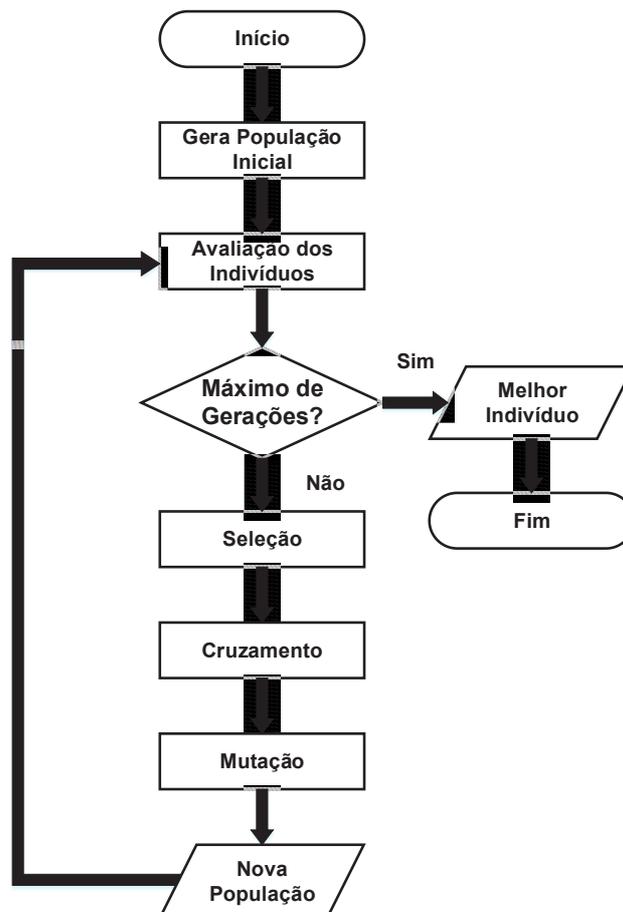
### 2.5.1.6 Critério de parada

O ideal para um problema de otimização seria que o AG interrompesse sua execução quando atingisse o ótimo global do problema. No entanto, para grande parte dos problemas de otimização, não se conhece este valor ótimo. Em virtude disso, normalmente, se adota um número máximo de gerações da população como critério de parada ou, então, um tempo limite de execução do algoritmo. O ajuste de um valor muito alto para o número máximo de gerações pode garantir a convergência da busca, porém exige um maior tempo de processamento computacional. Por outro lado, uma população muito pequena não leva a grandes custos computacionais, mas pode comprometer a convergência da busca (CANTÚ-PAZ e GOLDBERG, 1999).

Outro critério de parada que pode ser adotado é o encerramento do algoritmo quando a maior parte dos membros da população tornar-se semelhante, o que indica que o algoritmo perdeu diversidade e, possivelmente, não encontrará soluções melhores (HÜE, 1997).

A Figura 8 representa o fluxo de processamento de um AG.

Figura 8 - Fluxograma do AG



Fonte: Elaborado pelo autor.

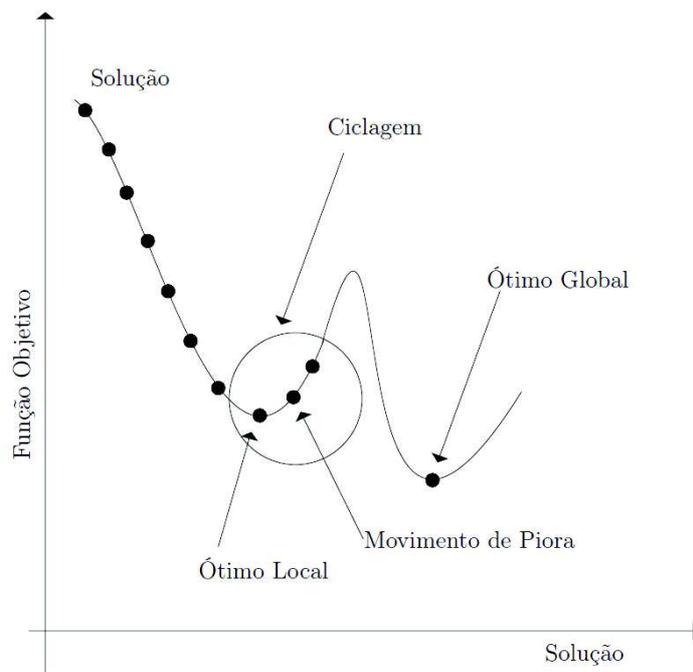
### 2.5.2 Busca Tabu (BT)

A forma padrão da Busca Tabu (BT) foi fundamentada nas ideias propostas por Glover, em 1986, tornando-se uma forma bem estabelecida de otimização e sendo utilizada, hoje, em diversos campos da ciência (FRAGA, 2006). Para Melián *et al.* (2003), a BT é uma metaheurística baseada em Busca Local que permite piora da solução corrente e utiliza estruturas de memória para escapar de ótimos locais. Por esses motivos, a BT é considerada uma heurística de busca global modificada.

A semelhança da BT com uma heurística de busca local reside no fato de partir de uma solução viável e tentar melhorá-la, buscando novas soluções candidatas no entorno da solução atual, chamado de “vizinhança”. Por outro lado, a BT difere da busca local por permitir movimentos de piora da solução, movendo-se, a cada iteração, para a melhor solução na vizinhança, a fim de explorar partes promissoras do espaço de busca, mesmo que este movimento resulte na piora do valor da FO.

A estratégia de assumir o melhor vizinho como a nova solução, utilizada para aprimorar a exploração da metaheurística, pode fazer com que o algoritmo forme ciclos, retornando a uma solução já visitada anteriormente e prendendo-se a um ótimo local. Este comportamento é denominado ciclagem. Neste caso, a BT se utiliza de estruturas de memória para que a busca não retorne às soluções visitadas anteriormente, o que lhe dá a capacidade de escapar de mínimos locais (GLOVER, 1986). Esta memória estruturada é chamada Lista Tabu (LT) e, de acordo com Glover e Laguna (1997), possui estrutura de curta e longa duração, as quais serão detalhadas nas próximas subseções. A ciclagem da BT é visualizada na Figura 9.

**Figura 9 - Ciclagem da BT**



Fonte: Fraga (2006).

A palavra tabu tem sua origem no idioma Tongan, sendo usada para indicar objetos que não podiam ser tocados por serem sagrados (GLOVER e LAGUNA, 1997). De acordo com o dicionário Aurélio da Língua Portuguesa, a palavra significa “proibição de determinada ação”. No caso da BT, a proibição imposta é a de retornar a soluções já visitadas por um número de iterações pré-determinado, evitando um ciclo no espaço de busca.

A Figura 10 apresenta o pseudocódigo de um algoritmo básico de BT. Considera-se  $s^*$  a melhor solução obtida até o momento e  $BTmax$  o número máximo de iterações sem melhora de  $s^*$ ,  $V$  um subconjunto da vizinhança da solução atual contido em  $N(s)$ ,  $m$  o movimento que leva a solução atual à melhor solução da vizinhança e  $f(s)$  a FO que se deseja minimizar.

**Figura 10 - Pseudocódigo da BT**

```

procedimento  $BT(f(\cdot), \mathcal{N}(\cdot), a(\cdot), |\mathcal{V}|, f_{min}, |T|, BTmax, s)$ 
1   $s^* \leftarrow s;$            {Melhor solução obtida até então}
2   $Iter \leftarrow 0;$        {Contador do número de iterações}
3   $MelhorIter \leftarrow 0;$  {Iteração mais recente que forneceu  $s^*$ }
4   $T \leftarrow \emptyset;$    {Lista Tabu}
5  Inicialize a função de aspiração  $a$ ;
6  enquanto  $(f(s) > f_{min} \text{ e } Iter - MelhorIter < BTmax)$  faça
7      $Iter \leftarrow Iter + 1;$ 
8     Seja  $s' \leftarrow s \oplus m$  o melhor elemento de  $\mathcal{V} \subset \mathcal{N}(s)$  tal que
        o movimento  $m$  não seja tabu ( $m \notin T$ ) ou
         $s'$  atenda a condição de aspiração ( $f(s') < a(f(s))$ );
9     Atualize a lista tabu  $T$ ;
10     $s \leftarrow s';$ 
11    se  $(f(s) < f(s^*))$  então
12        $s^* \leftarrow s;$ 
13        $MelhorIter \leftarrow Iter;$ 
14    fim-se;
15    Atualize a função de aspiração  $a$ ;
16 fim-enquanto;
17  $s \leftarrow s^*;$ 
18 Retorne  $s$ ;
fim  $BT$ ;

```

Fonte: Fraga (2006).

### 2.5.2.1 Memória de curta duração

A memória de curta duração, denominada Lista Tabu (LT), é a estrutura mais comumente usada em implementações da BT, sendo responsável por manter registradas somente as últimas soluções visitadas pelo algoritmo, excluindo as informações mais antigas da lista, mantendo, por consequência, um histórico recente das soluções visitadas. Estas soluções são rotuladas de *tabu* (GLOVER e LAGUNA, 1997).

Com o recurso da LT, o algoritmo tem a capacidade de saber se uma solução candidata, pertencente à vizinhança, já foi visitada anteriormente. Em caso afirmativo, essas soluções são proibidas de ser revisitadas enquanto elas estiverem na memória, evitando a

ciclagem do algoritmo. A construção da LT é realizada a cada iteração do algoritmo, isto é, a cada vez que a BT assume uma nova solução como solução atual, pois a lista armazena, a cada iteração, a solução anteriormente visitada.

Outra característica da BT, com relação à utilização da LT, é o custo computacional necessário para o armazenamento da solução completa, que é acentuado quando se utiliza estruturas de dados complexas. Conforme Glover e Laguna (1997), existe uma certa dificuldade em armazenar as soluções inteiras devido ao custo computacional envolvido ao testar se a solução pertence à LT. Por isso, a memória de curta duração armazena somente os movimentos executados para chegar às soluções visitadas até o momento.

### *2.5.2.2 Memória de longa duração*

A memória de frequência, considerada como uma memória de longa duração, complementa a informação fornecida pela memória de curta duração, armazenando a frequência com que determinadas características aparecem nas soluções visitadas, auxiliando o algoritmo a evitar ou visitar caminhos que apresentem estas características (GLOVER e LAGUNA, 1997).

A forma como estes dados são armazenados, segundo Glover e Laguna (1997), depende de como a informação será utilizada pelo algoritmo, podendo registrar com que frequência os atributos de uma solução mudam ou com que frequência os atributos fazem parte das soluções. As formas mais comuns de quantificar estas frequências são através de soma absoluta ou taxas.

As informações armazenadas na memória de longa duração permitem ao algoritmo fazer análises e inferências a respeito das soluções obtidas através da busca, possibilitando identificar movimentos que levem a regiões promissoras do espaço de busca e aumentar a probabilidade de encontrar o ótimo global. Por outro lado, a análise destas informações pode também fornecer ao algoritmo um conhecimento sobre as regiões menos promissoras do espaço de busca, viabilizando uma melhor orientação da trajetória na procura pela solução.

### *2.5.2.3 Função de aspiração*

A utilização da LT tem sua importância justificada pela perspectiva de escapar de ótimos locais. Contudo, ao armazenar apenas os movimentos que conduziram às últimas soluções visitadas, a busca pode tornar-se restritiva, ou seja, a lista pode impedir não só que se retorne a posições do espaço de busca que já foram exploradas, mas também pode ocasionar a proibição de soluções promissoras. Por este motivo, a BT dispõe de um recurso denominado “função de aspiração”, o qual permite que uma solução seja visitada, mesmo que o movimento que conduz até ela se encontre na LT.

A aceitação do movimento, apesar de ser tabu, se dá sob determinadas circunstâncias, dependendo da função de aspiração. Por exemplo, considerando a função de aspiração  $A(f(s)) = f(s^*)$ , tal que  $s^*$  é a melhor solução encontrada até o momento, aceita-se um movimento tabu somente se ele conduzir a uma solução da vizinhança que apresente valor melhor que  $s^*$ . Em outras palavras, se a solução candidata, pertencente à vizinhança, for gerada por um movimento tabu, pode-se permitir este movimento se o mesmo gerar uma solução com valor mais promissor que a melhor solução encontrada até o momento.

#### 2.5.2.4 Critério de parada

Em geral, o critério de parada da BT é definido ajustando-se um número máximo de iterações do algoritmo e ajustando o valor limite de iterações sem encontrar uma solução que aprimore o valor da FO da melhor solução conhecida. Outra opção é definir que a execução finalize quando a solução atinja um valor ótimo global (se o mesmo for conhecido). No caso de o ótimo global não ser conhecido, a interrupção da execução pode ser aplicada quando a solução encontrada possuir um valor de qualidade considerado satisfatório sob determinada condição. Esta última condição evita que o algoritmo continue sendo executado de forma desnecessária.

#### 2.5.3 Algoritmos híbridos

Segundo Raidl (2006), as diversas técnicas de otimização aplicadas aos problemas abordados na literatura mostram que não existe uma estratégia de otimização adequada para cada tipo de problema. Cada tipo de metaheurística possui particularidades que acentuam as características de exploração e intensificação. Por exemplo, algoritmos baseados em busca populacional têm uma forte inclinação à característica de diversificação, enquanto métodos baseados em busca em trajetória possuem características mais voltadas à intensificação (RAIDL, 2006).

Para problemas de otimização, a aplicação de metaheurísticas de forma simples tende a explorar apenas a característica mais marcante do método que foi escolhido, o que faz com que a orientação da busca pelo espaço de soluções se guie somente pela vantagem oferecida por este método. Neste contexto, surge a opção da hibridização, que visa explorar e unir as características de cada metaheurística, obtendo, assim, um melhor desempenho. Nos últimos anos, o crescimento no número de trabalhos propondo a aplicação de algoritmos que se utilizam da combinação de várias ideias algorítmicas, e não apenas dos conceitos de uma metaheurística simples, mostra a relevância da hibridização (RAIDL, 2006).

Diversos autores já propuseram trabalhos utilizando abordagens metaheurísticas híbridas. Gómez e Weissheimer Júnior (2012) propuseram um algoritmo híbrido, composto por AG e BT, com o objetivo de analisar parâmetros para a transmissão *Internet Protocol Television* (IPTV). Uma hibridização envolvendo Colônia Artificial de Abelhas e uma

heurística de busca local foi proposta por Sundar *et al.* (2010) para resolver o problema de recobrimento. Para resolver um problema de otimização combinatória multiobjetivo, Ölçer (2008) propôs um algoritmo híbrido baseado em Algoritmo Evolutivo, um AG para geração de uma Fronteira de Soluções Pareto-Ótima combinado com TOPSIS, um algoritmo de análise multicritério determinístico para ranqueamento das soluções, com a finalidade de ser um sistema de apoio à decisão.

Quando se deseja resolver um problema de otimização através de uma abordagem híbrida, existem algumas questões que devem ser consideradas e definidas na escolha das técnicas utilizadas na hibridização e na forma como será efetuada esta combinação. De acordo com Raidl (2006) e Talbi (2009), a hibridização pode diferir em relação às técnicas de otimização envolvidas, nível de hibridização, ordem de execução e estratégia de controle. Estes conceitos são explicados a seguir.

### 2.5.3.1 Estratégias de hibridização

A hibridização de algoritmos pode ser implementada através da combinação de diferentes tipos de técnicas de otimização, sendo elas determinísticas ou estocásticas. Raidl (2006) afirma que a hibridização de algoritmos pode ser realizada através das seguintes combinações:

- a) Diferentes estratégias metaheurísticas;
- b) Metaheurísticas com algoritmos específicos ou modelos de simulação;
- c) Técnicas de otimização de outras áreas, como, por exemplo, inteligência artificial.

A combinação híbrida de metaheurísticas apresenta como vantagem a união das características específicas de cada técnica. Por exemplo, combinar uma metaheurística de busca local, que possui forte traço de intensificação, com um algoritmo evolutivo de busca populacional, que possui característica predominante de diversificação, pode resultar em um algoritmo com propriedades marcantes de exploração e intensificação simultaneamente.

### 2.5.3.2 Nível de combinação

Em relação ao nível de combinação, as técnicas híbridas podem ser classificadas como *high-level* e *low-level* (RAIDL, 2006). As hibridizações *high-level* não alteram o funcionamento dos componentes internos de cada técnica utilizada, operando em modo combinado de forma a interagir sob determinadas definições de interface. As combinações híbridas classificadas como *low-level* possuem um acoplamento maior, onde as técnicas combinadas dependem uma da outra, trocando funções ou componentes individuais durante a execução do algoritmo.

### 2.5.3.3 Ordem de execução

De acordo com Raidl (2006), a ordem de execução das técnicas envolvidas no algoritmo híbrido podem ser:

- a) Sequencial, quando um algoritmo é executado posteriormente a outro e o segundo sempre recebe as informações do primeiro. Neste tipo de hibridização, a informação é repassada sempre em um sentido;
- b) Intercalada, onde as duas técnicas são executadas a cada iteração, havendo fluxo de informações entre a execução das mesmas. Raidl (2006) não especifica o fluxo de informações entre as técnicas que compõem o algoritmo híbrido intercalado.
- c) Paralela, na qual os algoritmos são executados simultaneamente, trocando dados entre si a cada execução, possibilitando repassar a informação em qualquer sentido.

### 2.5.3.4 Estratégia de controle

A estratégia de controle especifica a forma como as técnicas empregadas na abordagem híbrida se relacionam. Conforme Raidl (2006), as técnicas de otimização podem ser combinadas através de métodos *integrados* ou *colaborativos*. No método integrado, um dos algoritmos é considerado como parte integrante do outro, sendo uma abordagem *low-level* e intercalada, pois possuem uma ligação a nível de componentes internos e são executados, a cada iteração, de forma integrada. No esquema colaborativo, os métodos trabalham separadamente e somente trocam informação entre si, mantendo assim o seu funcionamento original preservado.

## 2.6 Processamento de alto desempenho

A organização básica do computador foi introduzida por Von Neumann nos anos 50, sendo batizada com seu nome, denominada de Arquitetura de Von Neumann. Esta arquitetura ainda serve de base para os computadores atuais (DALE e LEWIS, 2012) e é formada, basicamente, por cinco partes fundamentais: unidade de controle, unidade aritmética, memória e mecanismos de entrada/saída, sendo sua principal característica a possibilidade de armazenar as instruções no mesmo espaço de memória dos dados.

Os computadores executam uma instrução de cada vez, sendo que cada instrução executada pelo computador é composta de 3 partes: busca, decodificação e execução. Estas etapas são executadas de forma sequencial (STALLINGS, 2009). Quanto maior a velocidade de execução, maior é a capacidade de processamento.

Devido à demanda crescente pelo aumento de desempenho dos computadores (Lei de Moore), os grandes fabricantes passaram a concentrar seus esforços no incremento da velocidade dos processadores. No entanto, o aumento de desempenho que se concentra no aumento de velocidade do processador esbarra em algumas limitações físicas, tais como o tamanho do transístor, a dissipação de calor, o consumo energético, entre outros (KASIM *et al.*, 2008).

### 2.6.1 Paradigmas de programação paralela

No contexto da necessidade de aumento da velocidade de processamento e frente às limitações impostas a este aumento, torna-se necessário encontrar alternativas para viabilizar essas exigências. Uma solução viável é organizar recursos de forma que múltiplas unidades de processamento possam cooperar entre si para realizar tarefas paralelamente. Segundo Grama *et al.* (2003), a ideia central do paralelismo é dividir a demanda de processamento entre os recursos computacionais para que um sistema possa resolver um dado problema de maneira mais ágil que um sistema sequencial.

O paralelismo pode ser implementado pela concepção de sistemas *multicore* ou multicomputadores. No sistema *multicore*, múltiplos núcleos são integrados em um único chip para trabalhar de forma simultânea com memória compartilhada, caracterizando-se por processos escrevendo dados em posições específicas da memória, as quais podem ser acessadas pelos demais processos. Já nos sistemas multicomputadores, a interação entre as partes componentes é realizada sobre uma rede de interconexão por meio de trocas de mensagens, formando os chamados aglomerados computacionais ou *clusters* (MATTSON *et al.*, 2004).

A situação ideal para uma arquitetura paralela é que  $n$  processadores (ou computadores) possam aumentar  $n$  vezes a capacidade de processamento de um sistema, porém, na prática, não é o que se observa. Geralmente, o problema abordado não pode ser dividido igualmente entre os recursos, uma vez que existe interação entre eles e, por consequência, influencia no tempo total de resolução. A métrica de desempenho que elucida a capacidade de processamento com o aumento de recursos computacionais é chamada de *speed-up* (ALBA, 2005).

Apesar de a paralelização oferecer benefícios em relação ao alto desempenho, existe uma dificuldade intrínseca à implementação do código, pois um algoritmo naturalmente sequencial exige adequações para que possa ser executado em uma arquitetura paralela. As principais barreiras encontradas neste tipo de implementação são: necessidade de sincronismo entre as partes, o particionamento de dados ou de tarefas, dependência de operações, comunicação e balanceamento de carga (TANENBAUM, 2003). Apesar de exigir uma estratégia de programação mais elaborada, o paralelismo apresenta uma redução significativa no tempo de processamento, o que justifica os esforços de desenvolvimento de sistemas de alto desempenho.

Os melhores resultados encontrados para os problemas práticos de otimização abordados na literatura foram obtidos por algoritmos híbridos e paralelos. Combinações de algoritmos tais como Busca Local, *Simulated Annealing*, BT e Algoritmos Evolucionários mostraram-se algoritmos de busca bastante poderosos quando paralelizados devido ao fato de a paralelização permitir o acesso a recursos computacionais de alta performance, tais como os sistemas multiprocessadores ou as redes distribuídas (TALBI, 2009). Diferentes estratégias de paralelização podem ser adotadas para a implementação de uma ferramenta metaheurística paralela. Algumas das opções são discutidas a seguir (GRAMA *et al.*, 2003).

### 2.6.1.1 Mestre-Escravo

Essa abordagem pode ser utilizada quando se possui uma tarefa com alto custo de processamento e que possa ser particionada em outras tarefas de proporções menores, sem dependência entre elas. Dessa forma, cada tarefa menor pode ser tratada de forma independente por um processo escravo distinto, que envia o resultado de volta para o processo mestre (GRAMA *et al.*, 2003). Na abordagem *bag of tasks*, o mestre possui um conjunto grande de tarefas menores, que são enviadas para cada escravo, uma de cada vez. Com uma divisão adequada das tarefas, é possível obter um bom balanceamento de carga entre os escravos.

