



Programa Interdisciplinar de Pós-Graduação em
Computação Aplicada

Mestrado Acadêmico

Carlos Eduardo de Jesus Guimarães Oliveira

**PCAIISO-GT: UMA METAHEURÍSTICA CO-EVOLUTIVA PARALELA
DE OTIMIZAÇÃO APLICADA AO PROBLEMA DE ALOCAÇÃO DE
BERÇOS**

**São Leopoldo
2014**

Carlos Eduardo de Jesus Guimarães Oliveira

**PCAIISO-GT: UMA METAHEURÍSTICA CO-EVOLUTIVA PARALELA
DE OTIMIZAÇÃO APLICADA AO PROBLEMA DE ALOCAÇÃO DE
BERÇOS**

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

Orientador: Dr. Leonardo Dagnino
Chiwiacowsky.

**São Leopoldo
2014**

Ficha catalográfica

O48p Oliveira, Carlos Eduardo de Jesus Guimarães
PCAISO-GT : uma metaheurística co-evolutiva paralela de
otimização aplicada ao problema de alocação de berços / por
Carlos Eduardo de Jesus Guimarães Oliveira. – 2014.
90 f. : il., 30 cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos
Sinos, Programa de Pós-Graduação em Computação Aplicada,
2014.
Orientação: Prof. Dr. Leonardo Dagnino Chiwiacowsky.

1. Computação natural. 2. Sistema imunológico artificial.
3. Otimização combinatória. Computação evolucionária. 4. Co-
evolução e paralelização. I. Título.

CDU 004.8

Catálogo na Fonte:
Bibliotecária Vanessa Borges Nunes - CRB 10/1556

AGRADECIMENTOS

Agradeço aos meus dois pais, **Acácio Fernando Brito Elleres** e **Francisco Carlos de Andrade Guimarães**, pelo crédito e confiança em mim depositados. A minha amada mãe, **Ilcycarla**, a quem devo todos os exemplos de bravura e dedicação. Aos meus avós, **Carlito** e **Ildecy**, pelos ensinamentos que até hoje me guiam. Ao meu querido tio **Toninho**, presente em todos os momentos importantes da minha vida. A minha terceira mãe, tia **Dalva**, que sempre me mostrou o caminho certo a seguir. A **Fernanda**, minha linda namorada e companheira inseparável que, mais do que ninguém, me apoiou em todos os momentos e nunca me deixou desistir. Ao meu orientador **Leonardo** que foi extremamente importante nessa caminhada, sempre acreditando no meu trabalho e me apoiando em todos os momentos. E por último agradeço a Deus, por sempre me guiar e nunca me deixar sair do caminho do bem.

RESUMO

Este trabalho apresenta um algoritmo de otimização baseado na metaheurística dos Sistemas Imunológicos Artificiais, princípios de Teoria dos Jogos, Co-evolução e Paralelização. Busca-se a combinação adequada dos conceitos de Teoria dos Jogos, Co-evolução e Paralelização aplicados ao algoritmo AISO (*Artificial Immune System Optimization*) para resolução do Problema de Alocação de Berços (*PAB*). Dessa maneira, o algoritmo é formalizado a partir das técnicas citadas, formando o PCAISO-GT: *Parallel Coevolutionary Artificial Immune System Optimization with Game Theory*. Inicialmente, foram realizados experimentos visando à sintonia dos parâmetros empregados nas diferentes versões da ferramenta desenvolvida. Com base nas melhores configurações identificadas, foram realizados experimentos de avaliação através da solução de um conjunto de instâncias do PAB. Os resultados obtidos permitiram a indicação da versão co-evolutiva associada à teoria dos jogos como a melhor para solução do problema em estudo.

Palavras Chave: Computação natural, Sistema imunológico artificial, Otimização Combinatória, Computação evolucionária, Co-evolução e Paralelização.