### 2.6.1.2 Pipeline

Esta estratégia funciona de forma semelhante a uma linha de montagem. Caso o algoritmo possua um conjunto de passos a ser realizado sequencialmente, o *pipeline* pode ser utilizado, destinando um conjunto de instruções para cada processo, sendo bastante útil para aumentar a vazão total do sistema. Cada saída de um processo é tomada como entrada para o próximo, e assim sucessivamente. Após um determinado número de passos, o pipeline é preenchido e a vazão aumenta significativamente, pois cada estágio (ou estação de trabalho) da linha de montagem está processando uma tarefa paralelamente.

### 2.6.1.3 Divisão e Conquista

Na estratégia de “divisão e conquista”, o processamento paralelo é realizado dividindo a tarefa maior em diversas tarefas menores. O nó raiz verifica se o trabalho a ser desempenhado pode ser realizado localmente. Em caso negativo, ele divide a tarefa em partes menores e delega estas partes para dois ou mais processos escravos. Cada um dos processos escravos tem a capacidade de também definir se trata a tarefa ou realiza mais uma divisão do trabalho, delegando estas partes a outros processos. Esta estratégia assume então uma topologia em árvore para o tratamento do problema. Ao final do processamento dos nós folha

(processos mais baixos da árvore), os resultados são repassados para o processo pai, que trata estes resultados e os repassa para o processo acima. Esse procedimento continua sucessivamente, até que o nó raiz processe o resultado final.

#### 2.6.1.4 Fases Paralelas

Na estratégia de fases paralelas, todos os processos envolvidos executam paralelamente e sincronizam o processamento regularmente durante a execução, sendo necessário garantir que todos os processos participantes terminem seu processamento antes de iniciar a próxima fase.

#### 2.6.2 Métricas de desempenho

Existem diferentes métricas para avaliar o desempenho de algoritmos paralelos. Dentre estas métricas, as mais utilizadas são o *speed-up* e a eficiência, sendo o *speed-up* a mais importante (ALBA, 2005). O *speed-up* compara o tempo de execução do algoritmo sequencial com o tempo de execução do mesmo algoritmo executado de forma distribuída para a solução de um dado problema, calculando a razão entre eles. O cálculo de *speed-up* é definido por Alba (2005) e explicado na equação (8).

$$S(p) = \frac{T(1)}{T(p)}, \quad (8)$$

onde:

$S(p)$  é a medida de *speed-up* do algoritmo paralelo para  $p$  processadores;

$T(1)$  é o tempo de execução com um processador;

$T(p)$  é o tempo de execução com  $p$  processadores;

$p$  é o número de processadores.

A eficiência, por sua vez, trata da relação entre o *speed-up* e o número de processadores, e é também definida por Alba (2005) e elucidado na equação (9).

$$E(p) = \frac{S(p)}{p}, \quad (9)$$

onde:

$E(p)$  é a medida de eficiência do algoritmo paralelo para  $p$  processadores;

$S(p)$  é a medida de *speed-up* do algoritmo paralelo para  $p$  processadores;

$p$  é o número de processadores.

### 2.6.3 Algoritmo Genético Paralelo (AGP)

Conforme abordado anteriormente, há uma demanda por algoritmos de alto desempenho que ofereçam soluções em tempo aceitável para problemas reais, o que abriu espaço para o desenvolvimento dos algoritmos híbridos e paralelos. Os Algoritmos Genéticos são algoritmos que possuem características que os tornam implicitamente paralelos, pois possuem uma estrutura computacional que facilita a paralelização (SILVA, A. J. M., 2005).

Alguns métodos de paralelização do AG usam uma única população, aplicando operadores genéticos em paralelo. Este modelo, chamado Paralelização Global, é adequado a arquiteturas de memória compartilhada (CANTÚ-PAZ e GOLDBERG, 1999). Outros modelos de paralelização dividem a população em subpopulações que evoluem em paralelo, periodicamente “migrando” indivíduos de uma subpopulação para outra (ALBA, 2005).

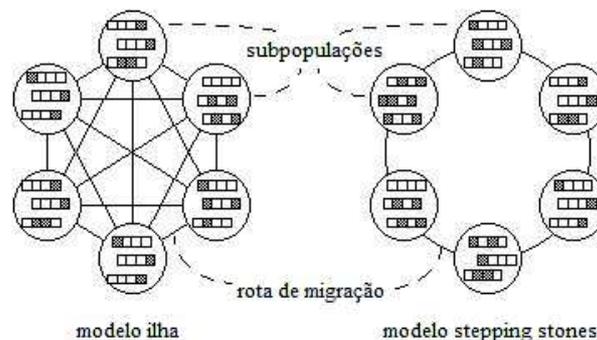
Muitos pesquisadores em Algoritmos Genéticos, tais como Cantú-Paz (2000), Schwelm (1996) e Levine (1994), acreditam que o AGP representado por múltiplas subpopulações é um modelo mais fiel à realidade da evolução das espécies.

#### 2.6.3.1 Modelos de migração *stepping-stone* e *island* (ilha)

As estratégias de migração podem ser limitadas ou não ao fator de vizinhança da subpopulação. Os modelos de migração apresentados a seguir são detalhados em Silva, A. J. M. (2005).

O modelo *stepping-stone* é caracterizado por um anel lógico de processos, sendo que a comunicação ocorre quando cada processador envia seu melhor indivíduo para os vizinhos. Após um número finito de etapas, todos os processadores têm a melhor solução global. No modelo *island* (ilha), indivíduos de melhor aptidão podem migrar para todos os outros processos. A migração pode ser implementada enviando a melhor solução para que o processo mestre distribua para os demais ou realizando múltiplas difusões, enviando sua própria solução para os outros. A Figura 11 ilustra as estratégias de migração entre as subpopulações.

Figura 11 - Estratégias de migração para Modelos *Island* e *Stepping-Stone*



Fonte: Silva, A. J. M. (2005).

#### 2.6.4 Busca Tabu Paralela (BTP)

A BT, particularmente, é uma metaheurística que se adapta bem a abordagens paralelas. De acordo com Crainic *et al.* (1997), existe um grande número de estratégias de paralelização, para a BT, resumidas a seguir:

- a) Avaliação de Vizinhança Paralela – este método é uma implementação da estratégia mestre-escravo de paralelização e é indicada para problemas cuja avaliação da vizinhança possui um custo computacional elevado. Nesta abordagem, a vizinhança é dividida em parcelas iguais entre os processos escravos, que, por sua vez, executam as avaliações das soluções baseadas em suas restrições, enviando o melhor vizinho encontrado de volta para o processo mestre;
- b) Busca Tabu Paralela Independente – neste modelo, um determinado número de execuções da BT é implementado simultaneamente e cada busca possui parâmetros diferentes, com o objetivo de explorar o espaço de soluções de forma mais abrangente. Esta abordagem é indicada para arquiteturas com memória distribuída, exigindo pouca comunicação entre os processos;
- c) Busca Tabu Paralela Interativa: Este modelo é similar ao modelo Busca Tabu Paralela Independente, ocorrendo interações entre os processos em intervalos predeterminados. Esta interação tem o objetivo de determinar qual das buscas é a mais promissora e transferir esta informação aos outros processos. Assim como a BT Paralela Independente, esta abordagem exige grande disponibilidade de comunicação;
- d) Divisão do Espaço de Busca: neste modelo, cada processo recebe uma porção diferente do espaço de busca, retornando ao processo mestre o resultado de sua busca após finalizada, sendo estas soluções combinadas em uma solução final. É indicado para problemas os quais podem ter seu espaço de busca dividido. Uma vantagem desta abordagem é que a mesma não gera *overhead* de comunicação.

Um exemplo de aplicação da BTP é apresentado no trabalho desenvolvido por He *et al.* (2006), no qual duas instâncias conhecidas do PCV foram aplicadas a uma abordagem da BTP. O modelo foi implementado seguindo a estratégia mestre-escravo, com comunicação assíncrona entre o processo mestre e os demais. A solução inicial do algoritmo foi desenvolvida através de um algoritmo guloso e, a partir deste ponto, o mestre divide o espaço de soluções entre os escravos, para que cada um intensifique a busca em uma área diferenciada de cidades. Quando um processo escravo termina a tarefa, ele envia o resultado para o processo mestre e recebe outra tarefa da fila.

### 3 REVISÃO BIBLIOGRÁFICA

Neste capítulo, são apresentados os trabalhos que tratam dos diferentes problemas relacionados às operações portuárias e suas abordagens, com uma evidência maior ao problema de planejamento de estiva.

Os trabalhos apresentados nesta seção propõem o emprego desde algoritmos exatos, com formulações binárias, até métodos estocásticos, considerando diferentes objetivos ligados à operação portuária, como por exemplo, a estabilidade da embarcação.

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

Um dos problemas enfrentados nas operações dos portos é encontrar o melhor escalonamento de tempo para navios que desejam aportar para realizar operações de carga e descarga. As posições de atracação, às quais os navios devem ser alocados, são denominadas “berços” que, segundo a Confederação Nacional de Transportes (CNT), é o “espaço destinado ao navio em um cais no qual ele pode operar com segurança”. Em outras palavras, o problema de alocação de berços (PAB) consiste em atribuir os navios que chegam a um determinado porto às posições de atracação disponíveis ao longo de um cais. As principais decisões a serem tomadas neste processo envolvem a escolha de onde e quando os navios deverão atracar (CORDEAU *et al.*, 2005).

A maior parte dos berços nos grandes portos é alugada pelos operadores dos navios para a movimentação da carga, a fim de alcançar maior produtividade (IMAI *et al.*, 2001). Este fato justifica a importância de limitar o número de berços operantes e minimizar o tempo de realização destas tarefas, pois o tempo de operação para um navio específico não é necessariamente o mesmo para cada berço. (BIERWIRTH e MEISEL, 2010).

Muitos trabalhos na literatura abordam o PAB. Um dos primeiros trabalhos relevantes a respeito do assunto foi apresentado por Lai e Shih (1992). Neste trabalho, assumiu-se o cais como uma linha contínua particionada em diversas seções, em cada uma das quais somente um navio poderia atracar, sendo desenvolvido um algoritmo heurístico baseado em regras *first-come-first-served*. Mais tarde, Brown *et al.* (1994) trataram o PAB em portos navais, procurando pela melhor configuração de atribuição navio-berço que maximizasse o somatório de benefícios por navio no porto, sujeito às limitações de comprimento e profundidade do berço, além da força dos cabos de amarração utilizados.

Imai *et al.* (1997) foram os primeiros a introduzir a ideia de que, para obter melhor desempenho no terminal, a solução ótima não deve ser baseada em regras *first-come-first-served*. A formulação proposta por eles era baseada em maximização do rendimento do berço e minimização do atraso. Mais tarde, Imai *et al.* (2003) estenderam a formulação de berços discretos, aplicando o modelo a berços contínuos. Guan e Cheung (2004) apresentaram um modelo de formulação seguindo a ideia de Imai *et al.* (2003), aplicando um coeficiente para cada navio, objetivando minimizar o tempo de permanência do navio no porto.

Imai *et al.* (2008) introduziram uma variação do PAB, focada no tempo de espera do navio, com o objetivo de minimizar o tempo total de serviço do mesmo. Uma metaheurística AG foi desenvolvida e resultados numéricos mostraram que o algoritmo cumpriu com o objetivo proposto. Wang e Lim (2006) resolveram o problema de alocação de berços discretos minimizando posições livres e atrasos usando a heurística *Beam Search*, que superou o estado da arte e a abordagem tradicional do algoritmo. Oliveira *et al.* (2012) aplicaram o método *Clustering Search* (CS) utilizando *Simulated Annealing* (SA) para geração das soluções para o PAB com tempos de chegada dinâmicos e berços estáticos.

Além do PAB, existem outros problemas relacionados às operações em terminais portuários que são de grande importância para a eficiência das operações realizadas neste contexto. Entre estes problemas, destaca-se a alocação de um conjunto de guindastes para cada berço, frequentemente abordado em pesquisas envolvendo minimização de custos e otimização de movimentação dos equipamentos. Este problema é denominado problema de alocação de guindastes (PAG). Relacionado ao PAG, destaca-se também o problema de agendamento das tarefas de transbordo, cujo objetivo é particionar as tarefas de carga e descarga entre os guindastes já alocados, definindo a ordem das tarefas a serem executadas. Este problema é referenciado na literatura como problema de escalonamento de guindastes (PEG) (CHOO, 2005).

Contribuições nestas outras modalidades de problemas incluem os trabalhos de Park *et al.* (2003), Imai *et al.* (2005), Liang *et al.* (2009) e Bierwirth e Meisel (2010), todos propondo o uso de abordagens metaheurísticas de solução.

### **3.2 O problema de planejamento de estiva de navios porta-contêineres**

Imai *et al.* (2007) afirmam que, enquanto o tempo total de operação do navio inclui os tempos de atracação, descarga, carga e saída do navio, a atividade que afeta de forma mais significativa este tempo é a operação de carga, sendo a mais sensível para a eficiência do processo, pois pode impactar no tempo de operação e, por consequência, no custo da mesma (CHOO, 2005). O planejamento do carregamento, definido pelo plano de estiva, visa minimizar o tempo e o custo da operação no terminal.

O objetivo do plano de estiva para navios porta-contêiner é determinar como armazenar um conjunto de contêineres em espaços disponíveis do navio, respeitando as limitações estruturais do mesmo, de forma a realizar operações de descarga, remanejamento e carregamento de contêineres de forma eficiente, obtendo, assim, rapidez na operação e maior rotatividade de embarcações (CHOO, 2005; SCIAMACHEN e TANFANI, 2004). Diferentes abordagens relacionadas ao PPE vêm sendo propostas na literatura, sendo aqui descritos apenas os trabalhos cujos resultados ou técnicas são relevantes para esta pesquisa.

Avriel e Penn (1993) apresentaram uma formulação linear binária para a solução do problema de planejamento de estiva. O modelo, aplicado ao GAMS, mostrou que a formulação exata é limitada devido ao número de variáveis envolvidas, sendo, então,

desenvolvidas técnicas heurísticas aproximativas que possibilitaram o planejamento da estiva de um número muito maior de contêineres.

Dubrovsky *et al.* (2002) desenvolveram um AG a fim de minimizar o número de movimentos de contêineres para resolver o problema de planejamento de estiva, incluindo algumas restrições de estabilidade do navio. Os autores reduziram significativamente o espaço de busca através de uma codificação compacta e eficiente da solução, obtendo boas soluções para navios de capacidade de 1000 TEU's. Nesta abordagem, é proposta uma implementação paralela do Algoritmo Genético com múltiplos processadores.

No trabalho de Avriel *et al.* (1998), foi proposto um modelo de programação inteira binária baseado na formulação matemática de Avriel e Penn (1993) para a modelagem do PPE. As restrições que foram estudadas se referem à alocação de contêineres em pilhas, a garantia de um contêiner ocupar no máximo uma posição em cada segmento de rota, movimentos de troca (remanejamentos) e capacidade do navio.

A modelagem do problema baseia-se em um navio porta-contêiner que possui apenas uma baía, com estrutura celular. Na prática, uma baía tem formato tridimensional, contudo, para este modelo, a geometria do navio foi considerada bidimensional, uma vez que não foram consideradas restrições de estabilidade, a fim de facilitar a utilização de matrizes para representação da estrutura em estudo. 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 pode alocar é de  $R \times C$  contêineres.

O navio inicia sua rota no Porto 1 com sua baía vazia, em seguida visita os portos 2, 3, ...,  $N$ . Em cada porto  $i = 1, \dots, N - 1$ , contêineres com destino  $j = i + 1, \dots, N$  podem ser carregados. No último porto  $N$ , o navio deve ser totalmente descarregado. O modelo define também a matriz de transporte,  $T = [T_{ij}]$  de tamanho  $(N - 1) \times (N - 1)$ , onde os elementos  $T_{ij}$  representam as quantidades 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 com movimentação em  $v$ , e 0 em caso contrário. De maneira similar, é definida uma variável  $y_i(r, c) = 1$ , se, partindo do porto  $i$ , a posição  $(r, c)$  estiver ocupada por um contêiner, e 0 se não estiver ocupada.

Para a referida abordagem, assume-se que o custo de movimentação por contêiner é unitário em todos os portos. 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 (10):

$$\text{Min } f(x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \sum_{v=i+1}^{j-1} \sum_{r=1}^R \sum_{c=1}^C x_{ijv}(r, c) \quad (10)$$

sujeita às restrições definidas nas equações (11) – (15).

A equação (11) determina  $T_{ij}$ , ou seja, a quantidade de contêineres carregados no Porto de origem  $i$  e descarregados no Porto de destino  $j$ .

$$\sum_{v=i+1}^j \sum_{r=1}^R \sum_{c=1}^C x_{ijv}(r, c) - \sum_{k=1}^{i-1} \sum_{r=1}^R \sum_{c=1}^C x_{kji}(r, c) = T_{ij}; i = 1, \dots, N-1; j = i+1, \dots, N. \quad (11)$$

A equação (12) 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.

$$\sum_{k=1}^i \sum_{j=i+1}^N \sum_{v=i+1}^j x_{kji}(r, c) = y_i(r, c); i = 1, \dots, N-1; r = 1, \dots, R; c = 1, \dots, C. \quad (12)$$

A restrição (13) 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, o lado esquerdo da expressão será negativo, indicando a violação da restrição.

$$y_i(r, c) - y_i(r+1, c) \geq 0; i = 1, \dots, N-1; r = i, \dots, R-1; c = 1, \dots, C. \quad (13)$$

Na equação (14) 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  $j$ .

$$\sum_{i=1}^{j-1} \sum_{p=j}^N x_{ipj}(r, c) + \sum_{i=1}^{j-1} \sum_{p=j+1}^N \sum_{v=j+1}^p x_{ipv}(r+1, c) \leq 1; \quad (14)$$

$$j = 2, \dots, N; r = 1, \dots, R-1; c = 1, \dots, C.$$

Por fim, a restrição (15) garante a integralidade das variáveis.

$$x_{ijv}(r, c) = 0 \text{ ou } 1; \quad y_i(r, c) = 0 \text{ ou } 1. \quad (15)$$

Na abordagem de Avriel *et al.* (1998), entende-se que o modelo binário é limitado em virtude do grande número de variáveis binárias. Por esse motivo, propôs-se o algoritmo chamado de heurística suspensória, no qual são criadas  $N-1$  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 nos portos  $i = 1, \dots, N-1$ . O plano de estiva é realizado para cada porto, onde as matrizes são configuradas nas etapas de *hanging*, *filling* e *dropping*, adequando a matriz de carregamento a cada porto, considerando os descarregamentos feitos no percurso.

Os autores Wilson e Roach (1999) empregaram o método *Branch and Bound* e a metaheurística BT compondo um método híbrido para solucionar o PPE. 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 BT 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 facilita o processo de otimização na configuração de arrumação de contêineres.

No artigo de Ambrosino *et al.* (2006), é empregado um modelo de programação inteira binária que consiste em minimizar o tempo total de carga dos navios, levando em conta restrições de peso, tamanho e estabilidade. Para a resolução do PPE, são seguidos três passos. Primeiramente, é definida uma árvore de ramificações, a qual divide o navio em porções diferentes, onde contêineres são associados a diferentes subconjuntos de baias. 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 associado. 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 encontrada.

No trabalho proposto por Azevedo *et al.* (2009), o objetivo da abordagem é minimizar os remanejamentos de contêineres, sendo empregado o AG na resolução do PPE. Este estudo teve como contribuição principal o emprego de um sistema de regras de carregamento e descarregamento de contêineres para codificação da solução, possibilitando a geração somente de soluções factíveis, reduzindo drasticamente o espaço de busca e, por consequência, o custo computacional.

### 3.3 A estabilidade no carregamento de navios

Diversos autores propuseram modelos de solução para o PPE considerando também a estabilidade. Os trabalhos que mais se aproximam desta pesquisa, e servem de base para a mesma, são descritos a seguir.

Em Azevedo *et al.* (2011), é feita uma generalização do modelo para o problema 2D apresentado por Avriel *et al.* (1998), sendo proposta a modelagem para o problema 3D, baseado na ideia de representar a estrutura celular completa do navio através de um vetor de matrizes, sendo que, cada matriz bidimensional, representaria uma baia. A formulação proposta do problema 3D assume a função objetivo composta de duas parcelas, conforme a equação (16),

$$\text{Min } f(x) = \alpha\phi_1(x) + \beta\phi_2(y) \quad (16)$$

e sujeita às restrições definidas pelas equações (22) - (26).

Na formulação apresentada na equação (16), a FO é formada por dois critérios distintos: o primeiro,  $\phi_1(x)$ , que depende da movimentação, e o segundo,  $\phi_2(y)$ , que está relacionado com a estabilidade, sendo combinados através do emprego de valores escalares  $\alpha$  e  $\beta$ , que representam os pesos de cada parcela no cálculo da FO, aplicados dentro de um modelo com dois objetivos.