ABSTRACT

This paper presents an optimization algorithm based on metaheuristic of Artificial Immune Systems, principles of Game Theory, Co-evolution and parallelization. The objective is find the appropriate combination of the concepts of Game Theory, Co-evolution and Parallelization applied to AISO algorithm (Artificial Immune System Optimization) for solving the Berth Allocation Problem (BAP). Thus, the algorithm is formalized from the above mentioned techniques, forming the PCAISO-GT: Parallel Coevolutionary Artificial Immune System Optimization with Game Theory. Initially, experiments aiming to tune the parameters were performed using different versions of the tool developed. Based on the identified best settings, evaluation experiments were carried out by solving a set of instances of the PAB. The results obtained allowed the appointment of co-evolutionary version associated with game theory as the best solution to the problem under study.

Keywords: Natural computation, artificial immune system, Combinatorial Optimization, Evolutionary Computation, Co-evolution and parallelization.

LISTA DE FIGURAS

FIGURA 2-1 ANTICORPOS ATACANDO UM ANTÍGENO	22
FIGURA 2-2 TABELA DE PAGAMENTOS DO JOGO DILEMA DO PRISIONEIRO (RAPOPORT, 1965).....	28
FIGURA 3-1 CARACTERIZAÇÃO DO BERÇO EM UM PORTO.....	35
FIGURA 4-1 REPRESENTAÇÃO DAS VARIÁVEIS DE TEMPO (MAURI ET AL. 2008A)	41
FIGURA 4-2 CARACTERIZAÇÃO DA CÉLULA AISO	45
FIGURA 4-3 FLUXOGRAMA DO ALGORITMO AISO	46
FIGURA 4-4 EXEMPLO DE UMA SOLUÇÃO PARA O PROBLEMA DE 3 BERÇOS E 9 NAVIOS	47
FIGURA 4-5 REPRESENTAÇÃO DA CÉLULA PARA A SOLUÇÃO DA FIGURA 4-4.....	48
FIGURA 4-6 EXEMPLO DE MUTAÇÃO NA SOLUÇÃO AISO	49
FIGURA 4-7 CARACTERIZAÇÃO DA CÉLULA DO AISO-GT	50
FIGURA 4-8 FLUXOGRAMA DO ALGORITMO AISO-GT	51
FIGURA 4-9 REPRESENTAÇÃO DO MODELO <i>STEPPING STONE</i>	53
FIGURA 4-10 REPRESENTAÇÃO DO MODELO ÍLHA	54
FIGURA 4-11 SIMULAÇÃO DE ACESSO À MEMÓRIA COMPARTILHADA.....	56
FIGURA 4-12 SIMULAÇÃO DE ACESSO À MEMÓRIA COMPARTILHADA COM SEMÁFORO.....	57
FIGURA 4-13 CARACTERIZAÇÃO DA CÉLULA DO PCAISO-GT	58
FIGURA 4-14 FLUXOGRAMA DO ALGORITMO PCAISO-GT	59
FIGURA 6-1 GRÁFICO DE EVOLUÇÃO DO ALGORITMO AISO	72
FIGURA 6-2 GRÁFICO DE EVOLUÇÃO DO ALGORITMO AISO	73
FIGURA 6-3 GRÁFICO DE EVOLUÇÃO DO ALGORITMO PCAISO.....	74
FIGURA 6-4 GRÁFICO DE EVOLUÇÃO DO ALGORITMO PCAISO.....	75
FIGURA 6-5 GRÁFICO DE EVOLUÇÃO DO ALGORITMO PCAISO.....	76
FIGURA 6-6 GRÁFICO DE EVOLUÇÃO DO ALGORITMO PCAISO-GT	77
FIGURA 6-7 GRÁFICO DE EVOLUÇÃO DO ALGORITMO PCAISO-GT	77
FIGURA 6-8 GRÁFICO COMPARATIVO DE <i>SPEEDUP</i>	80
FIGURA 6-9 GRÁFICO COMPARATIVO DE EFICIÊNCIA	80