Assumindo que o custo de movimentação é unitário e igual para todos os portos, tem-se  $\phi_1(x)$  como custo total, conforme equação (17):

$$\phi_1(x) = \sum_{i=1}^{N-1} \left[ \sum_{j=i+1}^N \left[ \sum_{v=i+1}^{j-1} \left[ \sum_{r=1}^R \sum_{c=1}^C \sum_{d=1}^D x_{ijv}(r, c, d) \right] \right] \right] \quad (17)$$

Por sua vez, o termo  $\phi_2(y)$  refere-se à estabilidade do navio, e é estimado como a soma dos quadrados dos desvios entre as coordenadas do centro de massa ( $xcm_{d,i}$  e  $zcm_{d,i}$ ) de cada baía e do centro geométrico da mesma. Assim, a estabilidade é dada pela equação (18):

$$\phi_2(y) = \sum_{i=1}^N \sum_{d=1}^D (\Delta xcm_{d,i})^2 + \sum_{d=1}^D (\Delta zcm_{d,i})^2, \quad (18)$$

onde cada termo é detalhado nas equações (19) - (21):

$$\Delta zcm_{d,i} = zcm_{d,i} - \frac{R}{2}, \Delta xcm_{d,i} = xcm_{d,i} - \frac{C}{2}; \quad (19)$$

$$zcm_{d,i} = \left( \sum_{r=1}^R \sum_{c=1}^C (y_i(r, c, d) \times (r - 0,5)) \right) \div \left( \sum_{r=1}^R \sum_{c=1}^C y_i(r, c, d) \right); \quad (20)$$

$$xcm_{d,i} = \left( \sum_{c=1}^C \sum_{r=1}^R (y_i(r, c, d) \times (c - 0,5)) \right) \div \left( \sum_{c=1}^C \sum_{r=1}^R y_i(r, c, d) \right). \quad (21)$$

A equação (22) determina o valor de  $T_{ij}$ , ou seja, a quantidade de contêineres carregados no Porto de origem  $i$  e descarregados no Porto de destino  $j$ .

$$\sum_{v=i+1}^j \sum_{r=1}^R \sum_{c=1}^C \sum_{d=1}^D x_{ijv}(r, c, d) - \sum_{k=1}^{i-1} \sum_{r=1}^R \sum_{c=1}^C \sum_{d=1}^D x_{kji}(r, c, d) = T_{ij};$$

$$i = 1, \dots, N - 1; j = i + 1, \dots, N; d = 1, \dots, D \quad (22)$$

A equação (23) indica se a posição  $(r, c, d)$  está ocupada ou desocupada em cada segmento de rota, garantindo que não existem rotas com mais de um contêiner nesta posição.

$$\sum_{k=1}^i \sum_{j=i+1}^N \sum_{v=i+1}^j x_{kji}(r, c, d) = y_i(r, c, d);$$

$$i = 1, \dots, N - 1; r = i, \dots, R; c = i, \dots, C; d = 1, \dots, D. \quad (23)$$

A restrição da equação (24) garante o carregamento dos contêineres em pilhas, uma vez que existindo contêiner na posição  $(r + 1, c, d)$ , então também deverá existir um contêiner em  $(r, c, d)$ .

$$y_i(r, c, d) - y_i(r + 1, c, d) \geq 0; i = 1, \dots, N - 1; r = i, \dots, R; c = 1, \dots, C; d = 1, \dots, D. \quad (24)$$

Na equação (25), é definido o movimento de remanejamento. Se um contêiner que ocupa a posição  $(r, c, d)$  é descarregado no porto  $j$ , então, ou não existem contêineres acima dele, ou o índice  $v$  do contêiner que ocupa o compartimento  $(r + 1, c, d)$  não é maior que  $j$ .

$$\sum_{i=1}^{j-1} \sum_{p=j}^N x_{ipj}(r, c, d) + \sum_{i=1}^{j-1} \sum_{p=j+1}^N \sum_{v=j+1}^p x_{ipv}(r+1, c, d) \leq 1;$$

$$j = 2, \dots, N; r = 1, \dots, R-1; c = 1, \dots, C; d = 1, \dots, D. \quad (25)$$

Por fim, a restrição (26) garante a integralidade das variáveis.

$$x_{ijv}(r, c, d) = 0 \text{ ou } 1; \quad y_i(r, c, d) = 0 \text{ ou } 1. \quad (26)$$

Em relação ao trabalho de Imai *et al.* (2006), os autores propuseram uma abordagem multiobjetivo para satisfazer os objetivos de estabilidade e minimização de remanejamentos, considerando os princípios de estabilidade do navio baseados em altura metacêntrica *GM* (apresentado na seção 2.3.3), *list* (apresentado na seção 2.3.6) e *trim* (apresentado na seção 2.3.7). De acordo com o autor, os ângulos de *list* e *trim* podem ser compensados com o uso de tanques de lastro, sendo *GM* o fator de estabilidade mais importante.

Para o fator *GM*, foi utilizada uma maximização da medida, já que a estabilidade está ligada diretamente à distância entre os pontos que definem esta medida, considerando assim apenas o produto do peso e da distância do contêiner em relação a *G* e desconsiderando o peso total das cargas. Assim, a formulação de maximização de *GM* é representada na equação (27):

$$\text{Max } GM = \sum_{i=1}^N \sum_{j=1}^N w_i \times lh_j \times x_{ij}, \quad (27)$$

onde:

$i$  é a posição do contêiner na pilha do pátio;

$j$  é a posição do contêiner no navio;

$N$  é o número de contêineres a serem carregados;

$w_i$  é a massa do contêiner  $i$ ;

$lh_j$  é a distância vertical entre o centro de gravidade do navio ( $G_0$ ) e o contêiner  $j$ ;

$x_{ij}$  determina se o contêiner na posição  $i$  do pátio foi carregado na posição  $j$  do navio;

e sujeita às restrições definidas pelas equações (28) - (30).

A formulação de estabilidade do trabalho de Imai *et al.* (2006) leva em consideração a posição em que o contêiner se encontra na pilha do pátio. Assim, as equações (28) e (29) assumem valor 1 se o contêiner na posição  $i$  do pátio for carregado na posição  $j$  do navio, caso contrário, assume valor 0. A equação (30) garante que as variáveis sejam binárias.

$$\sum_{j=1}^N x_{ij} = 1, \forall i; \quad (28)$$

$$\sum_{i=1}^N x_{ij} = 1, \forall j; \quad (29)$$

$$x_{ij} = \{0,1\}, \forall i, j. \quad (30)$$

Para as medidas *list* e *trim*, também foram implementados os mesmos cálculos propostos por Imai *et al.* (2006). Uma particularidade destas medidas é que, para o seu cálculo, é considerado apenas o valor absoluto, pois o ângulo de *list* e a medida de *trim* podem ser positivos ou negativos e, para ambos, a estabilidade é inversamente proporcional ao módulo destas medidas. Para as formulações de *list* e *trim*, foram utilizados, respectivamente, os seguintes modelos descritos pelas equações (31) e (32):

$$\text{Min } list = \left| \sum_{i=1}^N \sum_{j=1}^N w_i \times lw_j \times x_{ij} \right|, \quad (31)$$

$$\text{Min } trim = \left| \sum_{i=1}^N \sum_{j=1}^N w_i \times ll_j \times x_{ij} \right|, \quad (32)$$

onde:

$i$  é o contêiner na pilha do pátio;

$j$  é a posição do contêiner no navio;

$N$  é o número de contêineres a serem carregados;

$w_i$  é a massa do contêiner  $i$ ;

$lw_j$  é a distância horizontal transversal entre o centro de gravidade deslocado do navio ( $G$ ) e a posição  $j$ ;

$ll_i$  é a distância horizontal longitudinal entre o centro de gravidade deslocado do navio ( $G$ ) e a posição  $i$ ;

$x_{ij}$  determina se o contêiner  $i$  foi carregado na posição  $j$ ;

e sujeitas às mesmas restrições descritas nas equações (28) - (30).

No trabalho apresentado Imai *et al.* (2006), em contraponto à aplicação da formulação de estabilidade de maneira congruente à literatura técnica especializada, o autor considera o cálculo do número de movimentos de forma aproximada, utilizando o número estimado de remanejamentos, sendo este ponto uma das lacunas deixadas por este trabalho.

Por fim, apresentadas as formulações e abordagens encontradas na literatura, bem como as lacunas da literatura correlata, fica claro que o problema em estudo tem grande relevância para a comunidade científica e que margem para contribuição. Assim, o trabalho aqui proposto visa corroborar para a evolução das técnicas utilizadas na solução do PPE, utilizando-se de modelos já abordados e propondo novas estratégias de solução (otimização).

## 4 METODOLOGIA

Neste capítulo, é apresentada a modelagem matemática do PPE que foi adotada neste trabalho, com o objetivo de minimizar o número de movimentos e otimizar as medidas de estabilidade, utilizando a codificação baseada em regras. Serão detalhadas, também neste capítulo, as ferramentas computacionais adotadas para a solução, sendo definida a forma de emprego das mesmas e a interação entre elas. Utilizando os conceitos de AG, BT e algoritmos híbridos explanados no capítulo de fundamentação teórica, é apresentada aqui uma abordagem com a codificação orientada para a solução do problema alvo deste estudo.

Baseando-se no histórico das pesquisas relacionadas, apresentado na revisão bibliográfica, este capítulo apresenta uma proposta de solução com contribuição científica para o PPE, com o objetivo de encontrar resultados melhores que os apresentados na literatura até o presente momento.

### 4.1 Modelo Matemático

A modelagem do problema combina dois objetivos importantes e conflitantes: a redução do número de remanejamentos de contêineres ao longo da rota realizada pelo navio, definida por um determinado número de portos ( $N$ ), e a medida de estabilidade do arranjo de contêineres. A formulação matemática da abordagem proposta por esta pesquisa busca preencher as lacunas deixadas nos trabalhos de Azevedo *et al.* (2011) e Imai *et al.* (2006), adaptando os conceitos apresentados nos mesmos para atender ao modelo proposto.

A principal diferença entre a formulação da estabilidade de Azevedo *et al.* (2011) e de Imai *et al.* (2006) é no enfoque dado por este último trabalho, englobando os três conceitos de estabilidade na FO do modelo (a altura metacêntrica, o ângulo de *list* e a medida de *trim*), estando de acordo com os conceitos matemáticos empregados no cálculo da estabilidade padrão e cujo detalhamento é encontrado na literatura especializada (BARRASS e DERRETT, 2006). Contudo, o trabalho de Imai *et al.* (2006) apresenta a quantificação do número de remanejamentos de forma estimada, o que é corrigido com a combinação da primeira parcela da formulação de Azevedo *et al.* (2011), referente à minimização de movimentação e representada pela equação (17).

Para que o modelo de minimização de movimentos de Azevedo *et al.* (2011) seja adaptado à formulação de estabilidade proposta por Imai *et al.* (2006), é necessário que a mesma seja adequada à minimização. Portanto, a partir dos conceitos de Barrass e Derrett (2006), assume-se que a instabilidade é diretamente proporcional ao *list*, ao *trim*, e ao inverso de  $GM$ . Assim, se considerarmos a equação (33):

$$GM = KM - KG, \quad (33)$$

sendo  $KM$  a distância da quilha em relação ao metacentro e  $KG$  a distância da quilha em relação ao centro geométrico do navio carregado, pode-se dizer que se deseja minimizar o

termo  $KG$ , sendo esta minimização realizada pela própria minimização de  $G$ , o que resulta em maximizar  $GM$ .

O modelo de minimização da instabilidade aplicado neste trabalho, levando em conta as adequações das funções anteriores e considerando um peso unitário para cada contêiner, resulta na formulação apresentada na equação (34):

$$\text{Min } I = G + \text{list} + \text{trim}. \quad (34)$$

Assim sendo, a FO do modelo proposto considera a minimização de movimentos de remanejamento  $Mov$  e a instabilidade  $I$ , podendo ser representada pela equação (35), na qual  $Mov$  refere-se à equação (17),  $G$  é definido pela equação (36),  $\text{list}$  é representado pela equação (37) e  $\text{trim}$  pela equação (38), sendo que cada uma das parcelas que constitui o modelo proposto é multiplicada pelos pesos respectivos pesos  $\beta$ ,  $\alpha$ ,  $\chi$  e  $\delta$ .

$$\text{Min } \text{Mov}I = \beta \times \text{Mov} + \alpha \times G + \chi \times \text{list} + \delta \times \text{trim}, \quad (35)$$

onde:

$$G = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \sum_{d=1}^D \sum_{r=1}^R \sum_{c=1}^C (KG - KM)_{ij} \times x_{ij}(d, r, c) \quad (36)$$

$$\text{list} = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \left| \sum_{d=1}^D \sum_{r=1}^R \sum_{c=1}^C lw_{ij} \times x_{ij}(d, r, c) \right| \quad (37)$$

$$\text{trim} = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \left| \sum_{d=1}^D \sum_{r=1}^R \sum_{c=1}^C ll_{ij} \times x_{ij}(d, r, c) \right| \quad (38)$$

sujeito às restrições (22) - (26) e (28) - (30).

## 4.2 Ferramentas computacionais

As seções a seguir apresentam as ferramentas computacionais utilizadas na proposta de solução, percorrendo acerca das vantagens de empregar a codificação baseada em regras e das abordagens metaheurísticas propostas neste trabalho.

### 4.2.1 Codificação por regras

Para a representação da solução do PPE, não serão utilizadas as formulações baseadas no modelo matemático binário, devido ao grande número de variáveis envolvidas. Em vez disso, será utilizada a codificação da solução baseada em regras, que define, a cada porto, como o navio será carregado ou descarregado. A representação através da codificação por regras apresenta cinco vantagens em relação ao modelo matemático binário:

- a) Permite a codificação compacta da solução, através de um vetor com  $(N - 1)$  elementos, onde cada elemento representa a estratégia que será utilizada no porto;
- b) Quantidade reduzida de variáveis, sendo o número de variáveis do problema igual ao número de estratégias  $Re_j$ , com  $j = 1, 2, \dots, N - 1$ , pois no porto  $N$ , ocorre apenas o descarregamento;
- c) Permite incorporar a experiência dos profissionais envolvidos nas operações de carga e descarga do navio, através do emprego de regras que eles costumam utilizar;
- d) Geração somente de soluções factíveis;
- e) Resolve problemas de maior porte em tempo computacional razoável.

De acordo com Azevedo *et al.* (2011), a codificação por regras é definida através da atribuição de uma estratégia para cada porto ( $Re_j$ ). Cada estratégia é formada pela combinação de uma regra de carregamento (Rc), que define como o navio é carregado, e uma regra de descarregamento (Rd), que define a forma como o mesmo é descarregado.

Para o algoritmo proposto neste trabalho, serão utilizadas as regras (ou estratégias) de carga e descarga conforme proposto em Azevedo *et al.* (2011). Serão também propostas novas regras, a fim de aumentar as possibilidades de exploração do espaço de busca, sendo esta uma das contribuições deste trabalho.

#### 4.2.1.1 Regras de carregamento e de descarregamento

Considerando o navio como uma matriz de ocupação tridimensional  $\mathbf{B}$ , que fornece a quantidade de espaços disponíveis e a localização dos contêineres no navio, cada elemento  $\mathbf{B}_{drc}$  da matriz tridimensional  $\mathbf{B}$  representa o estado de uma célula  $(d, r, c)$ , isto é, se  $\mathbf{B}_{drc} = 0$ , significa que a célula que ocupa a baía  $d$ , a linha  $r$  e a coluna  $c$ , está vazia; no caso de  $\mathbf{B}_{drc} = j$ , significa que a célula contém um contêiner, cujo destino é o porto  $j = 2, \dots, N$ .

A matriz de ocupação  $\mathbf{B}$  representa o arranjo tridimensional dos contêineres existentes no navio ( $B_p$ ) antes de chegar ao porto  $p$ . Esse arranjo é modificado em cada porto, através da variável de decisão  $Re_j$ , que define a estratégia que indica como o navio será descarregado (regra de descarregamento Rd) e, depois, carregado (regra de carregamento Rc). Com o intuito de reduzir o número de remanejamentos, ao se fazer o carregamento de contêineres num dado porto  $j$ , deve-se levar em consideração os contêineres que já estão no navio, porque foram embarcados nos portos anteriores (portos de 1 até  $j - 1$ ), e que possuem como destino os portos de  $j + 1$  até  $N$ .

A seguir, são detalhadas as regras de carga Rc1 a Rc6 e as regras de descarga Rd1 e Rd2, baseadas no trabalho de Azevedo *et al.* (2011), bem como as regras de carga Rc7 a Rc12 e de descarga Rd3, propostas neste trabalho. Na explicação do uso das regras, considera-se

que o navio está no porto 1 para as regras de carregamento e no porto 2 para as regras de descarregamento, baseando-se na Matriz de Transporte  $T$  da Figura 12.

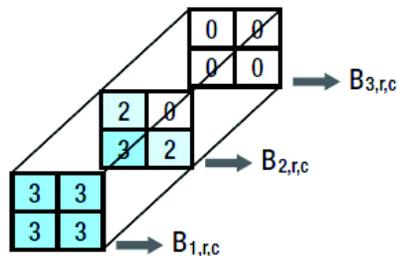
Figura 12 - Matriz de Transporte  $T$

	D2	D3	D4	D5
O1	2	5	0	0
O2	0	2	3	1
O3	0	0	2	2
O4	0	0	0	1

Fonte: Azevedo *et al.* (2011).

- a) Regra de carregamento Rc1: Esta regra preenche a matriz de ocupação  $\mathbf{B}$  (no porto  $p$ ) por baía, começando pela primeira linha da primeira baía e indo até a última linha desta mesma baía, da esquerda para a direita, iniciando o carregamento de cada baía pelos contêineres cujo destino é mais distante. O resultado da aplicação desta regra é mostrado na Figura 13.

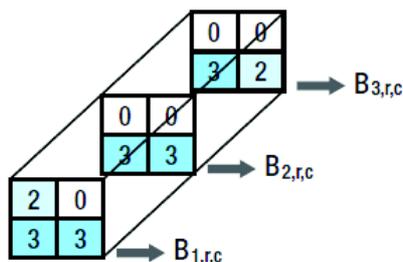
Figura 13 – Matriz de ocupação  $\mathbf{B}$  após aplicada regra de carregamento Rc1



Fonte: Azevedo *et al.* (2011).

- b) Regra de carregamento Rc2: Esta regra preenche a matriz de ocupação  $\mathbf{B}$  (no porto  $p$ ) por linha, começando pela primeira linha da primeira baía e indo até a última linha da última baía, da esquerda para a direita, iniciando o carregamento pelos contêineres cujo destino é mais distante. O resultado é visualizado na matriz  $\mathbf{B}$  da Figura 14.

Figura 14 - Matriz de ocupação  $\mathbf{B}$  após aplicada regra de carregamento Rc2

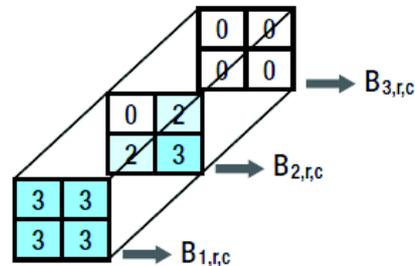


Fonte: Azevedo *et al.* (2011).

- c) Regra de carregamento Rc3: Esta regra é o espelho da regra de carregamento Rc1 na direção longitudinal do navio. No porto  $p$ , a matriz de ocupação  $\mathbf{B}$  será

preenchida por baia, da direita para a esquerda, iniciando o carregamento pelos contêineres cujo destino é mais distante. O resultado é visualizado na matriz **B** da Figura 15.

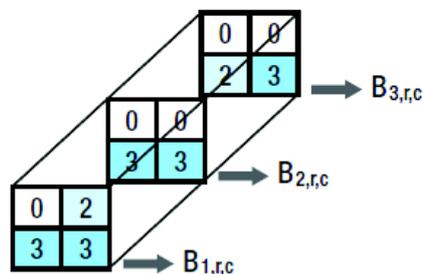
Figura 15 - Matriz de ocupação **B** após aplicada regra de carregamento Rc3



Fonte: Azevedo *et al.* (2011).

- d) Regra de carregamento Rc4: Esta regra é o espelho da regra de carga Rc2 na direção longitudinal do navio. No porto  $p$ , a matriz de ocupação **B** será preenchida por linha, da direita para a esquerda, iniciando o carregamento de cada baia pelos contêineres cujo destino é mais distante. O resultado é visualizado na matriz **B** da Figura 16.

Figura 16 - Matriz de ocupação **B** após aplicada regra de carregamento Rc4



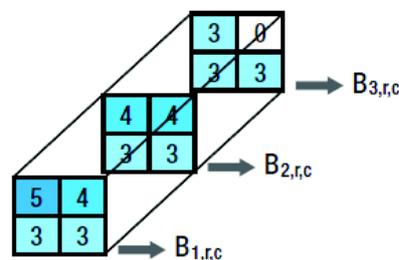
Fonte: Azevedo *et al.* (2011).

- e) Regra de carregamento Rc5: Nesta regra, o preenchimento da matriz de ocupação **B** em um porto  $p$ , é feito a partir da primeira baia até uma linha  $\theta p$ , começando pela coluna da esquerda e alocando primeiro os contêineres cujos destinos são mais distantes. A linha  $\theta p$  é resultante da soma do total de contêineres que estavam no navio e foram embarcados nos portos anteriores; menos a quantidade de contêineres que serão desembarcados em  $p$ , mais a quantidade de contêineres a serem embarcados em  $p$ , dividido pelo número de baias da matriz **B**. O número dessa linha pode ser calculado sobre a matriz de transporte  $T$ , através da equação (39), onde  $p$  é o porto atual do navio,  $T_{ij}$  é o número total de contêineres a serem embarcados no porto  $i$  com destino ao porto  $j$ ,  $D$  é o número total de baias representado pela matriz **B** de ocupação do navio e  $C$  é o número total de colunas do navio.

$$\theta p = \left\lceil \frac{\sum_{i=1}^p \sum_{j=p+1}^N T_{ij}}{D \times C} \right\rceil \quad (39)$$

Admitindo-se que, no momento em que o navio chega ao porto 2, o mesmo esteja com a matriz **B** conforme a Figura 14 e que os contêineres cujo destino é o porto 2 sejam descarregados, ao aplicar a regra de carga Rc5 para carregar o navio antes de seguir viagem para o porto 3, a matriz de ocupação **B** resultante é representada pela Figura 17, pois, no porto 3, são carregados dois contêineres cujo destino é o porto 3, três contêineres cujo destino é o porto 4 e um contêiner cujo destino é o porto 5.

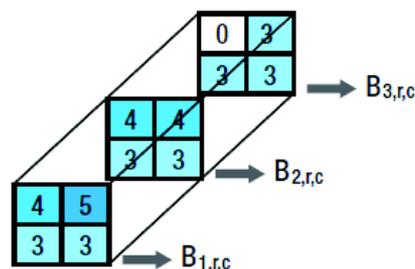
Figura 17 - Matriz de ocupação **B** após aplicada regra de carregamento Rc5



Fonte: Azevedo *et al.* (2011).

- f) Regra de carregamento Rc6: Esta regra é o espelho da regra Rc5 na direção longitudinal do navio. Ela faz o preenchimento da matriz de ocupação **B**, em um porto  $p$ , preenchendo, a partir da primeira baía, cada coluna até a linha  $\theta p$ , determinada pela equação (39), começando pela coluna da direita e colocando primeiramente os contêineres cujos destinos são os portos mais distantes, conforme Figura 18.

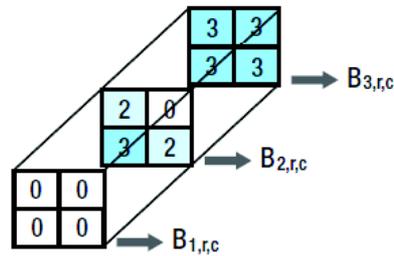
Figura 18 - Matriz de ocupação **B** após aplicada regra de carregamento Rc6



Fonte: Azevedo *et al.* (2011).

- g) Regra de carregamento Rc7: Esta regra é o espelho da regra de carga Rc1 na direção transversal do navio. Ela preenche a matriz de ocupação **B** (no porto  $p$ ) por baía, começando pela primeira linha da última baía e indo até a última linha desta mesma baía, da esquerda para a direita, iniciando o carregamento de cada baía pelos contêineres cujo destino é mais distante. O resultado da aplicação desta regra é mostrado na Figura 19.

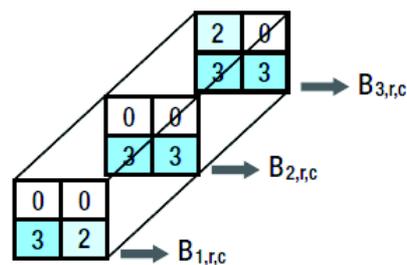
Figura 19 – Matriz de ocupação **B** após aplicada regra de carregamento Rc7



Fonte: Elaborado pelo autor.

- h) Regra de carregamento Rc8: Esta regra é o espelho da regra de carga Rc2 na direção transversal do navio. Ela preenche a matriz de ocupação **B** (no porto  $p$ ) por linha, começando pela primeira linha da última baia e indo até a primeira baia, da esquerda para a direita, alocando primeiramente os contêineres cujo destino são os portos mais distantes. O resultado é visualizado na matriz **B** da Figura 20.

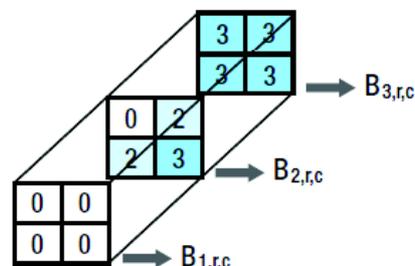
Figura 20 - Matriz de ocupação **B** após aplicada regra de carregamento Rc8



Fonte: Elaborado pelo autor.

- i) Regra de carregamento Rc9: Esta regra é o espelho da regra de carga Rc3 na direção transversal do navio. No porto  $p$ , a matriz de ocupação **B** será preenchida da última para a primeira baia, da direita para a esquerda, iniciando o carregamento de cada baia pelos contêineres cujo destino é mais distante. O resultado é visualizado na matriz **B** da Figura 21.

Figura 21 - Matriz de ocupação **B** após aplicada regra de carregamento Rc9

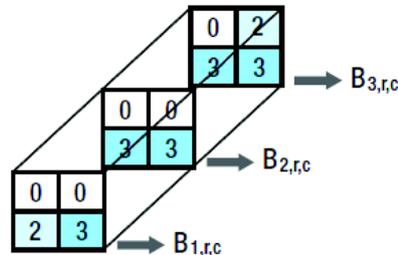


Fonte: Elaborado pelo autor.

- j) Regra de carregamento Rc10: Esta regra é o espelho da regra de carga Rc4 na direção transversal do navio. No porto  $p$ , a matriz de ocupação **B** será preenchida da última para a primeira baia por linha, da direita para a esquerda, alocando

primeiramente os contêineres cujo destino é mais distante. O resultado é visualizado na matriz **B** da Figura 22.

Figura 22 - Matriz de ocupação **B** após aplicada regra de carregamento Rc10

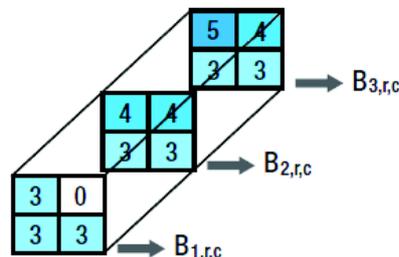


Fonte: Elaborado pelo autor.

- k) Regra de carregamento Rc11: Esta regra é o espelho da regra de carga Rc5 na direção transversal do navio. Nesta regra, o preenchimento da matriz de ocupação **B** em um porto  $p$ , é feito da última para a primeira baía até uma linha  $\theta p$ , começando pela coluna da esquerda e alocando primeiro os contêineres cujos destinos são os portos mais distantes. A linha  $\theta p$  é calculada sobre a matriz de transporte  $T$ , através da equação (39), onde  $p$  é o porto atual do navio,  $T_{ij}$  é o número total de contêineres a serem embarcados no porto  $i$ , com destino ao porto  $j$ ,  $D$  é o número total de baías representado pela matriz **B** de ocupação do navio e  $C$  é o número total de colunas do navio.

Assumindo que, quando o navio chega ao porto 2, ele esteja com a matriz **B** conforme a Figura 22 e que os contêineres cujo destino é o porto 2, são descarregados. Aplicando-se a regra Rc11 para carregar o navio antes de seguir viagem para o porto 3, a matriz **B** resultante é apresentada na Figura 23, pois no porto 3, são carregados dois contêineres cujo destino é o porto 3, três contêineres cujo destino é o porto 4 e um contêiner cujo destino é o porto 5.

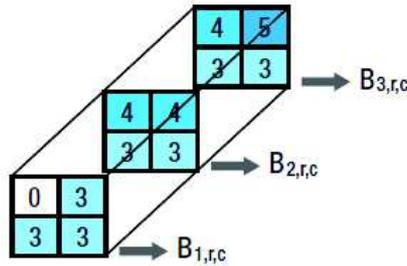
Figura 23 - Matriz de ocupação **B** após aplicada regra de carregamento Rc11



Fonte: Elaborado pelo autor.

- l) Regra de carregamento Rc12: Esta regra é o espelho da regra de carga Rc6 na direção transversal do navio. Ela faz o preenchimento da matriz de ocupação **B**, em um porto  $p$ , da última para a primeira baía, preenchendo cada coluna até a linha  $\theta p$  (equação (39)), começando pela coluna da direita e alocando primeiramente os contêineres cujos destinos são os portos mais distantes, resultando na Figura 24.

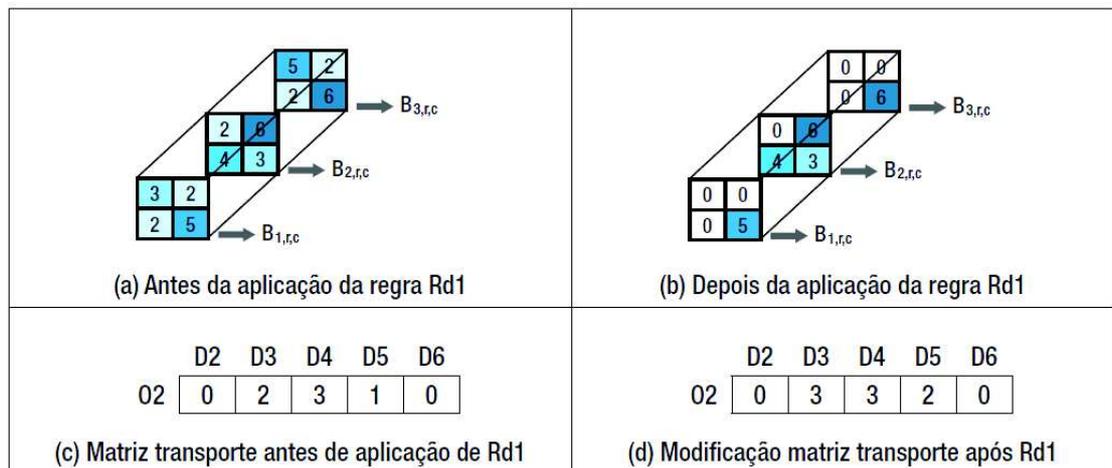
Figura 24 - Matriz de ocupação B após aplicada regra de carregamento Rc12



Fonte: Elaborado pelo autor.

m) Regra de descarregamento Rd1: Nesta regra, quando o navio chega a um porto  $p$ , são removidos todos os contêineres cujo destino é  $p$  e todos os contêineres que estão acima, bloqueando os contêineres do porto  $p$ , e cujos destinos são os portos  $p + j$ , para  $j = 1, \dots, N - 1$  tal que  $(p + j) \leq N$ . Assumindo que, ao se chegar ao porto 2, a matriz de ocupação  $\mathbf{B}$  seja a representada na Figura 25(a). De acordo com esta regra, a matriz  $\mathbf{B}$  ficaria como mostrada na Figura 25(b). Além disso, após a aplicação da regra, a matriz de transporte  $T$  tem que ser atualizada, porque foram retirados um contêiner com destino ao porto 3 e um contêiner com destino ao porto 5, que devem ser novamente carregados e, portanto, serão adicionados na matriz de transporte relativa ao porto 2. A Figura 25(c) e a Figura 25(d) mostram a linha respectiva da matriz de transporte, antes e depois da aplicação desta regra.

Figura 25 - Matriz de ocupação B após aplicar regra de descarregamento Rd1



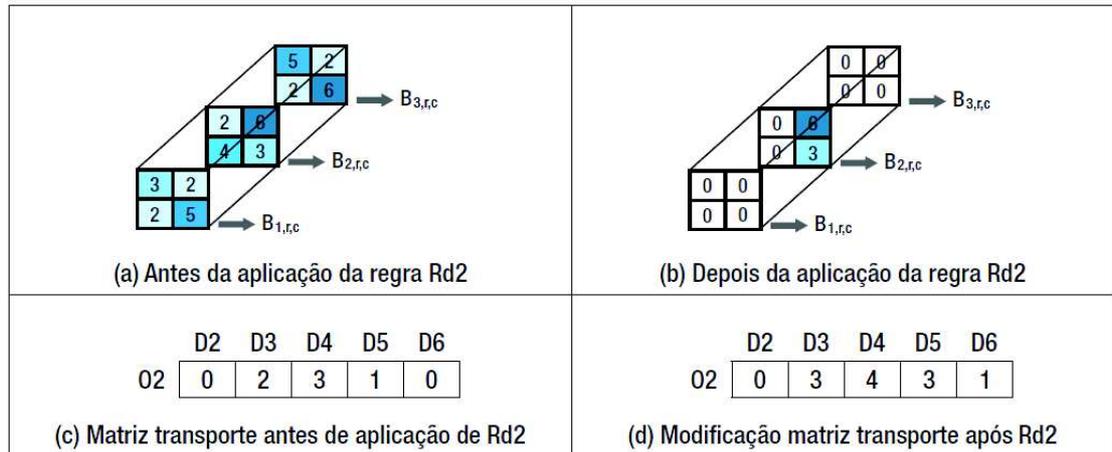
Fonte: Azevedo *et al.* (2011).

n) Regra de descarregamento Rd2: Nesta regra, quando o navio chega ao porto  $p$ , são removidos todos os contêineres de cada pilha onde exista pelo menos um contêiner cujo destino é  $p$ . Assumindo que, ao se chegar ao porto 2, a matriz de ocupação  $\mathbf{B}$  seja a representada na Figura 26(a). De acordo com esta regra, a matriz  $\mathbf{B}$  ficaria como mostrada na Figura 26(b). Além disso, após a aplicação da regra, a matriz de transporte  $T$  tem que ser atualizada, porque foram retirados um contêiner com destino ao porto 3, um com destino ao porto 4, dois com destino ao porto 5 e um

com destino ao porto 6, que devem ser novamente carregados e, portanto, serão adicionados na matriz de transporte relativa ao porto 2.

A Figura 26(c) e a Figura 26(d) mostram a linha respectiva da matriz de transporte, antes e depois da aplicação desta regra.

**Figura 26 - Matriz de ocupação B após aplicar regra de descarregamento Rd2**



Fonte: Elaborado pelo autor.

- o) Regra de descarregamento Rd3: Nesta regra, quando o navio chega ao porto  $p$ , todos os contêineres são removidos do navio para que todas as pilhas sejam reordenadas por alguma regra de carregamento, a ser aplicada posteriormente. Assim como foi feito nas regras Rd1 e Rd2, a matriz de transporte  $T$  deve ser atualizada com a inclusão dos contêineres removidos e cujos destinos são os portos além do porto 2.

As combinações de regras, que formam as possíveis estratégias  $Re_j$  aplicadas em cada porto, são representadas na Tabela 3.

**Tabela 3 - Regras utilizadas em cada porto**

Regras $Re_j$ utilizadas em cada porto $j$												
Regra de carregamento	Rc1	Rc1	Rc2	Rc2	Rc3	Rc3	Rc4	Rc4	Rc5	Rc5	Rc6	Rc6
Regra de descarregamento	Rd1	Rd2										
Regra $Re_j$ usada no porto $p$	$Re_1$	$Re_2$	$Re_3$	$Re_4$	$Re_5$	$Re_6$	$Re_7$	$Re_8$	$Re_9$	$Re_{10}$	$Re_{11}$	$Re_{12}$
Regras $Re_j$ utilizadas em cada porto $j$												
Regra de carregamento	Rc7	Rc7	Rc8	Rc8	Rc9	Rc9	Rc10	Rc10	Rc11	Rc11	Rc12	Rc12
Regra de descarregamento	Rd1	Rd2										
Regra $Re_j$ usada no porto $p$	$Re_{13}$	$Re_{14}$	$Re_{15}$	$Re_{16}$	$Re_{17}$	$Re_{18}$	$Re_{19}$	$Re_{20}$	$Re_{21}$	$Re_{22}$	$Re_{23}$	$Re_{24}$
Regras $Re_j$ utilizadas em cada porto $j$												
Regra de carregamento	Rc1	Rc2	Rc3	Rc4	Rc5	Rc6	Rc7	Rc8	Rc9	Rc10	Rc11	Rc12
Regra de descarregamento	Rd3											
Regra $Re_j$ usada no porto $p$	$Re_{25}$	$Re_{26}$	$Re_{27}$	$Re_{28}$	$Re_{29}$	$Re_{30}$	$Re_{31}$	$Re_{32}$	$Re_{33}$	$Re_{34}$	$Re_{35}$	$Re_{36}$

Fonte: adaptado de Azevedo *et al.* (2011).

## 4.2.2 A metaheurística Algoritmo Genético

Uma das abordagens propostas para a solução do PPE é a metaheurística AG. Para a implementação do AG, é necessário definir a forma como ele será aplicado ao problema, especificando os operadores genéticos e os parâmetros empregados. A seguir, serão detalhados a forma de codificação da solução para o problema abordado e o critério de parada, bem como os operadores de seleção, cruzamento e mutação.

### 4.2.2.1 Codificação da solução

A codificação empregada para a representação do cromossomo (ou do indivíduo) é implementada através do uso das combinações de regras de carga e descarga, propostas inicialmente por Azevedo *et al.* (2009) e utilizadas por Azevedo *et al.* (2011) e Carraro *et al.* (2013). Uma solução compacta pode ser representada através de um vetor  $S$  cujo elemento  $s[j]$  de valor igual a  $k$  indica qual regra de carga e descarga foi utilizada para o porto  $j$  (AZEVEDO *et al.*, 2009).

Considerando um exemplo no qual um navio sai do seu porto de origem e visita outros 3 portos (portos 2 a 4), a Figura 27 indica que a estratégia de carregamento 3 foi utilizada no porto 1. Nos portos seguintes, a estratégia 12 será utilizada no porto 2 e a estratégia 7 no porto 3. Note que o porto 4 não precisa ser representado na codificação da solução, pois este é o último porto e todos os contêineres serão descarregados, não havendo carregamento, o que evidencia que a regra utilizada não irá interferir no resultado.

**Figura 27 - Codificação do Cromossomo**

3	12	7
---	----	---

Fonte: Elaborado pelo autor.

### 4.2.2.2 Operador de Seleção

O objetivo do operador de seleção é escolher os indivíduos com melhor aptidão para serem utilizados na etapa de cruzamento, com o objetivo de obter uma geração com características melhores que as gerações anteriores. Os indivíduos são selecionados de maneira estocástica, favorecendo os indivíduos mais aptos. No entanto, uma seleção adequada não pode considerar apenas os indivíduos mais aptos, pois fará com que as populações geradas percam diversidade de forma prematura (MITCHELL, 1997).

Os operadores utilizados nesta abordagem foram propostos por Mitchell (1997), e são detalhados a seguir:

- a) Seleção por torneio (*tournament selection*), no qual, após confirmada a aplicação do operador de seleção, são selecionados dois indivíduos de maneira aleatória,

classificando-os de acordo com sua aptidão. Posteriormente, a escolha do indivíduo é feita em um torneio entre os dois, no qual determina-se que o mais apto é escolhido com uma probabilidade  $rand$  e o menos apto com uma probabilidade  $1 - rand$ ;

- b) Seleção por ranqueamento (*rank selection*), no qual os indivíduos são classificados segundo a sua função de aptidão, sendo aplicada a probabilidade de escolha dos mesmos proporcionalmente ao valor de aptidão de cada um, favorecendo os mais aptos.

Na aplicação desenvolvida neste estudo, foi definida uma probabilidade de 50% para cada operador de seleção. Dessa forma, ao sortear um número aleatório com distribuição uniforme no intervalo  $[0,1]$ , se o mesmo assumir valor menor que 0,5, é utilizado o operador de seleção torneio. Caso contrário, a seleção é realizada por ranqueamento. Para o operador de seleção por torneio, foi aplicada uma probabilidade de 75% para a escolha do indivíduo mais apto.

#### 4.2.2.3 Operador de Cruzamento

O cruzamento é o operador aplicado na etapa de reprodução, após selecionar os cromossomos Pais. Este operador combina características de dois indivíduos, contribuindo para a intensificação do AG, pois perpetua informações capazes de intensificar a busca da solução em áreas promissoras do espaço de busca. A aplicação do operador é dada através do sorteio de um número aleatório, cuja função densidade de probabilidade obedece uma distribuição uniforme, no intervalo  $[0, 1]$ , sendo posteriormente avaliado se este valor atende à taxa de cruzamento ( $pc$ ), definida previamente. Para valores acima da taxa de cruzamento, o operador não será aplicado, sendo os cromossomos filhos assumidos idênticos aos cromossomos pais (CHAN *et al.*, 1998).

O operador escolhido para o problema foi o Cruzamento Um Ponto de Corte, detalhado em Chan *et al.* (1998). O operador de Cruzamento Um Ponto de Corte consiste no sorteio de um número aleatório, com distribuição uniforme, no 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 é 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.

Na Figura 28, foi selecionada, aleatoriamente, a posição 4 como ponto de corte. Logo, o Filho 1 resulta de uma união dos genes do Pai 1 até a posição 4 e dos genes Pai 2 a partir da posição 5. 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. Os dois filhos passam então a compor a nova população.

**Figura 28 - Operador Um Ponto de Corte**

Pai 1	12	7	9	3	4	1
Pai 2	6	3	8	11	14	2
Filho 1	12	7	9	3	14	2
Filho 2	6	3	8	11	4	1

Fonte: Elaborado pelo autor.

Para a abordagem proposta neste trabalho, o intervalo dentro do qual o ponto de corte é sorteado foi definido como sendo  $[1, N - 1]$ , assumindo-se  $N$  como o número de portos que o navio irá visitar.

#### 4.2.2.4 Operador de Mutação

A aplicação da mutação foi condicionada à probabilidade de mutação definida no algoritmo, sendo realizada nos indivíduos Filhos, após a geração dos mesmos. A mutação será aplicada definindo uma porcentagem dos genes a serem modificados, resultando assim em um número de portos que terão suas regras modificadas (*ngen*). A mutação será realizada até que o número de genes mutados seja igual ao valor definido de 30% do cromossomo.

O operador de mutação empregado realiza o sorteio aleatório de um gene pertencente ao cromossomo para que tenha sua regra modificada. O valor deste gene será definido aleatoriamente dentro do intervalo  $[1, numRegras]$ , sendo *numRegras* o número máximo de regras utilizado. O operador implementado realiza um controle de modo a gerar valores com diversidade para os genes mutados, de forma que não são repetidas regras para um mesmo indivíduo. Outro recurso de controle que foi implementado é a comparação do valor anterior do gene com o valor sorteado para substituir o mesmo, devendo ser diferentes, a fim de tornar a mutação eficiente.

#### 4.2.2.5 Critério de Parada

O critério de parada adotado para a o algoritmo genético proposto foi o número máximo de gerações da população. Não foi considerado definir um limite inferior para a minimização de número de movimentos ou de instabilidade da embarcação, porquanto a finalidade da abordagem proposta é encontrar o menor valor possível para o problema em questão.

### 4.2.3 A metaheurística Busca Tabu

A outra abordagem proposta por este trabalho na solução do PPE é a aplicação da metaheurística Busca Tabu. A BT possui alguns parâmetros que necessitam ser definidos para delimitar como o algoritmo guiará sua busca pelo espaço de soluções. Estes detalhes são elucidados a seguir.

A codificação de solução utilizada na BT será a mesma utilizada no AG, sendo definida por um vetor solução  $S$  de  $N - 1$  posições, sendo  $N$  o número de portos visitados pelo navio.

#### 4.2.3.1 Estrutura de vizinhança

Na abordagem proposta neste trabalho, a estrutura de vizinhança adotada foi baseada na modificação aleatória a partir da solução atual. Primeiramente, é selecionada uma posição do vetor  $S$  de forma aleatória, segundo uma distribuição uniforme, que representa um determinado porto, constante na rota do navio, com um valor de regra associado a ele. A posição escolhida tem então seu valor substituído por uma nova regra, selecionada segundo o mesmo critério aleatório, no entanto, diferente da regra atual, a fim de não gerar vizinhos iguais à solução atual. Assim, cada vizinho gerado é diferente da solução atual em apenas um porto.

#### 4.2.3.2 Lista Tabu

A Lista Tabu (LT) é a memória recente das últimas soluções visitadas. Devido à dificuldade em armazenar as soluções completas em virtude do custo computacional envolvido (GLOVER e LAGUNA, 1997), no presente trabalho, foram armazenados os movimentos realizados pelas últimas soluções, registrando o porto que teve a sua regra alterada e a regra que era utilizada pela solução anteriormente visitada. O tamanho da LT utilizado foi determinado segundo procedimento de sintonia de parâmetros, a ser explanado a seguir.

#### 4.2.3.3 Critério de aspiração

O critério de aspiração utilizado para a abordagem proposta por este trabalho foi a condição de a nova solução possuir um valor para a FO que se aproxime mais da solução ótima do que a solução atual. Neste caso, o movimento não será considerado proibido, sendo adotado para gerar a solução atual.

#### 4.2.3.4 Critério de Parada

O algoritmo BT desenvolvido neste trabalho foi concebido com um critério de parada baseado no número de movimentos sem melhora do valor da FO para o problema. Não foi considerado definir um limite inferior para a minimização de número de movimentos ou de instabilidade, porquanto a finalidade da abordagem proposta, assim como no AG, é encontrar o menor valor possível para o problema em questão.

#### 4.2.4 Abordagem híbrida

Além dos dois algoritmos detalhados para a proposta desta solução, foi implementado também um algoritmo híbrido, combinando as duas metaheurísticas anteriormente explanadas em um modelo denominado AG+BT, cujas características são definidas, segundo a classificação de Raidl (2006), como hibridização de metaheurísticas com nível de combinação *high-level* e ordem de execução intercalada. A combinação entre os algoritmos AG e BT foi implementada de maneira sequencial nesta abordagem, mantendo, desta forma, a estrutura canônica de cada metaheurística.