LISTA DE TABELAS

TABELA 3-1 LISTA DA MOVIMENTAÇÃO DE CONTÊINERES DOS PRINCIPAIS PORTOS DO MUNDO.....	33
TABELA 3-2 LISTA DO PREÇO DE MOVIMENTAÇÃO DE UM CONTAINER EM US\$ EM ALGUNS PORTOS DO MUNDO...	34
TABELA 5-1 RESULTADOS (LOPES ET AL. 2011).....	62
TABELA 5-2 SOLUÇÕES ÓTIMAS DAS INSTÂNCIAS DE TESTE.....	64
TABELA 5-3 PARÂMETROS DE TESTE	65
TABELA 5-4 CONJUNTO DE PARÂMETROS DE TESTE	66
TABELA 5-5 PARÂMETROS DE TESTE DO ALGORITMO CO-EVOLUTIVO	67
TABELA 5-6 CONJUNTO DE PARÂMETROS DE TESTE DO ALGORITMO CO-EVOLUTIVO	67
TABELA 5-7 PARÂMETROS DE TESTE DO ALGORITMO GT	68
TABELA 5-8 CONJUNTO DE PARÂMETROS DE TESTE DO ALGORITMO GT	68
TABELA 6-1 RESULTADOS CONSOLIDADOS DOS TESTES	71
TABELA 6-2 COMPARATIVO ENTRE METAHEURÍSTICAS.....	79
TABELA 6-3 TEMPO MÉDIO DE EXECUÇÃO, EFICIÊNCIA E <i>SPEEDUP</i>	81

SUMÁRIO

1	INTRODUÇÃO	17
1.1	OBJETIVOS DO TRABALHO	19
1.2	ESTRUTURA DO TRABALHO	19
2	FUNDAMENTAÇÃO TEÓRICA.....	21
2.1	SISTEMAS IMUNOLÓGICOS	21
2.1.1	Sistema Imunológico Humano.....	21
2.1.2	Sistema Imunológico Artificial.....	23
2.1.2.1	Teoria da Seleção de Clones.....	23
2.1.2.2	Reforço de Aprendizado e Memória Imunológica.....	24
2.1.2.3	Hipermutação Somática.....	24
2.1.3	TRABALHOS RELACIONADOS	25
2.2	TEORIA DOS JOGOS (<i>GAME THEORY</i>).....	27
2.3	CO-EVOLUÇÃO	29
2.3.1	Modelo Co-evolutivo Genético	30
2.4	RESUMO	30
3	PROBLEMA DE ALOCAÇÃO DE BERÇOS.....	32
3.1	TRANSPORTE MARÍTIMO	32
3.2	OPERAÇÃO PORTUÁRIA	34
3.3	DEFINIÇÃO DO PROBLEMA	37
3.4	TRABALHOS RELACIONADOS	38
3.5	RESUMO	39
4	METODOLOGIA PROPOSTA	41
4.1	MODELO MATEMÁTICO DO PAB	41
4.2	ALGORITMO AISO (<i>ARTIFICIAL IMMUNE SYSTEM OPTIMIZATION</i>)	44
4.3	AISO APLICADO AO PAB.....	47
4.3.1	Codificação da Solução	47
4.4	TEORIA DOS JOGOS APLICADA AO AISO	49
4.5	ARQUITETURA PARALELA	51
4.5.1	Paralelização de AIS.....	52
4.5.2	Modelo de Paralelização em Subpopulações.....	53
4.5.3	Técnicas de Paralelização	55
4.6	PROPOSTA DA NOVA METAHEURÍSTICA DE BUSCA.....	57
4.6.1	Proposta do Algoritmo PCAISO-GT	57
4.7	TRABALHOS RELACIONADOS	60
4.8	RESUMO	60
5	VALIDAÇÃO	62
5.1	INSTÂNCIAS DE TESTE	62
5.2	VALIDAÇÃO E SINTONIA DOS ALGORITMOS.....	63
5.3	PARÂMETROS DA METAHEURÍSTICA.....	64
5.3.1	Parâmetros Gerais (AISO)	65