O modelo de hibridização escolhido para esta abordagem tem por objetivo intensificar características específicas de cada metaheurística. Dessa forma, o AG realiza uma busca pelo espaço de soluções, com parâmetros definidos de forma a focar na característica de diversificação. A partir da melhor solução encontrada pelo AG, será realizada uma busca em trajetória, executada pela BT, com parâmetros focados, por sua vez, na intensificação da busca.

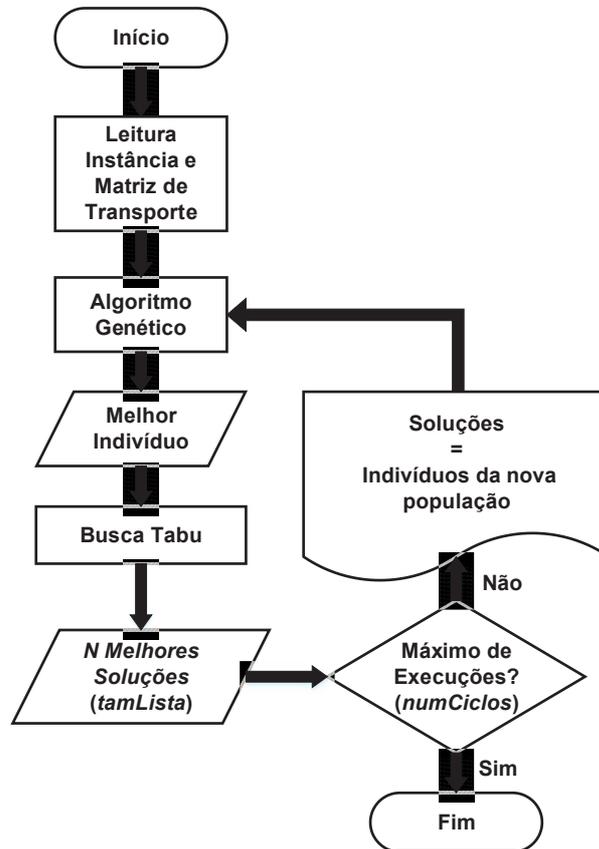
A interação entre as metaheurísticas ocorre quando os melhores resultados encontrados pela BT retornam ao AG, sendo empregados como indivíduos de uma nova população, executando assim um novo ciclo. Este procedimento é realizado repetidamente por um número de iterações, definido conforme configuração do algoritmo híbrido.

O número de soluções da BT a serem utilizadas no AG como indivíduos de uma nova população inicial é definido por um parâmetro chamado *tamLista*, sendo que o restante dos indivíduos que compõem a população é gerado aleatoriamente. O número de ciclos do AG+BT é determinado através de um parâmetro chamado *numCiclos*. Na abordagem proposta, ambos foram definidos por um procedimento de sintonia.

A hibridização proposta neste trabalho combina, segundo a classificação de Melián *et al.* (2003), uma metaheurística evolutiva com uma busca por entornos. Apesar de combinar o AG com uma busca por entornos, o AG+BT proposto não é caracterizada como Algoritmo Memético, visto que o funcionamento convencional de cada uma das metaheurísticas foi mantido. No conceito denominado Algoritmo Memético, a informação genética de cada cromossomo passa por diversos refinamentos através de Busca Local, alterando assim a estrutura do AG.

A Figura 29 mostra o fluxo de execução do algoritmo híbrido AG+BT conforme descrito anteriormente. A proposta de hibridização aqui descrita é uma das contribuições deste trabalho, visto que não foram encontrados registros da aplicação deste modelo híbrido ao problema em questão.

**Figura 29 - Fluxograma do algoritmo híbrido AG+BT**



Fonte: Elaborado pelo autor.

## 5 EXPERIMENTOS E RESULTADOS

A fim de validar as técnicas propostas, os resultados obtidos após a aplicação das mesmas devem ser comparados com resultados já conhecidos da literatura. Após a validação, são realizados experimentos com as técnicas propostas, gerando dados estatísticos de desempenho e resultados dos algoritmos.

Neste capítulo, são detalhados os procedimentos realizados na etapa de validação e experimentos do algoritmo proposto, bem como o detalhamento dos problemas-teste empregados nos experimentos. A parametrização das técnicas utilizadas para as etapas de validação e experimentos também é detalhada neste capítulo, na etapa que trata da sintonia de parâmetros.

As aplicações propostas neste estudo foram desenvolvidas em linguagem C# sobre a plataforma .NET. Os experimentos foram realizados em um computador com processador Intel Core i7 930, de arquitetura *quad-core*.

### 5.1 Tipos de Matrizes de Transporte

Para a realização dos experimentos, foram utilizados três tipos de matrizes de transporte, com problemas-teste de 10, 15, 20, 25 e 30 portos, classificadas de acordo com o tempo de permanência dos contêineres a bordo do navio. Os três tipos de matrizes de transporte são assim definidos (AVRIEL *et al.*, 1998):

- a) Distância Mista (tipo 1), sendo os elementos determinados de forma aleatória;
- b) Longa Distância (tipo 2), nas quais os contêineres permanecem a bordo por longo período;
- c) Curta Distância (tipo 3), nas quais os contêineres permanecem a bordo por um curto período.

### 5.2 Problemas-teste utilizados

A fim de possibilitar uma comparação com as técnicas propostas na literatura, os problemas-teste aplicados aos trabalhos correlatos foram aplicados também nesta pesquisa. Como a técnica desenvolvida é uma proposta que tem por objetivo melhorar os resultados obtidos no trabalho de Azevedo *et al.* (2011), os mesmos problemas-teste foram aplicados, seguindo a classificação do tipo de problema-teste de Avriel *et al.* (1998), que foi detalhada na seção anterior.

Os problemas-teste empregados nos experimentos computacionais são classificadas de acordo com o tipo de matriz de transporte e o número de portos visitados. Como já mencionado, foram empregados problemas-teste com 10, 15, 20, 25 e 30 portos, sendo

aplicados os três tipos de matriz de transporte para cada quantidade de portos visitados. Assim, foram avaliadas, com as abordagens aqui propostas, um total de 15 problemas-teste distintos em tipo de matriz de transporte e quantidade de portos.

Para a aplicação dos problemas-teste, será considerado um navio porta-contêineres com 6 baías, com uma estrutura celular de 6 linhas e 50 colunas cada uma ( $D \times R \times C = 5 \times 6 \times 50$ ), resultando na capacidade máxima de 1500 contêineres (AZEVEDO *et al.*, 2011).

### 5.3 Sintonia de parâmetros

A sintonia de parâmetros consiste em escolher valores para os parâmetros de uma metaheurística que preconizem a obtenção de resultados eficientes de otimização. Neste estudo, a definição destes valores foi realizada através da composição de cenários de testes, fixando o número de avaliações da FO como critério de parada e alterando os valores dos parâmetros, comparando, assim, os resultados. A FO empregada na sintonia assumiu apenas o critério de minimização dos movimentos e os parâmetros foram definidos extraindo os valores dos cenários que mostraram resultados mais promissores.

#### 5.3.1 Sintonia dos parâmetros do Algoritmo Genético

A sintonia dos parâmetros empregados na metaheurística AG foi obtida fixando os parâmetros que atuam diretamente sobre a quantidade de avaliações realizadas da FO, a fim de obter um número de 20000 avaliações. Estes parâmetros foram fixados em:

- a) Tamanho da população = 20;
- b) Número de gerações = 1000.

Para a metaheurística AG, os parâmetros sintonizados foram a probabilidade de mutação ( $pm$ ) e a probabilidade de cruzamento ( $pc$ ). Ambos parâmetros, combinados, compuseram 9 cenários de teste, que seguem:

- a)  $pc = 80\%$  e  $pm = 5\%$ ;
- b)  $pc = 80\%$  e  $pm = 10\%$ ;
- c)  $pc = 80\%$  e  $pm = 15\%$ ;
- d)  $pc = 90\%$  e  $pm = 5\%$ ;
- e)  $pc = 90\%$  e  $pm = 10\%$ ;
- f)  $pc = 90\%$  e  $pm = 15\%$ ;
- g)  $pc = 100\%$  e  $pm = 5\%$ ;
- h)  $pc = 100\%$  e  $pm = 10\%$ ;
- i)  $pc = 100\%$  e  $pm = 15\%$ .

Os 15 problemas-teste foram submetidos à execução do AG configurado conforme os parâmetros de cada um dos 9 cenários propostos. Cada cenário foi testado repetindo 50 vezes a execução do AG. A Tabela 4 mostra os resultados dos testes de sintonia do AG, onde cada valor expresso indica a porcentagem do número de problemas-teste avaliados para os quais um cenário específico apresentou um melhor valor de FO que os demais.

**Tabela 4 – Resultados dos cenários de testes para sintonia do AG**

	Cenários de teste do AG		
	$pc = 80\%$	$pc = 90\%$	$pc = 100\%$
$pm = 5\%$	6,67%	6,67%	0%
$pm = 10\%$	6,67%	0%	0%
$pm = 15\%$	53,33%	0%	26,67%

Fonte: Elaborado pelo autor.

De acordo com os resultados obtidos no processo de sintonia dos parâmetros da metaheurística AG, pode-se concluir que, dentre os cenários propostos e testados, o que apresentou melhores resultados na minimização da FO empregada neste processo de sintonia foi o cenário cujos parâmetros possuem valor de 80% para a probabilidade de cruzamento dos indivíduos pais e 15% para a probabilidade de mutação dos indivíduos filhos, uma vez que nesta configuração obteve-se melhores resultados em 53,33% dos problemas-teste avaliados.

### 5.3.2 Sintonia dos parâmetros da Busca Tabu

A sintonia dos parâmetros da metaheurística BT, assim como a sintonia dos parâmetros do AG, foi obtida fixando os parâmetros que atuam diretamente sobre a quantidade de avaliações realizadas da FO, obtendo também um número de 20000 avaliações. Estes parâmetros foram fixados em:

- a) Tamanho da vizinhança = 20;
- b) Número máximo de iterações = 1000.

Geralmente, o critério de parada da BT é definido ajustando-se o  $BTmax$ , um parâmetro que define o número máximo de iterações do algoritmo sem observar melhora do valor da FO em relação à melhor solução. Para os testes de sintonia da BT, este critério de parada foi desabilitado, com o objetivo de forçar a execução pelo número de avaliações definido, equalizando o número de execuções entre os cenários de teste. Para a metaheurística BT, o parâmetro sintonizado foi o tamanho da LT ( $tamLT$ ), variando o valor desejado em 4 cenários diferentes, que seguem:

- a)  $tamLT = 10$  movimentos;
- b)  $tamLT = 20$  movimentos;
- c)  $tamLT = 30$  movimentos;
- d)  $tamLT = 40$  movimentos.

Os 15 problemas-teste foram submetidos à execução da BT configurada conforme cada um dos 4 cenários propostos. Cada cenário foi testado repetindo 50 vezes a execução da BT. A Tabela 5 mostra os resultados dos testes de sintonia da BT, onde cada valor expresso indica a porcentagem do número de problemas-teste avaliados para as quais um cenário específico apresentou um melhor valor de FO que os demais.

**Tabela 5 – Resultados dos cenários de testes para sintonia da BT**

Cenários de teste da BT			
<i>tamLT</i>			
10 soluções	20 soluções	30 soluções	40 soluções
20%	60%	6,67%	13,33%

Fonte: Elaborado pelo autor.

De acordo com os resultados obtidos no processo de sintonia dos parâmetros da metaheurística BT, pode-se concluir que, dentre os cenários propostos e testados, o que apresentou melhores resultados na minimização da FO foi o cenário que possui valor de 20 movimentos para o tamanho da LT, pois foi obtido melhor resultado em 60% dos problemas-teste nesta configuração.

### 5.3.3 Sintonia dos parâmetros do AG+BT

O algoritmo AG+BT proposto neste trabalho possui um acoplamento *high-level* com estratégia de controle colaborativa e execução intercalada (RAIDL, 2006), o que induz o algoritmo a trocar informações, executando uma metaheurística de cada vez. Esta troca de informações é realizada ao final da execução do AG, quando a melhor solução é repassada à BT, e também no fim da execução da BT, quando as melhores soluções são repassadas ao AG.

Na implementação do AG+BT, foram mantidos os valores dos parâmetros sintonizados anteriormente para o AG e para a BT. A quantidade de soluções repassadas da BT para o AG, configurada pela variável *tamLista*, assim como o número de ciclos de execução intercalada dos algoritmos, determinado por *numCiclos*, são parâmetros específicos do algoritmo híbrido que devem ser sintonizados.

Para a sintonia do algoritmo híbrido AG+BT, foram criados 9 cenários de teste, combinando 3 valores diferentes para a variável *numCiclos* e mais 3 valores distintos para a variável *tamLista*:

- numCiclos* = 2 ciclos e *tamLista* = 2 soluções;
- numCiclos* = 2 ciclos e *tamLista* = 5 soluções;
- numCiclos* = 2 ciclos e *tamLista* = 10 soluções;
- numCiclos* = 5 ciclos e *tamLista* = 2 soluções;
- numCiclos* = 5 ciclos e *tamLista* = 5 soluções;

- f) *numCiclos* = 5 ciclos e *tamLista* = 10 soluções;
- g) *numCiclos* = 10 ciclos e *tamLista* = 2 soluções;
- h) *numCiclos* = 10 ciclos e *tamLista* = 5 soluções;
- i) *numCiclos* = 10 ciclos e *tamLista* = 10 soluções.

Os 15 problemas-teste foram submetidos à execução do AG+BT configurado conforme cada um dos 9 cenários propostos. Cada cenário foi testado repetindo 50 vezes a execução do AG+BT. A Tabela 6 mostra os resultados dos testes de sintonia do AG+BT, onde cada valor expresso indica a porcentagem do número de problemas-teste avaliados para as quais um cenário específico apresentou um melhor valor de FO que os demais.

**Tabela 6 – Resultados dos cenários de testes para sintonia do AG+BT**

<i>numCiclos</i>	Cenários de teste do AG+BT		
	<i>tamLista</i>		
	2 Soluções	5 Soluções	10 Soluções
2 ciclos	6,67%	6,67%	13,33%
5 ciclos	13,33%	33,33%	6,67%
10 ciclos	6,67%	6,67%	6,67%

Fonte: Elaborado pelo autor.

De acordo com os resultados obtidos no processo de sintonia dos parâmetros do AG+BT, pode-se concluir que, dentre os cenários propostos e testados, o que apresentou melhores resultados na minimização da FO foi o cenário cujos parâmetros possuem valor 5 para a quantidade de ciclos de execução do algoritmo híbrido e valor 5 para a quantidade de soluções repassadas da BT para o AG, pois nesta configuração obteve-se melhores resultados em 33,33% dos problemas-teste.

#### 5.4 Validação dos algoritmos AG, BT e AG+BT

Os algoritmos propostos neste trabalho têm a finalidade de otimizar o número de movimentos e a instabilidade da embarcação. A parcela da FO que calcula a instabilidade foi formulada segundo os conceitos de estabilidade já conhecidos e relatados no trabalho de Imai *et al.* (2006). Já a parcela responsável pela quantificação do número de movimentos é baseada no modelo proposto por Azevedo *et al.* (2011).

A fim de que os algoritmos propostos AG, BT e AG+BT possam ser aplicados aos problemas-teste e fornecer resultados confiáveis, é necessário validá-los. Para tal, serão utilizados os mesmos problemas-teste já mencionados na seção 5.2, sendo submetidos aos algoritmos propostos voltados unicamente à minimização de movimentos, comparando os resultados obtidos com os resultados já conhecidos da literatura.

Todos os testes executados foram realizados submetendo os problemas-teste aos algoritmos configurados para trabalhar com 12 regras, conforme Azevedo *et al.* (2011),

buscando resultados próximos ou idênticos a este trabalho correlato. A Tabela 7 mostra os resultados obtidos após 50 execuções de cada algoritmo.

**Tabela 7 – Resultados dos testes de validação dos algoritmos com 12 regras**

I	N	M	NMin	AG *		AG		BT		AG+BT	
				FO	GAP (%)	FO	GAP (%)	FO	GAP (%)	FO	GAP (%)
1	10	1	6994	7072	1,12%	7516	7,46%	7516	7,46%	7516	7,46%
2	10	2	4172	4214	1,01%	4714	12,99%	4714	12,99%	4714	12,99%
3	10	3	17060	17116	0,33%	17186	0,74%	17186	0,74%	17186	0,74%
4	15	1	9974	10584	6,12%	10556	5,84%	10452	4,79%	10452	4,79%
5	15	2	4824	5030	4,27%	5968	23,71%	5968	23,71%	5968	23,71%
6	15	3	24908	25046	0,55%	25032	0,50%	25034	0,51%	25032	0,50%
7	20	1	10262	10802	5,26%	10826	5,50%	10816	5,40%	10816	5,40%
8	20	2	4982	5500	10,40%	6074	21,92%	6066	21,76%	6058	21,60%
9	20	3	32602	32638	0,11%	32630	0,09%	32630	0,09%	32630	0,09%
10	25	1	11014	11848	7,57%	11752	6,70%	11744	6,63%	11744	6,63%
11	25	2	5002	5466	9,28%	5558	11,12%	5542	10,80%	5520	10,36%
12	25	3	43722	44082	0,82%	43842	0,27%	43804	0,19%	43798	0,17%
13	30	1	11082	12580	13,52%	11862	7,04%	11842	6,86%	11834	6,79%
14	30	2	4720	5312	12,54%	5174	9,62%	5152	9,15%	5012	6,19%
15	30	3	53592	54398	1,50%	53746	0,29%	53712	0,22%	53594	0,00%
<b>Média</b>			<b>16327,3</b>	<b>16779,2</b>	<b>4,96%</b>	<b>16829,1</b>	<b>7,59%</b>	<b>16811,9</b>	<b>7,42%</b>	<b>16791,6</b>	<b>7,2%</b>

\* Resultados do AG proposto por Azevedo *et al.* (2011)

Fonte: Elaborado pelo autor.

A coluna **I** representa o número de identificação do problema-teste; A coluna **N** representa o número de portos; a coluna **M** identifica o tipo de matriz de transporte (explanado na seção 5.1); **NMin** indica o valor mínimo de movimentos, resultante do número de contêineres movimentados multiplicado por 2; a coluna **FO** mostra o menor número de movimentos resultante da execução do algoritmo; a coluna **GAP (%)** será a métrica utilizada para quantificar o desempenho do algoritmo, resultando do cálculo da diferença percentual do resultado da FO em relação ao valor mínimo de cada problema-teste; e a linha inferior da tabela apresenta os valores médios de cada coluna para todos os problemas-testes avaliados neste experimento.

A partir dos resultados, pode-se observar que o número de movimentos obtido na execução do AG, da BT e do AG+BT apresentou melhora em relação ao algoritmo proposto por Azevedo *et al.* (2011) em 8 dos 15 problemas-teste avaliados (53,33%) para os três algoritmos propostos. Nos casos em que não houve melhora, a diferença de desvio máxima em relação ao AG de Azevedo *et al.* (2011) foi de 19,44% no problema-teste 5, também para os três casos.

A média de desvios para o AG proposto apresentou um aumento absoluto de 2,63% em relação à média de desvios do AG apresentado no trabalho de Azevedo *et al.* (2011). No caso da BT, a mesma apresentou um aumento absoluto de 2,46% em relação à média de desvios do AG apresentado no trabalho de Azevedo *et al.* (2011). O AG+BT apresentou a menor média de desvios dentre os três algoritmos propostos, apresentando aumento absoluto

de 2,2% em relação à média de desvios apresentada pelo AG proposto em Azevedo *et al.* (2011).

A semelhança dos valores de minimização obtidos nos testes de validação do AG, da BT e do algoritmo híbrido AG+BT em comparação aos valores de otimização apresentados na pesquisa de Azevedo *et al.* (2011) demonstra a consistência das soluções propostas nesta abordagem. Assim, consideram-se validados os algoritmos propostos.

## 5.5 Experimentos com AG, BT e AG+BT

A etapa de sintonia dos algoritmos propostos possibilitou determinar, dentre todas as situações às quais os mesmos foram submetidos, a combinação de parâmetros que apresentou resultados mais próximos do ótimo. A partir de então, foi possível aplicar estes algoritmos aos problemas-teste com uma configuração eficaz, a fim de obter bons resultados e confrontar os valores obtidos, além de avaliar a eficiência do algoritmo AG+BT em relação às metaheurísticas puras.

A FO empregada nos algoritmos, para fins de comparação entre os algoritmos, assumiu apenas a parcela referente ao critério de minimização do número de movimentos e os resultados obtidos a partir dos testes foram analisados através da média, do desvio padrão e do erro relativo em relação à solução ótima (GAP), com base no conjunto de replicações dos algoritmos.

Foram registrados os tempos de execução dos três algoritmos, quando aplicados a cada problema-teste, a fim de analisar o esforço computacional exigido na busca pela solução. No entanto, os valores obtidos a partir dos testes não apresentaram diferenças quando do emprego de quantidades distintas de regras. Por isso, o tempo de execução não será considerado na análise destes resultados.

### 5.5.1 Minimização da movimentação com o AG

Para a etapa de experimentos com o AG proposto, foram mantidos os parâmetros definidos na sintonia, ajustando *pc* para 80% e *pm* para 15%. O parâmetro de tamanho da população foi fixado em 100 indivíduos e o critério de parada adotado foi de 50 gerações. Foram, então, propostos 3 cenários de teste, variando o número de regras utilizadas pelo algoritmo em 12, 24 e 36 regras. Os parâmetros referentes ao número de avaliações da FO foram configurados a fim de assegurar a diversidade do algoritmo, com uma quantidade relativa de cromossomos e gerações.

Na Tabela 8, são apresentados os valores de média e desvio padrão obtidos a partir das amostras de 50 execuções do AG para cada um dos 15 problemas-teste, sendo comparados os resultados do AG nas configurações de 12, 24 e 36 regras.

**Tabela 8 – Resultados médios dos testes com o AG para 12, 24 e 36 regras**

I	N	M	NMin	AG (12 REGRAS)		AG (24 REGRAS)		AG (36 REGRAS)	
				FO Média	Desvio Padrão ( $\sigma$ )	FO Média	Desvio Padrão ( $\sigma$ )	FO Média	Desvio Padrão ( $\sigma$ )
1	10	1	6994	7536,13	47,13	7195,93	61,50	7273,80	120,36
2	10	2	4172	4714,27	0,69	4569,73	12,74	4583,93	29,39
3	10	3	17060	17190,47	10,03	17073,53	5,27	17078,53	7,84
4	15	1	9974	10619,07	50,07	10280,00	88,60	10540,87	180,09
5	15	2	4824	6017,40	46,68	5769,93	62,69	5914,73	121,79
6	15	3	24908	25093,40	48,26	24923,47	14,35	24946,87	34,83
7	20	1	10262	10868,00	27,99	10715,40	62,40	10894,33	158,71
8	20	2	4982	6142,00	58,34	6004,33	54,05	6162,87	144,80
9	20	3	32602	32635,93	5,11	32602,13	0,73	32607,07	4,42
10	25	1	11014	11920,40	109,70	11672,20	120,63	12002,20	248,55
11	25	2	5002	5604,80	25,06	5460,93	50,65	5593,47	68,29
12	25	3	43722	43889,47	33,35	43749,20	15,21	43823,20	67,98
13	30	1	11082	11934,67	60,96	11887,73	49,02	12260,93	189,90
14	30	2	4720	5232,67	47,24	5140,27	75,18	5434,47	106,33
15	30	3	53592	53882,60	96,16	53655,13	36,44	53808,20	113,61
<b>Média</b>			<b>16327,3</b>	<b>16885,4</b>	<b>44,5</b>	<b>16713,3</b>	<b>47,3</b>	<b>16861,7</b>	<b>106,5</b>

Fonte: Elaborado pelo autor.

Considerado os resultados apresentados na Tabela 8, verifica-se que o desvio padrão obtido, para o caso do AG de 12 regras, varia de 0,69 movimentos (no problema-teste 2) a 109,70 movimentos (no problema-teste 10), com média de 44,5 movimentos para os 15 problemas-teste. Para o AG de 24 regras, a média entre os desvios padrão foi de 47,30 movimentos, variando de 0,73 (no problema-teste 9) a 120,63 (no problema-teste 10). No caso do AG de 36 regras, a variação dos desvios foi de 4,42 movimentos (no problema-teste 9) a 248,55 movimentos (no problema-teste 10), com média de 106,5 entre os valores de desvio padrão.

Observa-se também, para o AG de 24 regras, uma redução da FO Média em relação ao AG de 12 regras (de 16885,4 para 16713,3), o que demonstra que o uso das regras 13 a 24 tornou o AG mais eficiente, alcançando valores mais próximos do ótimo. Já na comparação entre o AG de 24 e o de 36 regras, observa-se um aumento desta média (de 16713,3 para 16861,7), o que denota que o uso das regras 25 a 36 tornou a busca menos eficiente.

Ao passo que se aumenta o número de regras utilizadas, há um aumento na média dos desvios e na FO Média, o que pode ser visto como reflexo do aumento do espaço de soluções associado à incapacidade do AG em intensificar, quer seja pela redução de diversidade, ou pelo fato de as regras 25 a 36 serem baseadas em uma regra de descarga que desfavorece a minimização de movimentos (Rd3).

Ainda nos experimentos com o AG, os valores mínimos obtidos para cada problema-teste e o desvio relativo deste resultado em relação à solução ótima são apresentados na Tabela 9, em comparação com o AG proposto por Azevedo *et al.* (2011). Os valores médios da FO e do GAP de cada algoritmo são apresentados na linha inferior da tabela.

Tabela 9 – Resultados mínimos dos testes com o AG para 12, 24 e 36 regras

I	N	M	NMin	AG *		AG (12 REGRAS)		AG (24 REGRAS)		AG (36 REGRAS)	
				FO	GAP (%)						
1	10	1	6994	7072	1,12%	7516	7,46%	7148	2,20%	7148	2,20%
2	10	2	4172	4214	1,01%	4714	12,99%	4556	9,20%	4556	9,20%
3	10	3	17060	17116	0,33%	17186	0,74%	17064	0,02%	17066	0,04%
4	15	1	9974	10584	6,12%	10494	5,21%	10160	1,86%	10298	3,25%
5	15	2	4824	5030	4,27%	5970	23,76%	5652	17,16%	5704	18,24%
6	15	3	24908	25046	0,55%	25034	0,51%	24918	0,04%	24922	0,06%
7	20	1	10262	10802	5,26%	10830	5,53%	10632	3,61%	10716	4,42%
8	20	2	4982	5500	10,40%	6076	21,96%	5852	17,46%	6006	20,55%
9	20	3	32602	32638	0,11%	32630	0,09%	32602	0,00%	32606	0,01%
10	25	1	11014	11848	7,57%	11804	7,17%	11478	4,21%	11528	4,67%
11	25	2	5002	5466	9,28%	5542	10,80%	5382	7,60%	5434	8,64%
12	25	3	43722	44082	0,82%	43834	0,26%	43728	0,01%	43738	0,04%
13	30	1	11082	12580	13,52%	11870	7,11%	11806	6,53%	11876	7,16%
14	30	2	4720	5312	12,54%	5168	9,49%	5038	6,74%	5216	10,51%
15	30	3	53592	54398	1,50%	53758	0,31%	53612	0,04%	53652	0,11%
<b>Média</b>			<b>16327,3</b>	<b>16779,2</b>	<b>4,96%</b>	<b>16828,4</b>	<b>7,56%</b>	<b>16641,9</b>	<b>5,11%</b>	<b>16697,7</b>	<b>5,94%</b>

\* Resultados do AG proposto por Azevedo et al. (2011)

Fonte: Elaborado pelo autor.

A partir dos resultados mínimos obtidos, pode-se observar que o AG, ao ser configurado com 12 regras, apresentou melhora na minimização da FO em relação ao algoritmo proposto em Azevedo *et al.* (2011), pois gerou resultados melhores em 8 dos 15 casos (53,33%). Para os casos em que não houve melhora, a diferença absoluta máxima entre o GAP do AG de 12 regras em relação ao AG de Azevedo *et al.* (2011) foi de 19,49% no problema-teste 5. A média dos GAP's de todos os problemas-teste para o AG de 12 regras proposto foi de 7,56%, apresentando uma diferença absoluta de +2,6% em relação à média de GAP's do algoritmo de Azevedo *et al.* (2011), que foi de 4,96%.

Já para a configuração de 24 regras do AG, os resultados foram ainda mais efetivos, apresentando melhora em 11 dos 15 casos (73,33%) em relação ao algoritmo proposto na literatura. Para os casos em que não houve melhora, a diferença absoluta máxima entre o GAP do AG de 24 regras em relação ao AG de Azevedo *et al.* (2011) foi de 12,89% no problema-teste 5. A média dos GAP's de todos os problemas-teste para o AG de 24 regras proposto foi de 5,11%, apresentando uma diferença absoluta de +0,15% em relação à média de GAP's do algoritmo de Azevedo *et al.* (2011).

Comparado ao algoritmo de Azevedo *et al.* (2011), o AG de 36 regras apresentou melhora em 11 dos 15 casos e nos mesmos problemas-teste em que a versão de 24 regras apresentou melhora (73,33%). No entanto, os resultados apresentados foram menos eficientes, pois possuem um GAP maior que o algoritmo de 24 regras. Para os casos em que não houve melhora, a diferença absoluta máxima entre o GAP do AG de 36 regras em relação ao GAP do mesmo problema-teste no AG de Azevedo *et al.* (2011) foi de 13,97%, também no problema-teste 5. A média dos GAP's de todos os problemas-teste para o AG 36 regras

proposto foi de 5,94%, apresentando uma diferença absoluta de +0,98% em relação à média de GAP's do algoritmo de Azevedo *et al.* (2011).

### 5.5.2 Minimização da movimentação com a BT

Para a etapa de experimentos com a BT proposta, foram mantidos os parâmetros anteriormente definidos na sintonia, ajustando o tamanho da LT em 20 movimentos. Foram, então, propostos 3 cenários de teste, variando o número de regras utilizadas pelo algoritmo em 12, 24 e 36 regras. Os parâmetros referentes ao número de avaliações da FO foram configurados a fim de garantir a intensificação do algoritmo sem impactar no tempo de execução do mesmo, mantendo um equilíbrio entre o tamanho da vizinhança e o critério de parada *BTmax*. Tanto o valor do tamanho da vizinhança quanto o *BTmax* foram fixados em 20.

Na Tabela 10, são apresentados os valores de média e desvio padrão obtidos a partir das amostras de 50 execuções da BT para cada um dos 15 problemas-teste, sendo comparados os resultados da BT nas configurações de 12, 24 e 36 regras.

**Tabela 10 – Resultados médios dos testes com a BT para 12, 24 e 36 regras**

I	N	M	NMin	BT (12 REGRAS)		BT (24 REGRAS)		BT (36 REGRAS)	
				FO Média	Desvio Padrão ( $\sigma$ )	FO Média	Desvio Padrão ( $\sigma$ )	FO Média	Desvio Padrão ( $\sigma$ )
1	10	1	6994	7589,80	65,89	7279,47	129,85	7252,20	104,26
2	10	2	4172	4714,07	0,37	4601,87	67,28	4592,27	61,72
3	10	3	17060	17217,13	43,97	17072,87	4,51	17073,60	7,73
4	15	1	9974	10637,47	213,75	10268,33	87,19	10314,47	173,64
5	15	2	4824	6121,73	181,69	5788,13	128,82	5777,00	82,39
6	15	3	24908	25077,80	25,05	24919,60	30,25	24924,67	24,80
7	20	1	10262	10891,67	111,59	10766,13	144,30	10730,73	115,05
8	20	2	4982	6212,80	136,29	5988,60	147,82	5985,60	112,59
9	20	3	32602	32647,40	69,85	32602,60	1,40	33016,00	666,23
10	25	1	11014	11976,20	221,02	11630,33	148,30	11678,27	188,83
11	25	2	5002	5585,00	34,72	5427,53	45,95	5591,60	423,41
12	25	3	43722	43894,53	84,86	43728,73	15,36	43731,53	7,20
13	30	1	11082	12569,93	876,00	11813,47	97,74	11919,00	131,48
14	30	2	4720	5479,20	425,79	5104,67	86,49	5142,60	97,35
15	30	3	53592	53767,93	57,10	53610,80	26,51	53624,20	60,14
<b>Média</b>			<b>16327,3</b>	<b>16958,8</b>	<b>169,9</b>	<b>16706,9</b>	<b>77,5</b>	<b>16756,9</b>	<b>150,5</b>

Fonte: Elaborado pelo autor.

Comparando os dados expostos na Tabela 10, é possível visualizar que o desvio padrão obtido das 50 amostras, para o caso da BT de 12 regras, varia de 0,37 movimentos (no problema-teste 2) a 876 movimentos (no problema-teste 13), com uma média de 169,9 movimentos entre os desvios padrão para os 15 problemas-teste. Para a BT de 24 regras, a média entre os desvios padrão foi de 77,5 movimentos, variando de 1,4 (no problema-teste 9)

a 148,3 (no problema-teste 10). No caso da BT de 36 regras, a variação dos desvios foi de 7,2 (no problema-teste 12) a 666,23 movimentos (no problema-teste 9), com uma média entre os desvios padrão de 150,5 movimentos.

A partir dos resultados dos experimentos com a BT, efetuando a média entre os valores de FO Média para os 15 problemas-teste analisados, observa-se, para a BT de 24 regras, uma redução deste valor médio em relação à BT de 12 regras (de 16958,8 para 16706,9), o que mostra que o uso das regras 13 a 24 tornou a BT mais eficiente. Já na comparação entre a BT de 24 e a BT de 36 regras, observa-se um aumento desta média (de 16706,9 para 16756,9), o que denota que o uso das regras 25 a 36 tornou a busca menos eficiente.

De maneira similar ao ocorrido nos experimentos com o AG, para a BT de 36 regras, o aumento observado nos desvios padrão e na FO Média pode também ser explicado pelo aumento do espaço de soluções associado à característica de intensificação da BT e pelo fato de as regras 25 a 36 serem baseadas em uma regra de descarga que desfavorece a minimização de movimentos (Rd3).

Ainda nos experimentos com a BT, os valores mínimos obtidos para cada problema-teste e o desvio relativo deste resultado em relação à solução ótima são apresentados na Tabela 11, em comparação com o AG proposto por Azevedo *et al.* (2011). Os valores médios da FO e do GAP de cada algoritmo são apresentados na linha inferior da tabela.

**Tabela 11 – Resultados mínimos dos testes com a BT para 12, 24 e 36 regras**

I	N	M	NMin	AG *		BT (12 REGRAS)		BT (24 REGRAS)		BT (36 REGRAS)	
				FO	GAP (%)						
1	10	1	6994	7072	1,12%	7516	7,46%	7148	2,20%	7148	2,20%
2	10	2	4172	4214	1,01%	4714	12,99%	4554	9,16%	4554	9,16%
3	10	3	17060	17116	0,33%	17186	0,74%	17064	0,02%	17064	0,02%
4	15	1	9974	10584	6,12%	10464	4,91%	10150	1,76%	10162	1,88%
5	15	2	4824	5030	4,27%	5968	23,71%	5638	16,87%	5638	16,87%
6	15	3	24908	25046	0,55%	25032	0,50%	24918	0,04%	24910	0,01%
7	20	1	10262	10802	5,26%	10816	5,40%	10618	3,47%	10616	3,45%
8	20	2	4982	5500	10,40%	6072	21,88%	5844	17,30%	5860	17,62%
9	20	3	32602	32638	0,11%	32630	0,09%	32604	0,01%	32602	0,00%
10	25	1	11014	11848	7,57%	11744	6,63%	11340	2,96%	11354	3,09%
11	25	2	5002	5466	9,28%	5546	10,88%	5366	7,28%	5372	7,40%
12	25	3	43722	44082	0,82%	43802	0,18%	43732	0,02%	43724	0,00%
13	30	1	11082	12580	13,52%	11834	6,79%	11670	5,31%	11682	5,41%
14	30	2	4720	5312	12,54%	5142	8,94%	5030	6,57%	5006	6,06%
15	30	3	53592	54398	1,50%	53712	0,22%	53594	0,00%	53594	0,00%
<b>Média</b>			<b>16327,3</b>	<b>16779,2</b>	<b>4,96%</b>	<b>16811,9</b>	<b>7,42%</b>	<b>16618,0</b>	<b>4,86%</b>	<b>16619,1</b>	<b>4,88%</b>

\* Resultados do AG proposto por Azevedo et al. (2011)

Fonte: Elaborado pelo autor.

Os resultados obtidos com a execução da BT na configuração de 12 regras apontam melhora na minimização da FO em relação ao algoritmo proposto em Azevedo *et al.* (2011),

gerando resultados melhores em 8 dos 15 casos (53,33%). Para os casos em que não houve melhora, a diferença absoluta máxima entre o GAP da BT de 12 regras e o GAP do AG de Azevedo *et al.* (2011) foi de 19,44% no problema-teste 5. A média dos GAP's de todos os problemas-teste para a BT de 12 regras proposta foi de 7,42%, apresentando uma diferença absoluta de +2,46% em relação à média dos GAP's do algoritmo de Azevedo *et al.* (2011), que foi de 4,96%.

Já no que diz respeito à configuração de 24 regras da BT, os resultados apresentados superaram os valores obtidos com a BT de 12 regras, obtendo melhora em 11 dos 15 casos avaliados (73,33%) em relação ao AG de 12 regras proposto na literatura por Azevedo *et al.* (2011). Para os casos em que não houve melhora, a diferença absoluta máxima do GAP da BT de 24 regras em relação ao AG de Azevedo *et al.* (2011) foi de 12,6%, ocorrida no problema-teste 5. A média dos GAP's de todos os problemas-teste para a BT de 24 regras proposta foi de 4,86%, melhorando a marca de 7,42% do AG de 12 regras e apresentando uma diferença de -0,1% em relação à média de GAP's do algoritmo de Azevedo *et al.* (2011), ou seja, apresentando resultados mais eficientes que a literatura correlata.

Comparada ao algoritmo de Azevedo *et al.* (2011), a configuração de 36 regras da BT apresentou resultados mais eficientes em 11 dos 15 casos e nos mesmos problemas-teste em que a versão de 24 regras apresentou melhora (73,33%). Além disso, a configuração de 36 regras da BT apresentou resultados de minimização com média de desvios de 4,88%, melhorando os resultados encontrados na literatura correlata (4,96%), em Azevedo *et al.* (2011).

### 5.5.3 Minimização da movimentação com o AG+BT

Para a etapa de experimentos com o AG+BT, foram mantidos os parâmetros definidos na sintonia das metaheurísticas puras e do próprio AG+BT. Dessa forma, os parâmetros do AG+BT foram configurados em 5 ciclos para a variável de iteração do algoritmo híbrido e em 5 soluções para o tamanho da lista de soluções da BT repassada ao AG (*tamLista*) como composição da população.

Em relação às metaheurísticas que compõem o algoritmo híbrido, para o AG, os parâmetros foram mantidos em 80% para a probabilidade de cruzamento dos indivíduos pais e 15% para a probabilidade de mutação dos indivíduos filhos, conforme procedimento de sintonia. Para a BT, o parâmetro *tamLT* foi definido em 20 movimentos, também de acordo com procedimento de sintonia.

Assim como os experimentos com as metaheurísticas puras, foram definidos os mesmos 3 cenários de experimento, variando a configuração de quantidade de regras utilizadas em 12, 24 e 36 regras. Foram realizados testes através de 50 execuções do algoritmo AG+BT. Na Tabela 12, são demonstrados os resultados médios obtidos nos experimentos realizados e o valor do desvio padrão das 50 execuções, para o AG+BT de 12, 24 e 36 regras.

**Tabela 12 – Resultados médios dos testes com o AG+BT para 12, 24 e 36 regras**

I	N	M	NMin	AG+BT (12 REGRAS)		AG+BT (24 REGRAS)		AG+BT (36 REGRAS)	
				FO Média	Desvio Padrão ( $\sigma$ )	FO Média	Desvio Padrão ( $\sigma$ )	FO Média	Desvio Padrão ( $\sigma$ )
1	10	1	6994	7550,40	32,26	7182,60	46,27	7240,32	87,32
2	10	2	4172	4717,13	3,63	4560,07	4,68	4576,44	31,44
3	10	3	17060	17195,93	8,06	17071,93	6,00	17075,80	9,28
4	15	1	9974	10570,80	55,98	10208,47	51,73	10274,93	93,41
5	15	2	4824	5989,93	28,23	5715,53	50,75	5722,87	57,87
6	15	3	24908	25057,27	22,19	24910,80	4,74	24927,00	27,16
7	20	1	10262	10836,93	36,69	10633,60	30,21	10657,13	47,64
8	20	2	4982	6127,20	66,43	5911,00	45,94	5933,67	45,65
9	20	3	32602	32630,27	0,87	32602,00	0,00	32602,00	0,00
10	25	1	11014	11809,13	99,34	11468,33	132,32	11496,00	132,15
11	25	2	5002	5556,20	9,18	5388,13	23,86	5493,27	46,42
12	25	3	43722	43813,67	8,66	43722,67	1,69	43725,73	4,60
13	30	1	11082	11858,07	15,21	11778,33	61,60	11805,80	101,46
14	30	2	4720	5031,40	45,25	5036,87	35,95	5058,53	45,75
15	30	3	53592	53597,80	3,46	53596,67	3,08	53602,13	9,98
<b>Média</b>			<b>16327,3</b>	<b>16822,8</b>	<b>29,0</b>	<b>16652,5</b>	<b>33,3</b>	<b>16679,4</b>	<b>49,3</b>

Fonte: Elaborado pelo autor.

Comparando os valores dos resultados evidenciados na Tabela 12, nota-se que o desvio padrão obtido das 50 amostras, para o caso do AG+BT de 12 regras, varia de 0,87 (no problema-teste 9) a 99,34 (no problema-teste 10), com uma média de 29,0 movimentos entre os desvios padrão para os 15 problemas-teste. Para o AG+BT 24 regras, a média entre os desvios padrão foi de 33,3 ao variar de 0 (no problema-teste 9) a 132,32 (no problema-teste 10). No caso do AG+BT 36 regras, a variação dos desvios foi de 0 (no problema-teste 9) a 132,15 (no problema-teste 10), com uma média de 49,3 movimentos entre os desvios.

A partir dos resultados dos experimentos com o AG+BT, efetuando a média entre os valores de FO Média dos 15 problemas-teste analisados, observa-se, para o AG+BT de 24 regras, uma redução deste valor médio em relação ao AG+BT de 12 regras (de 16822,8 para 16652,5), o que mostra que o uso das regras 13 a 24 tornou o AG+BT mais eficiente. Já na comparação entre o AG+BT de 24 e o AG+BT de 36 regras, observa-se um aumento desta média (de 16652,5 para 16679,4), o que denota que o uso das regras 25 a 36 tornou a busca menos eficiente.

Assim como o ocorrido nos experimentos com o AG e com a BT de 36 regras, o aumento observado nos desvios padrão e na FO Média pode ser explicado pelo aumento do espaço de soluções ou pelo fato de as regras 25 a 36 serem baseadas em uma regra de descarga que desfavorece a minimização de movimentos (Rd3).

Ainda nos experimentos com o AG+BT, os valores mínimos obtidos para cada problema-teste e o desvio relativo deste resultado em relação à solução ótima são apresentados na Tabela 13, em comparação com o AG de Azevedo *et al.* (2011). Os valores médios da FO e do GAP de cada algoritmo são apresentados na linha inferior da tabela.

Tabela 13 – Resultados mínimos dos testes com a o AG+BT para 12, 24 e 36 regras

I	N	M	NMin	AG *		AG+BT (12 REGRAS)		AG+BT (24 REGRAS)		AG+BT (36 REGRAS)	
				FO	GAP (%)	FO	GAP (%)	FO	GAP (%)	FO	GAP (%)
1	10	1	6994	7072	1,12%	7516	7,46%	7148	2,20%	7148	2,20%
2	10	2	4172	4214	1,01%	4714	12,99%	4554	9,16%	4554	9,16%
3	10	3	17060	17116	0,33%	17186	0,74%	17064	0,02%	17064	0,02%
4	15	1	9974	10584	6,12%	10452	4,79%	10124	1,50%	10124	1,50%
5	15	2	4824	5030	4,27%	5968	23,71%	5638	16,87%	5638	16,87%
6	15	3	24908	25046	0,55%	25032	0,50%	24908	0,00%	24908	0,00%
7	20	1	10262	10802	5,26%	10816	5,40%	10612	3,41%	10612	3,41%
8	20	2	4982	5500	10,40%	6058	21,60%	5848	17,38%	5838	17,18%
9	20	3	32602	32638	0,11%	32630	0,09%	32602	0,00%	32602	0,00%
10	25	1	11014	11848	7,57%	11744	6,63%	11330	2,87%	11324	2,81%
11	25	2	5002	5466	9,28%	5542	10,80%	5366	7,28%	5368	7,32%
12	25	3	43722	44082	0,82%	43798	0,17%	43722	0,00%	43722	0,00%
13	30	1	11082	12580	13,52%	11834	6,79%	11680	5,40%	11680	5,40%
14	30	2	4720	5312	12,54%	5012	6,19%	4996	5,85%	4990	5,72%
15	30	3	53592	54398	1,50%	53594	0,00%	53594	0,00%	53594	0,00%
<b>Média</b>			<b>16327,3</b>	<b>16779,2</b>	<b>4,96%</b>	<b>16793,1</b>	<b>7,19%</b>	<b>16612,4</b>	<b>4,80%</b>	<b>16611,1</b>	<b>4,77%</b>

\* Resultados do AG proposto por Azevedo et al. (2011)

Fonte: Elaborado pelo autor.