5.3.2	Parâmetros do Algoritmo Co-evolutivo (PCAIISO).....	66
5.3.3	Algoritmo com Teoria de Jogos (PCAIISO-GT).....	68
5.3.4	Resumo	69
6	EXPERIMENTOS E RESULTADOS	70
6.1	ALGORITMO AISO	72
6.2	ALGORITMO PCAISO	73
6.3	ALGORITMO PCAISO-GT	76
6.4	COMPARATIVO ENTRE METAHEURÍSTICAS	78
7	CONSIDERAÇÕES FINAIS.....	82
8	REFERÊNCIAS BIBLIOGRÁFICAS.....	84

1 INTRODUÇÃO

De acordo com Ballard (1999), a computação natural envolve todos os sistemas computacionais desenvolvidos com inspiração ou utilização de algum mecanismo natural de processamento de informação. Fundamentalmente, a computação natural é constituída por novas abordagens computacionais caracterizadas por uma maior proximidade com a natureza. A computação natural se subdivide em grupos, sendo um deles a Computação Evolucionária que compreende diversos algoritmos inspirados no princípio Darwiniano da evolução das espécies e na genética (TEIXEIRA, 2005).

Os algoritmos evolucionários são probabilísticos e fornecem um mecanismo de busca paralela e adaptativa, baseado no princípio de sobrevivência dos mais aptos e na reprodução. O mecanismo é obtido a partir de uma população de indivíduos (soluções), codificados de diferentes formas (palavras binárias, vetores, matrizes), cada um associado a uma aptidão (avaliação da solução do problema), que são submetidos a um processo de evolução (seleção, cruzamento e mutação) por vários ciclos (TEIXEIRA, 2005).

Um exemplo clássico são os Algoritmos Genéticos (AG) criados por John Holland (HOLLAND, 1975), que representam uma solução potencial para um problema específico numa estrutura semelhante à de um cromossomo, e aplicam operadores de seleção e cruzamento (*crossover*) a essas estruturas, de forma a preservar informações críticas relativas à solução do problema.

Muitos algoritmos tiveram inspiração na natureza, como as Redes Neurais Artificiais que se baseiam no funcionamento do cérebro humano (TEIXEIRA, 2005); as Colônias de Formigas que se baseiam no comportamento das formigas, sua forma de organização e cooperação (TEIXEIRA, 2005); os Enxames de Partículas que se baseiam no comportamento dos enxames de insetos, bando de pássaros e outros grupos (TEIXEIRA, 2005); e os Sistemas Imunológicos Artificiais que se baseiam no sistema imunológico humano.

O Sistema Imunológico Artificial (*Artificial Immune System* - AIS) é uma técnica inspirada nos princípios e processos do sistema imunológico humano. O AIS representa uma classe de Algoritmos Evolutivos (AE) inspirados em mecanismos imunológicos de seleção e reprodução. Como em outras metaheurísticas, por exemplo Algoritmos Genéticos (AG), as soluções candidatas são representadas por indivíduos em uma população, que interagem com

o problema, passando por um processo de otimização no qual os mais aptos têm mais chances de sobreviver.

O AIS é uma técnica relativamente nova, foi criada em meados dos anos 80 por Farmer, Packard e Perelson (FARMER et al. 1986), entretanto apenas em meados dos anos 90 veio se tornar uma área de estudo. Os estudos começaram em 1994 com os Algoritmos de Seleção Negativa (FORREST et al. 1994). Em 1995 Hunt e Cooke começaram a trabalhar em Modelos de Redes Imunológicas (COOKE et al. 1995). Em 2002, um conjunto de quatro estudiosos formado por De Castro e Von Zuben (DE CASTRO et al. 2002) e Nicosia e Cutello (NICOSIA et al. 2002) publicaram informações sobre a Seleção de Clones. O primeiro livro foi publicado em 1999 por Dasgupta (DASGUPTA, 1999).

Atualmente, o AIS é alvo de muitos estudos, um deles tange a sua utilização como ferramenta de otimização multi-função, onde, diferentemente de outras técnicas que conseguem otimizar apenas uma função por vez, várias funções podem ser otimizadas de uma vez só (DE CASTRO et al. 2002).