Os resultados obtidos com a execução do AG+BT de 12 regras apontam melhora na minimização da FO em relação ao algoritmo proposto em Azevedo *et al.* (2011), gerando resultados menores em 8 dos 15 casos (53,33%). Para os casos em que não houve melhora, a diferença absoluta máxima entre o GAP do AG+BT 12 regras e o GAP do AG de Azevedo *et al.* (2011) foi de 19,44% no problema-teste 5. A média dos GAP's de todos os problemas-teste para o AG+BT 12 regras proposto foi de 7,19%, apresentando uma diferença absoluta de +2,23% em relação à média de GAP's do algoritmo de Azevedo *et al.* (2011), que foi de 4,96%.

Já para a configuração de 24 regras do AG+BT, os resultados mostraram eficiência maior que a configuração 12 regras, apresentando melhora em 11 dos 15 casos (73,33%) em relação ao algoritmo proposto na literatura. Para os casos em que não houve melhora, a diferença absoluta máxima entre o GAP do AG+BT 24 regras e do AG de Azevedo *et al.* (2011) foi de 12,6% no problema-teste 5. A média dos GAP's de todos os problemas-teste para o AG+BT 24 regras proposto foi de 4,80%, com uma diferença absoluta de -0,16% em relação à média de GAP's do algoritmo de Azevedo *et al.* (2011), ou seja, a média de GAP's para o AG+BT de 24 regras foi menor que a média de GAP's do algoritmo apresentado na literatura.

Comparado ao algoritmo de Azevedo *et al.* (2011), o AG+BT de 36 regras apresentou melhora nos mesmos 11 casos em que a versão de 24 regras apresentou melhora (73,33% dos casos). A configuração de 36 regras do AG+BT apresentou resultados de minimização ainda melhores que a configuração de 24 regras, reduzindo a média dos GAP's para 4,77%,

superando os resultados apresentados na literatura correlata, que é de 4,96% (AZEVEDO *et al.*, 2011).

## 5.6 Minimização da movimentação e da estabilidade

Dando seguimento aos experimentos e resultados, esta seção foca-se no último objetivo específico desta pesquisa, que é o de propor um modelo de minimização da instabilidade em conjunto com a minimização do número de remanejamentos de contêineres, elaborando, desta forma, uma ferramenta computacional para fazer a gestão de ambos os aspectos. A relevância de gerir os dois objetivos reside no fato de os mesmos serem conflitantes, isto é, uma otimização focada em minimizar apenas um dos objetivos ocasiona no aumento do outro.

Os dados e registros evidenciados a partir dos testes realizados permitem chegar à conclusão de que o algoritmo AG+BT com 36 regras apresenta melhor resultado se comparado às metaheurísticas puras propostas neste trabalho, justificando sua utilização nesta etapa de experimentos. A configuração dos objetivos da minimização a ser realizada pelo algoritmo é implementada através do balanceamento dos pesos de cada parcela da FO, a saber:  $\beta$ ,  $\alpha$ ,  $\chi$  e  $\delta$ , que representam, respectivamente, os pesos de Movimentação, GM, *List* e *Trim*. Todas as parcelas que compõem a FO foram normalizadas, ou seja, possuem uma configuração que varia de 0 a 1, contribuindo em proporções iguais para o cálculo da FO.

Para os testes de otimização, foram propostos 3 cenários: (i) minimização do número de movimentos de contêineres, (ii) minimização de instabilidade da embarcação e (iii) minimização com múltiplos objetivos, ou seja, minimizando instabilidade e movimentação simultaneamente.

Com a finalidade de configurar o AG+BT para a minimização do número de movimentos, foi necessário configurar os pesos relativos à estabilidade ( $\alpha$ ,  $\chi$  e  $\delta$ ) para o valor 0 e ajustar para o valor 1 o peso relativo à movimentação ( $\beta$ ). Para a minimização da instabilidade, os pesos relativos à estabilidade ( $\alpha$ ,  $\chi$  e  $\delta$ ) foram ajustados para o valor 1 e o parâmetro relativo à movimentação ( $\beta$ ) foi configurado para o valor 0. A configuração para minimização com múltiplos objetivos foi realizada igualando a contribuição dos dois objetivos conflitantes em 50%, ou seja,  $\alpha$ ,  $\chi$  e  $\delta$  assumem valor 1 e  $\beta$  assume valor igual à soma de  $\alpha$ ,  $\chi$  e  $\delta$  ( $\alpha + \chi + \delta = 1 + 1 + 1 = 3$ ).

Os testes de otimização descritos a seguir foram realizados com o algoritmo AG+BT, que, perante o confronto com os resultados do AG e BT, apresentou resultado mais eficiente. A configuração de regras utilizada para os testes de otimização foi a de 36 regras, que, para o AG+BT, possibilitou a obtenção de melhores resultados em relação às metaheurísticas puras. A Tabela 14 mostra a comparação de dois cenários, a saber: minimização de movimentos e minimização de instabilidade. Os valores que compõem a tabela são resultados decorrentes de 50 execuções do AG+BT configurado para 36 regras, aplicado aos 15 problemas-teste previamente definidos.

Tabela 14 – Comparação AG+BT com minimização de movimentos *versus* instabilidade

I	N	M	NMin	AG+BT 36 regras							
				$\beta = 1, \alpha = 0, \chi = 0 \text{ e } \delta = 0$				$\beta = 0, \alpha = 1, \chi = 1 \text{ e } \delta = 1$			
				MOV	GM	LIST	TRIM	MOV	GM	LIST	TRIM
1	10	1	6994	7148	22,3579	5,679	5,2225	10672	20,1678	2,1476	2,7756
2	10	2	4172	4554	24,907	6,5411	4,3503	9574	19,7077	2,025	2,3917
3	10	3	17060	17064	25,4126	9,961	4,6541	18842	21,083	1,7706	2,774
4	15	1	9974	10126	38,8538	4,036	9,2029	17916	36,5627	1,4595	4,3612
5	15	2	4824	5638	38,1886	12,4921	11,7796	13560	31,5866	2,559	3,7637
6	15	3	24908	24908	33,2373	9,1204	6,8861	26310	32,2583	2,4114	3,5127
7	20	1	10262	10612	53,7032	13,8388	9,6747	18348	50,2898	1,6136	6,3438
8	20	2	4982	5840	53,1223	7,3146	13,7775	13538	47,8552	3,4476	5,6362
9	20	3	32602	32602	44,0301	27,9609	14,6677	33998	39,1302	7,7405	7,6058
10	25	1	11014	11326	67,1912	9,9764	13,1736	28424	64,7962	1,174	8,6685
11	25	2	5002	5366	66,7866	16,4307	16,2046	22518	57,6888	5,7863	9,3333
12	25	3	43722	43722	59,0942	6,5605	11,1231	46102	57,2723	3,4016	6,7409
13	30	1	11082	11714	85,3807	15,2214	16,9993	26466	78,7843	2,5799	10,1545
14	30	2	4720	5010	84,3606	17,6912	19,2138	14854	77,1775	2,4327	9,8042
15	30	3	53592	53594	69,5873	15,7214	18,0114	57922	67,4766	5,4957	7,8303
Média			16327,3	16614,9	51,1	11,9	11,7	23936,3	46,8	3,1	6,1

Fonte: Elaborado pelo autor.

Comparando os resultados dos experimentos de minimização do número de movimentos em relação à minimização da instabilidade, foi possível verificar que o foco somente na minimização da movimentação, de fato, reduz o número de movimentos, no entanto acarreta na obtenção de valores mais altos de GM, *List* e *Trim*, se comparados à minimização de instabilidade, e isso ocorre para todos os problemas-teste avaliados. Na prática, ocorre a minimização do custo da operação de carga e descarga, decorrente da minimização da quantidade de movimentos, porém a solução compromete tanto a estabilidade quanto a segurança da embarcação, agravada pelos esforços estruturais aos quais o casco é submetido.

Da mesma forma, o foco somente na minimização de instabilidade influi na redução dos parâmetros GM, *List* e *Trim* e no aumento do número de movimentos, também para todos os problemas-teste avaliados. Em outras palavras, pelo fato de a estabilidade ser o único foco da otimização, o número de movimentos se torna elevado, aumentando, assim, o custo das operações no terminal portuário e reduzindo a lucratividade do mesmo.

Os resultados apresentados pelos dois cenários demonstram o conflito entre a minimização do número de movimentos e da instabilidade da embarcação. Dado este contexto, evidencia-se a importância de fazer a gestão de ambos os objetivos, configurando o algoritmo para atender tanto à otimização da movimentação quanto da estabilidade. A Tabela 15 mostra os resultados obtidos para o terceiro cenário, que consiste no AG+BT configurado para otimização de ambos objetivos. Os valores que compõem a tabela são resultados decorrentes de 50 execuções do AG+BT configurado para 36 regras, aplicado aos 15 problemas-teste previamente definidos.

Tabela 15 – AG+BT com otimização de múltiplos objetivos

I	N	M	NMin	AG+BT 36 regras				
				$\beta = 3, \alpha = 1, \chi = 1$ e $\delta = 1$				
				MOV	GM	LIST	TRIM	TEMPO (S)
1	10	1	6994	7156	21,5675	3,2228	3,0775	30,0439
2	10	2	4172	4976	21,072	5,2001	2,5883	30,7014
3	10	3	17060	17084	21,1313	3,5726	3,6171	32,4024
4	15	1	9974	10632	37,2263	4,9024	5,0332	40,1538
5	15	2	4824	6276	32,8112	4,567	4,978	38,1977
6	15	3	24908	24922	32,1819	5,9576	4,3802	39,1933
7	20	1	10262	10790	51,8046	6,5379	7,5618	48,1478
8	20	2	4982	6808	48,3166	4,3492	6,5034	45,8934
9	20	3	32602	32606	39,8325	8,6226	7,6362	45,0973
10	25	1	11014	11594	65,5665	3,6088	9,8145	55,9953
11	25	2	5002	7062	58,2556	8,3589	10,8672	53,7343
12	25	3	43722	43732	56,881	4,5947	8,3071	56,8538
13	30	1	11082	12046	81,2133	6,5516	12,1177	66,1334
14	30	2	4720	5770	78,167	5,7383	9,7447	61,7795
15	30	3	53592	53602	67,0173	8,1684	10,4105	61,8348
<b>Média</b>			<b>16327,3</b>	<b>17003,7</b>	<b>47,5</b>	<b>5,6</b>	<b>7,1</b>	<b>47,1</b>

Fonte: Elaborado pelo autor.

Os resultados apresentados pela Tabela 15 indicam que, neste último cenário, a configuração de múltiplos objetivos da otimização possibilitou a obtenção de valores intermediários tanto para a movimentação quanto para a estabilidade, isto é, como o foco da minimização não ficou restrito a apenas um objetivo, foi possível minimizar tanto a movimentação quanto a instabilidade.

A facilidade de balanceamento dos pesos de cada parcela da FO permite configurar o algoritmo de maneira que atenda à necessidade de minimização, sendo esta uma ferramenta de gestão da minimização de movimentação e de instabilidade para navios porta-contêineres.

## 5.7 Abordagem de alto desempenho

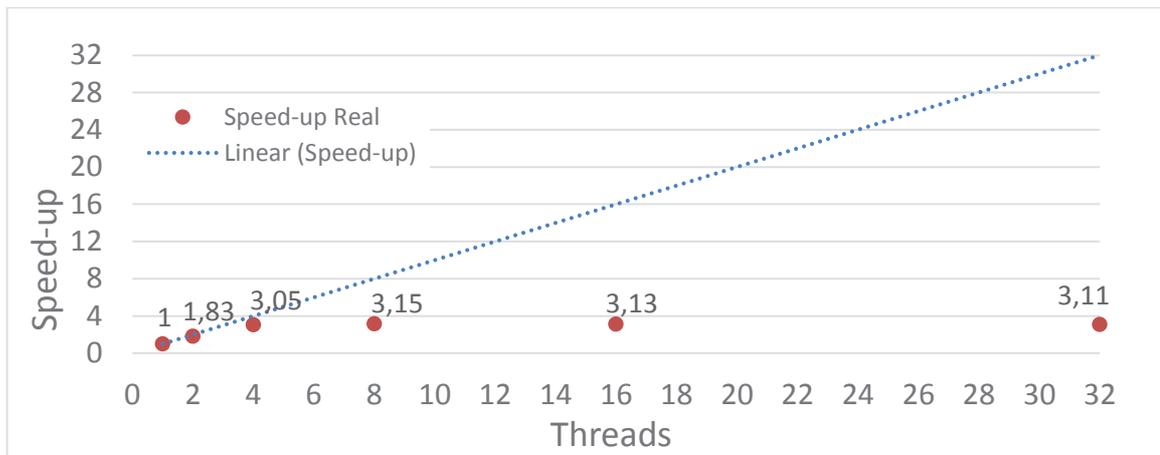
De acordo com o que foi abordado na seção 2.6, há uma preocupação por parte dos fabricantes com o aumento de desempenho dos computadores e das aplicações. No referido contexto, este estudo propõe uma abordagem paralela de alto desempenho para o algoritmo AG+BT, com o objetivo de comprovar a obtenção de uma maior eficiência de tempo em relação a uma versão sequencial do algoritmo. Foram comparados os desempenhos da aplicação híbrida executada sequencialmente e da implementação paralela da mesma.

A abordagem de paralelização empregada no algoritmo híbrido obedece ao modelo mestre-escravo e foi implementada na etapa de avaliação da função objetivo. Esta abordagem paralela tem como objetivo obter um desempenho, relativo ao tempo de processamento,

superior à versão sequencial através do *Multithreading*, utilizando os recursos de arquitetura Multicore dos processadores (ALBA, 2005). As aplicações comparadas neste estudo foram desenvolvidas em linguagem C# sobre a plataforma .NET, utilizando a biblioteca *Threading*.

Os experimentos foram realizados considerando 1, 2, 4, 8, 16 e 32 *threads*, com amostras de 30 execuções para cada um dos 15 problemas-teste avaliados, tendo sido calculada a média de desempenho entre elas. As métricas de desempenho utilizadas na comparação entre os algoritmos sequencial e paralelo foram o *speed-up* e a eficiência (ALBA, 2005). A análise de *speed-up* pode ser observada no Gráfico 1.

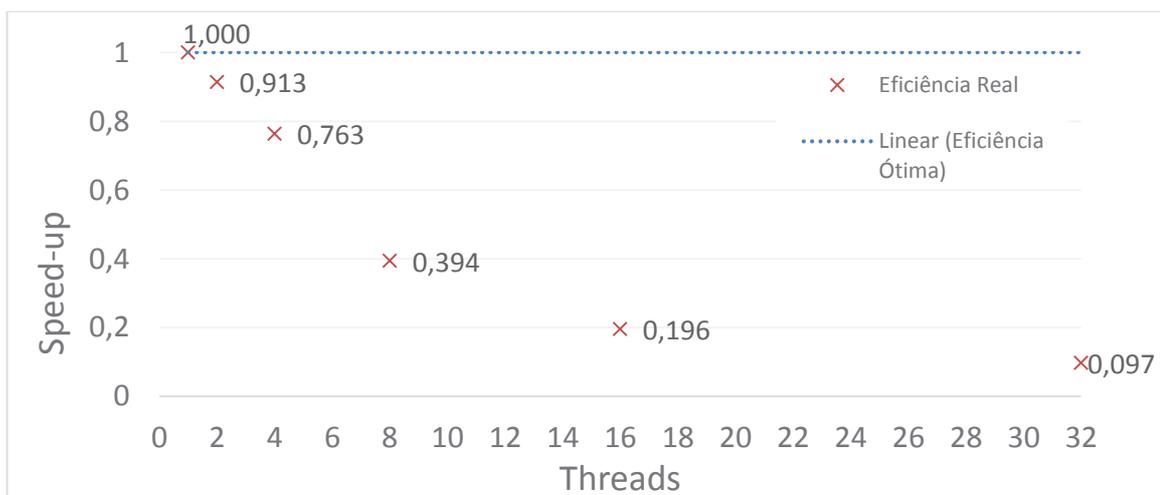
**Gráfico 1 - *Speed-up* da metaheurística híbrida paralela**



Fonte: Elaborado pelo autor.

A eficiência da abordagem paralela calcula a relação entre o ganho de velocidade (expresso pelo *speed-up*) e o número de elementos envolvidos na paralelização. A eficiência do AG+BT proposto pode ser observada no Gráfico 2.

**Gráfico 2 - Eficiência da metaheurística híbrida paralela**



Fonte: Elaborado pelo autor.

Analisando os dados obtidos a partir dos experimentos de temporização, é possível observar um *speed-up* sublinear, devido às partes essencialmente sequenciais do código, comprovando, assim, a Lei de Amhdal. O maior *speed-up* obtido nos experimentos realizados

foi na aplicação com 8 threads, atingindo valor de  $S(8) = 3,15$ . A análise de eficiência mostrou um desempenho acima de 75% na aplicação com 4 threads, apresentando queda abrupta ao executar o AG+BT com 8 threads.

Em uma análise qualitativa dos resultados obtidos, conclui-se que a paralelização desta aplicação metaheurística híbrida apresenta um ganho de desempenho de tempo em relação ao algoritmo sequencial, o que corrobora com o emprego de estratégias de HPC, pois evidencia as possibilidades de aprimoramento e aplicação da pesquisa em problemas reais. Apesar deste ganho, o grão fino da paralelização incide em um alto *overhead* de comunicação entre as threads, limitando, assim, o *speed-up* e a eficiência da mesma.

Além de mostrar desempenho superior ao algoritmo sequencial, a aplicação paralela apresentou valores de minimização idênticos ao algoritmo sequencial, evidenciando que os resultados de otimização desejados não foram comprometidos.

## 6 CONSIDERAÇÕES FINAIS

A eficiência de um terminal portuário está diretamente ligada ao planejamento eficaz da estiva dos navios porta-contêineres, sendo mensurada pela capacidade de carregar mais contêineres em menos tempo. Devido à importância do transporte marítimo na economia, é essencial que os esforços da comunidade científica se voltem para os problemas relacionados aos terminais portuários. Os problemas de eficiência já foram destacados e carecem de técnicas que possam fornecer soluções satisfatórias em tempo relativa e operacionalmente aceitáveis. Neste contexto, o trabalho aqui apresentado buscou contribuir para a solução do problema de planejamento de estiva de navios porta-contêineres.

A codificação da solução através de regras de carregamento 3D foi inicialmente adotada por Azevedo *et al.* (2011). Para o problema de planejamento de estiva de navios porta-contêineres, abordado neste trabalho, e utilizando a codificação por regras, o emprego de um número maior de regras implica em uma expansão do espaço de busca das soluções viáveis, ampliando a probabilidade de encontrar soluções diferentes e promissoras. Dessa forma, este trabalho sugeriu o emprego de 24 novas estratégias de carga e descarga nas abordagens aqui propostas, triplicando, assim, a quantidade de regras propostas na literatura correlata.

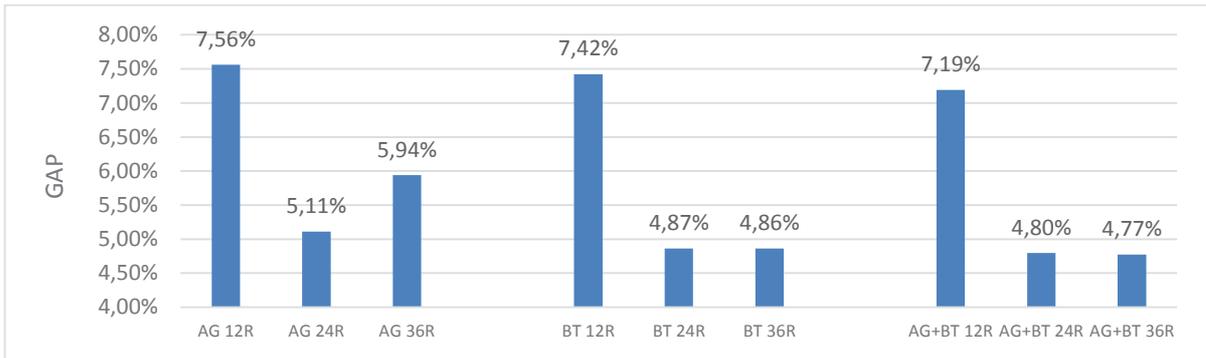
A aplicação de metaheurísticas no problema de planejamento de estiva dos navios porta-contêineres, conforme evidenciado na revisão bibliográfica, é uma técnica já estudada. Neste sentido, a pesquisa aqui relatada propõe três métodos metaheurísticos distintos: o Algoritmo Genético, a Busca Tabu e a combinação híbrida destas duas técnicas, a fim de comprovar que a hibridização produz resultados mais atrativos que as metaheurísticas puras, em virtude da união das características específicas de diversificação do AG e de intensificação da BT. As metaheurísticas propostas foram aplicadas ao PPE e comparados quantitativamente, sendo que o uso da BT em associação à codificação por regras é uma contribuição deste trabalho, pois ainda não há relato do uso combinado destas técnicas. A hibridização, tal como foi proposta, também é uma contribuição deste trabalho.

A partir dos resultados obtidos nos experimentos com a BT, pode-se observar que aplicação da técnica metaheurística associada à codificação por regras é uma alternativa eficiente para a minimização de movimentação, visto que apresentou uma redução de 0,1% em relação ao estado da arte apresentado na literatura, atingindo GAP médio de 4,86% para os problemas-teste avaliados.

Em relação ao AG+BT, observa-se que, para os 15 problemas-teste avaliados, o valor médio entre as FO Média é menor para o AG+BT nas 3 configurações de regras, o que evidencia a capacidade do AG+BT em encontrar soluções mais próximas do ótimo global do que as metaheurísticas puras. Comparado ao algoritmo de Azevedo *et al.* (2011), o AG+BT de 36 regras apresentou redução da média de GAP's para 4,77%, melhorando o resultado encontrado no trabalho de Azevedo *et al.* (2011), que é de 4,96%.

Considerando o fato de que o aumento de regras implica no aumento da probabilidade de encontrar resultados melhores, ao se observar os resultados obtidos após cada um dos testes realizados, pode-se verificar que, de fato, o aumento do número de regras possibilitou a obtenção de melhores resultados para a BT e para o AG+BT, tanto na comparação entre 12 *versus* 24 regras quanto na comparação 24 *versus* 36 regras. O Gráfico 3 mostra a média percentual dos desvios dos 15 problemas-teste utilizados neste trabalho, para cada algoritmo.

**Gráfico 3 - Análise dos GAP's (%) por algoritmo**

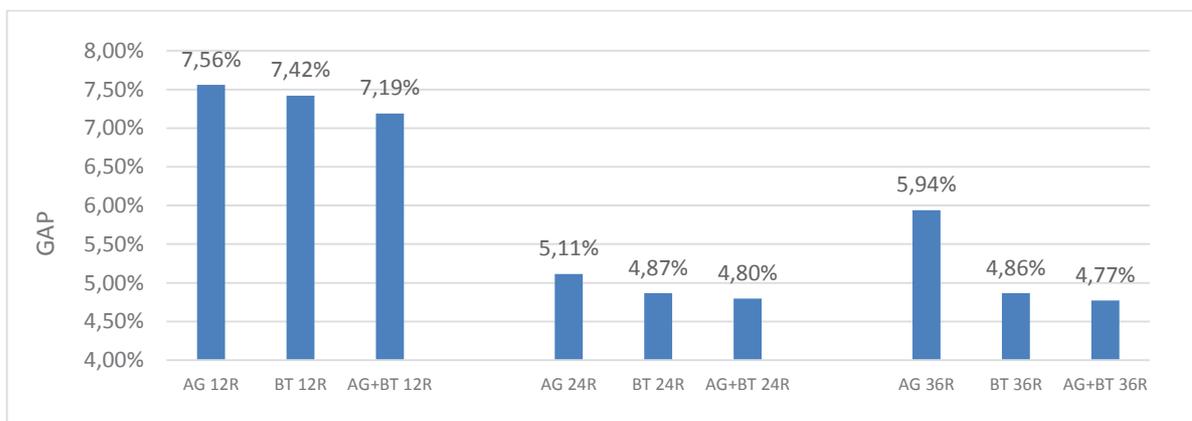


Fonte: Elaborado pelo autor.

No caso do AG, a utilização da configuração de 36 regras não implicou na melhora de resultados em relação ao AG 24 regras. A causa desta discrepância em relação aos outros algoritmos pode ser atribuída à perda prematura de diversidade da população, impossibilitando a exploração eficaz do espaço de soluções. Outra hipótese é que a utilização das últimas 12 regras (no caso da configuração 36 regras) estão ligadas a uma regra de descarga que tem um impacto significativo na FO, pois a descarga relacionada a estas regras retira todos os contêineres de dentro do navio.

De acordo com o Gráfico 4, também pode-se observar que a BT apresentou melhores resultados que o AG para todas as configurações de número de regras, assim como o AG+BT mostrou maior eficiência em relação à BT e, conseqüentemente, em relação ao AG.

**Gráfico 4 - Análise dos GAP's (%) por configuração de regras**



Fonte: Elaborado pelo autor.

Os dados e registros evidenciados a partir dos experimentos realizados permitem chegar à conclusão de que o algoritmo híbrido proposto (AG+BT) apresentou maior eficiência que o Algoritmo Genético e a Busca Tabu quando empregadas separadamente, visto que a média do desvio percentual, resultante da execução dos mesmos, foi menor para a abordagem híbrida AG+BT. Além disso, o uso de novas regras, como pôde ser visto na etapa de experimentos, possibilitou a melhora dos resultados obtidos por Azevedo *et al.* (2011), comprovando que as novas regras propostas são eficientes e que o uso de um número maior de regras viabiliza um incremento na eficiência da operação do terminal.

A importância de considerar o critério de estabilidade do navio foi abordada neste trabalho. A relevância desta restrição, somada à carência de soluções que abordem o cálculo de estabilidade, motivou a contribuição deste estudo no sentido de empregar o cálculo da estabilidade conforme literatura técnica especializada, redundando este cálculo em uma otimização de múltiplos objetivos. Com o emprego de uma parcela referente à estabilidade, foi possível mostrar o conflito de objetivos existente com a minimização da movimentação e a importância de considerar a instabilidade em conjunto com a movimentação. Os resultados mostraram que a proposta de uma abordagem conjunta pode fornecer soluções otimizadas para ambos objetivos.

A Tabela 15 mostra os resultados do AG+BT configurado para otimização de ambos objetivos e é possível verificar que a abordagem com múltiplos objetivos permite otimizar a movimentação e a estabilidade da embarcação simultaneamente, gerando soluções de compromisso entre dois objetivos diferentes e conflitantes. Assim, obtém-se uma ferramenta computacional capaz de fazer a gestão do número de remanejamentos e da instabilidade.

Analisando os resultados obtidos e as considerações realizadas, conclui-se que o trabalho proposto cumpriu com os objetivos propostos. Apesar de apresentar bons resultados, que contribuem de maneira significativa para a solução do problema abordado, no decorrer desta pesquisa foram evidenciados alguns aspectos nos quais há margem para novas pesquisas e aprimoramento das técnicas já utilizadas, listados a seguir:

- a) Utilização do conceito de hiper-heurística na sintonia dos parâmetros das metaheurísticas, buscando uma melhor eficiência dos algoritmos;
- b) Submeter o algoritmo AG+BT a uma autêntica abordagem de otimização multiobjetivo, com a tarefa de encontrar uma solução de compromisso dentro de um conjunto ótimo de Pareto;
- c) Realizar testes assumindo problemas-teste maiores, seja com relação ao número de portos que constituem a rota, seja na dimensão do navio;
- d) Considerar a aplicação da abordagem proposta em problemas-teste que representem navios de estrutura celular não-regular, isto é, que representem navios com baias de formatos diferentes, aproximando-se assim da realidade de algumas embarcações que possuem baias com escotilha ou baias de tamanho diferenciado à proa ou à popa;

- e) Considerar um custo distinto de operação em cada porto, aproximando-se, assim, da realidade e tornando a otimização mais efetiva em relação ao custo de operação do terminal;
- f) Realizar um estudo probabilístico do emprego das regras propostas, avaliando a contribuição de cada uma delas em relação a determinado objetivo e concluindo acerca da eficiência e da aplicabilidade de cada regra;
- g) Propor uma abordagem paralela para os algoritmos AG e BT, propondo, assim, uma nova dinâmica de hibridização;
- h) Promover a integração do PPE com problemas relacionados, como PAB, PEG, entre outros.

## REFERÊNCIAS BIBLIOGRÁFICAS

- AAPA - American Association of Port Authorities. **Statistics/World Port Rankings (2011 a 2013)**. Disponível em: <<http://www.aapa-ports.org/>>. Acesso em: 1 junho 2015.
- ALBA, E. **Parallel Metaheuristics: a new class of algorithms**. John Wiley & Sons, 2005.
- ÁLVAREZ, J. F. **A Heuristic for Vessel Planning in a Reach Stacker Terminal**. Journal of Maritime Research, v. III, n. 1, p. 3-16, 2006.
- 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.; PENN M., 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.; PENN, M.; SHPIRER, N. **Container Ship Stowage Problem: complexity and connection to the coloring of circle graphs**. Discrete Applied Mathematics, v. 103, p. 271-279, 2000.
- AZEVEDO, A. T., **Problemas & Dados, Instâncias Empregadas nos Artigos do SIMPEP, 2009**. Disponível em: <<https://sites.google.com/site/projetonavio/programas-dados>>. Acesso em: 10 outubro 2014.
- AZEVEDO, A. T.; SOBRAL, C. M.; DEUS, N. M. R., Resolução do problema de carregamento e descarregamento de contêineres em terminais portuários via algoritmo genético. **XVI Simpósio de Engenharia de Produção**, 2009.
- AZEVEDO, A. 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.
- AZEVEDO, A. T.; RIBEIRO, C. M.; NETO, L. L. S.; SILVA, M. P. E.; SILVESTRE, M. C. Resolução do problema de carregamento e descarregamento 3d de contêineres em terminais portuários via representação por regras e algoritmo genético. **Revista GEPROS – Gestão da Produção, Operações e Sistemas**, ano 6, n. 4, out-dez/2011, p. 91-110, 2011.
- BARRASS, B.; DERRETT, D. R. **Ship Stability for Masters and Mates**. 6th ed., 2006. Butterworth-Heinemann, Oxford.

- BARROS, A. J. M. **O Negócio Atlântico**: as redes comerciais portuenses e as novas geografias do trato internacional (séculos XVI-XVII). Revista da Faculdade de Letras, Porto, III Série, v. 8, p. 29-47, 2007.
- BIERWIRTH, C.; MEISEL, F. **A Survey of Berth Allocation and Quay Crane Scheduling Problems in Container Terminals**. European Journal of Operational Research 202, p. 615-627, 2010.
- BOGOSSIAN, M. P. **Entraves Operacionais Portuários**: plataforma de análise comparativa. Universidade de Brasília, 2011.
- BROWN, G. G.; LAWPHONGPANICH, S.; THURMAN, K. P. **Optimizing Ship Berthing**. Naval Research Logistics, v. 41, p. 1-15, 1994.
- CANTÚ-PAZ, E., **Efficient and Accurate Parallel Genetic Algorithms**. Dordrecht, The Netherlands: Kluwer Academic Publishers Group, 2000.
- CANTÚ-PAZ, E.; GOLDBERG, D. E. **Efficient Parallel Genetic Algorithms**: theory and practice. Technical report, University of Illinois at Urbana-Champaign, 1999.
- CARRARO, L. F.; CHIWIACOWSKY, L. D.; GÓMEZ, A. T. Uma aplicação das metaheurísticas algoritmo genético e colônia artificial de abelhas através da codificação por regras para resolver o problema de carregamento de navios-contêineres. **Anais do SBPO**, 1792-1803, 2013.
- CCE (Comissão das Comunidades Europeias). **Objetivos Estratégicos e Recomendações para a Política Comunitária de Transporte Marítimo no Horizonte de 2018**. Comunicação da Comissão ao Parlamento Europeu, ao Conselho, ao Comitê Econômico e Social Europeu e ao Comitê das Regiões, 2009.
- CHAN, F. T. S.; MAK, K. L.; WONG, Y. S. **A Genetic Algorithm for Facility Layout Problems**. Computers Integrated Manufacturing Systems. 11 (1-2), 113-12, 1998.
- CHIWIACOWSKY, L. D.; MONTEIRO, R. S. **Uma Aplicação da Metaheurística Algoritmo Genético ao Problema de Carregamento de Navios-Contêineres**. Revista Pesquisa Operacional para o Desenvolvimento. Rio de Janeiro, v. 4, n. 1, p. 52-64, 2012.
- CHOO, S. **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, 2005.
- COOK, S. A. **An overview of Computational Complexity**. Communications of the ACM. v. 26, n. 6, p. 401-7, junho de 1983.
- COOK, S. A. **The Complexity of Theorem Proving Procedures**. Proc. 3rd ACM Syrup on Theory of Computing. Shaker Heights, Ohio, p.151-158, 1971.

CRAINIC, T.; TOULOUSE, M.; GENDREAU, M. **Toward a Taxonomy of Parallel Tabu Search Heuristics**. *INFORMS Journal on Computing*, 9, p. 61-71, 1997.

DALE, N.; LEWIS, J. **Computer Science Illuminated**. Jones and Bartlett Publishers, 5th Ed., 2012.

DIAS, J. S.; GROSKO, A. P.; GORSKI, J. R. **Algoritmo Genético: revisão histórica e exemplificação**. 2006.

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. 585-599, 2002.

FERREIRA, I. M.; NETO, C. C. **Estrutura Tarifária dos Principais Portos Nacionais**. *Revista Desafios do Desenvolvimento – IPEA*, ed. 64, 2011.

FRAGA, M. C. P. **Uma Metodologia Híbrida Colônia de Formigas - Busca Tabu - Reconexão por Caminhos para Resolução do Problema de Roteamento de Veículos com Janelas de Tempo**. Dissertação de Mestrado. CEFET/MG. 2006.

GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability**. A guide to the theory of NP-Completeness, W.H. Freeman and Company: New York, 1979.

GIEMSCH, P.; JELLINGHAUS, A. **Optimization Models for the Containership Stowage Problem**. *International Conference on Operations Research Proceedings*, Berlin, Germany. Springer, 2003.

GILLMER, T. C. **Fundamentals of Construction and Stability of Naval Ship**, 2.ed, The United States Naval Institute, 1959.

GLOVER, F. **Future Paths for Integer Programming and Links to Artificial Intelligence**. *Computers and Operations Research*, Oxford, UK, v. 13, p. 533-549, 1986.

GLOVER, F.; LAGUNA, M. Tabu search. In: Reeves, C.R. (ed.) **Modern Heuristic Techniques for Combinatorial Problems**, p. 70-150. Blackwell Scientific Publications, Oxford, 1993.

GLOVER, F.; LAGUNA, M. **Tabu Search**. Kluwer Academic Publishers. Boston, 1997.

GOLDBARD, M. C.; GOLDBARD, E. F. G.; LUNA, H. P. L. **Programação Linear e Fluxo em Redes**. Elsevier Brasil, 2015.

GOLDBERG, D. E. **Genetic Algorithms: in search, optimization & machine learning**. Addison Wesley, 1989.

- GÓMEZ, A. T.; WEISSHEIMER JÚNIOR, C. A. **Um Modelo Computacional Baseado em Metaheurísticas Aplicado a Transmissão Internet Protocol Television (IPTV)**. Revista de Radiodifusão - SET, v.6, n.6, 2012.
- GRAMA, A.; GUPTA, A.; KARYPIS, G.; KUMAR, V. **Introduction to Parallel Computing**. Addison Wesley, 2003.
- GUAN, Y.; CHEUNG, R. K. **The Berth Allocation Problem**: models and solution methods. OR Spectrum, v. 26, p.75-92, 2004.
- GUILBERT, N.; PAQUIN, B. **Container Vessel Stowage Planning**. 2007.
- HE Y.; LIU G.; QIU Y. **A Parallel Tabu Search Algorithm Based on Partitioning Principles for TSPs**. IJCSNS Journal of Computer Science and Network Security, v. 6, n. 8A, 2006.
- HERTZ, A.; WIDMER, M. **Guidelines for the Use of Meta-Heuristics in Combinatorial Optimization**. European Journal of Operational Research 151, p. 247-252, 2003.
- HOLLAND, J. H., **Adaptation in Natural and Artificial Systems**. The University of Michigan Press, 1975.
- HÜE, X. **Genetic Algorithms for Optimization**: background and applications. Version 1.0, University of Edinburgh, 1997.
- IMAI, A.; NAGAIWA, K.; TAT, C-W. **Efficient Planning of Berth Allocation for Container Terminals in Asia**. Journal of Advanced Transportation, v. 31, p. 75-94, 1997.
- IMAI, A.; NISHIMURA, E.; HATTORI, M.; PAPADIMITRIOU S. **Berth Allocation at Indented Berths for Mega-Containerships**. European Journal of Operational Research, 179(2), p. 579–593, 2007.
- IMAI, A.; NISHIMURA E.; PAPADIMITRIOU S. **Berthing Ships at a Multi-User Container Terminal with a Limited Quay Capacity**. Transportation Research Part E, v. 44, p. 136-151, 2008.
- IMAI, A.; NISHIMURA, E.; PAPADIMITRIOU, S. **Berth Allocation with Service Priority**. Transportation Research Part B, .v. 37, p. 437-457, 2003.
- IMAI, A.; NISHIMURA, E.; PAPADIMITRIOU, S. **The Dynamic Berth Allocation Problem for a Container Port**. Transportation Research Part B, 35(4), p. 401-417, 2001.
- IMAI, A.; SASAKI, K.; NISHIMURA, E.; 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, p. 373-389, 2006.

IMAI, A.; SUN, X.; NISHIMURA, E.; PAPADIMITRIOU, S. **Berth Allocation in a Container Port**: using a continuous location space approach. *Transportation Research Part B* 39 (3), p. 199-221, 2005.

KASIM, H.; MARCH, V.; ZHANG, R. **Survey on Parallel Programming Model**. *Network and Parallel Computing*, v. 5245, p. 266-275, 2008.

LACERDA, S. M. **Navegação e Portos no Transporte de Contêineres**. *Revista do BNDES*, v. 11, n. 22, p. 215-243, 2004.

LAI, K. K.; SHIH, K. **A Study of Container Berth Allocation**. *Journal of Advanced Transportation*, v. 26, p. 45-60, 1992.

LEVINE, D. **A Parallel Genetic Algorithm for the Set Partitioning Problem**. Tese de Doutorado, Illinois Institute of Technology, 1994.

LEWIS, E. V. **Principles of Naval Architecture**. The Society of Naval Architecture and Marine Engineers, 1988.

LIANG, C.; HUANG, Y.; YANG, Y. **A Quay Crane Dynamic Scheduling Problem by Hybrid Evolutionary Algorithm for Berth Allocation Planning**. *Computers & Industrial Engineering* 56(3): p.1021-1028, 2009.

MATTSON T. G.; SANDERS B. A.; MASSINGILL B. L. **Patterns for Parallel Programming**. Addison Wesley, 2004.

MELIÁN, B.; PÉREZ, J.; VEGA, J. **Metaheuristics**: a global view. *Revista Iberoamericana de Inteligencia Artificial (Asociación Española de Inteligencia Artificial)*, 2(19), 2003.

MITCHELL, M. **An Introduction to Genetic Algorithms**. Cambridge, Massachusetts: MIT Press, 1996.

MONACO, M. F.; SAMMARRA, M.; SORRENTINO, G. **The Terminal-Oriented Ship Stowage Planning Problem**. *European Journal of Operational Research*, v. 239, p.256-265, 2014.

ÖLÇER, A. I. **A Hybrid Approach for Multi-Objective Combinatorial Optimisation Problems in Ship Design and Shipping**. *Computers & Operations Research* 35, p. 2760-2775, 2008.

PARK, Y. M.; K.H. KIM. **A Scheduling Method for Berth and Quay Cranes**. *OR Spectrum* 25, p.1-23, 2003.

PAZZINATO, A. L.; SENISE, M. H. V. **História Moderna e Contemporânea**. Editora Ática. 1992.

- PEDREIRA, A. F. **Os Recentes Avanços da Multimodalidade no Brasil**. Dissertação de Mestrado. PUC RIO, 2006.
- PEIXOTO, J. B. **Os Transportes no Atual Desenvolvimento do Brasil**. Rio de Janeiro: Biblioteca do Exército, 1977.
- RAIDL, G. R. **A Unified View on Hybrid Metaheuristics**. Hybrid Metaheuristics. Volume 4030, 2006, p. 1-12, 2006. Disponível em: <<https://link.springer.com/bookseries/558>>. Acesso em: 01 fevereiro 2015.
- RAYWARD-SMITH, V. J.; OSMAN, I. H.; REEVES, C. R.; SMITH, G. D. **Modern Heuristic Search Methods**, John Wiley & Sons, p. 1-25, 1996.
- ROCHA, I. **Uma Abordagem Otimizada para o Problema de Alocação de Equipes e Escalonamento de Tarefas para a Obtenção de Cronogramas Eficientes**. Dissertação de Mestrado. UECE, 2011.
- SANCHES, D. S. **Estratégia de Modelagem por Algoritmo Genético Adaptativo para Programação Reativa da Produção de Produtos com Uso Simultâneo de Máquinas e Sistemas de Transporte em Sistemas de Manufatura**. Dissertação de Mestrado. Universidade Federal de São Carlos, 2008.
- SCHRIJVER, A. **A Course in Combinatorial Optimization**. University of Amsterdam, 2012.
- SCHWEHM, M. **Parallel Population Models for Genetic Algorithms**, 1996.
- SCIOMACHEN, A.; TANFANI, E. **A 3D-BPP Approach for Optimizing Stowage Plans and Terminal Productivity**. European Journal of Operational Research, 2007.
- SCIOMACHEN, A.; TANFANI, E. **The Master Bay Plan Problem**: a solution method based on its connection to the three-dimensional bin packing problem, 2004.
- SHIELDS, J. J. **A Computer-Aided Preplanning System**. Marine Technol. 21, p. 370-383, 1984.
- SILVA, A. J. M. **Implementação de um Algoritmo Genético Utilizando o Modelo de Ilhas**. Dissertação de Mestrado, UFRJ, Rio de Janeiro, 2005.
- SILVA, G.; COCCO, G. **Cidade e Portos**: os espaços da globalização. Rio de Janeiro: DP&A, 1999.
- SPINRAD, J. P. **Recognition of Circle Graphs**, J. Algorithms 16, p. 264-282, 1994.
- STALLINGS, W. **Computer Organization and Architecture**: designing for performance. Pearson, 2009.

- SUNDAR, S.; SINGH, A.; ROSSI, A. **An Artificial Bee Colony Algorithm for the 0-1 Multidimensional Knapsack Problem**. University of Florida, Communications in computer and information science, v. 94, p. 141-151, 2010.
- TALBI, E. G., **Metaheuristics: from design to implementation**, John Wiley & Sons, New Jersey, 2009.
- TANENBAUM, A. S. **Sistemas Operacionais Modernos**. 2<sup>a</sup> ed. Pearson, 2003.
- TODD, D. S.; SEN, P. A multiple criteria genetic algorithm for container ship loading. **Proceedings of the Seventh International Conference on Genetic Algorithms**, p. 674-681. East Lansing, Michigan, USA, 1997.
- URSEM R. K. **Diversity-Guided Evolutionary Algorithms**. Proceedings of PPSN, p. 462-471, 2002.
- WANG, F.; LIM, A. **A Stochastic Beam Search for the Berth Allocation Problem**. Decision Support Systems, 2006.
- 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.
- 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.
- WRIGHT, S. J.; NOCEDAL, J. **Numerical optimization**. New York: Springer-Verlag, 1999.
- YANG, X. S. **Nature-Inspired Metaheuristic Algorithms**. Second Edition, University of Cambridge, 2010.