Nesse contexto de crescente interesse pelo AIS, busca-se aplicá-lo a problemas complexos com o objetivo de avaliar sua capacidade de solução. No contexto de transportes, é vasto o número de problemas com relevância para o desenvolvimento econômico do país e mundial. Especificamente no âmbito do transporte marítimo, são identificados diferentes problemas relacionados à operação de terminais portuários, com destaque para o Problema de Alocação de Berços (PAB).

O PAB vem sendo estudado desde a introdução dos contêineres no transporte marítimo, que aconteceu no início do anos 50. Nos anos 90, o número de contêineres quintuplicou em relação aos anos 50 (UNCTAD, 2009). Existem, atualmente, mais de 2000 portos no mundo, buscando uma forma simples de lidar com o manuseio de cargas e buscando aumentar a sua capacidade de movimentação de cargas, de poucas centenas de toneladas por ano, em 1950, para mais de 300 milhões de toneladas, nos dias atuais.

O PAB, de modo simples, consiste em atribuir os navios que chegam ao porto a uma posição no cais de atracação, chamada berço. Entretanto, essa simples ação envolve uma grande quantidade de variáveis que devem ser consideradas e avaliadas para descobrir o melhor momento e a melhor posição para alocar o navio.

Tendo isso em vista, nesse trabalho é utilizado o AIS com a aplicação de conceitos de Teoria dos Jogos e Co-evolução para resolver o PAB, buscando avaliar a metaheurística quando empregada na solução de um problema complexo e de importância destacada.

1.1 OBJETIVOS DO TRABALHO

O objetivo geral deste trabalho é desenvolver um algoritmo evolucionário baseado na técnica dos Sistemas Imunológicos Artificiais, utilizando conceitos de Teoria dos Jogos, Co-evolução e paralelização para solução do problema de alocação de berços, e para isso se faz necessário o alcance das seguintes metas específicas:

1. Compreender a técnica de otimização avançada, Sistemas Imunológicos Artificiais, no que se refere à inspiração, definição, aplicação e implementação;
2. Compreender a melhor maneira de utilizar a Teoria dos Jogos aplicada ao AIS;
3. Agregar conceitos de Teoria dos Jogos a técnica dos Sistemas Imunológicos Artificiais, criando assim uma versão do algoritmo com interação social.
4. Compreender os modelos de co-evolução existentes a fim de utilizar o melhor modelo na implementação do novo algoritmo;
5. Compreender as técnicas de Paralelismo atualmente existentes e verificar a melhor maneira de aplicá-las ao novo algoritmo;
6. Agregar técnicas de Paralelismo e conceitos de co-evolução ao algoritmo AISO, criando assim uma versão co-evolutiva e paralela da metaheurística;
7. Compreender o PAB no que se refere à definição, modelagem e aplicação para posterior implementação;
8. Compreender a melhor maneira de codificar a solução do PAB nas metaheurísticas propostas;
9. Realizar simulações para verificar o desempenho dos algoritmos;
10. Analisar os resultados obtidos nas simulações.

1.2 ESTRUTURA DO TRABALHO

Adicionalmente a este capítulo inicial de introdução, nos capítulos 2 e 3, são abordados os aspectos que fundamentam a teoria relacionada ao escopo do trabalho. No capítulo 2 é apresentada a visão biológica dos Sistemas Imunológicos sendo apresentadas as suas

principais características e, em seguida, é apresentado o modelo computacional que simula o funcionamento do modelo biológico, denominado Sistemas Imunológicos Artificiais. Ainda neste capítulo, são apresentadas as bases teóricas da Teoria de Jogos e dos conceitos de Co-evolução, visando a sua aplicação na solução do problema em estudo. No capítulo 3 é apresentado o PAB, problema cuja solução será buscada através do emprego do AIS. São descritos, detalhadamente, os fundamentos teóricos, sendo apresentados os principais conceitos e características.

No capítulo 4 é apresentado o modelo de algoritmo empregado, detalhando qual o modelo matemático utilizado para o PAB e os detalhes de como a implementação foi feita com base no AIS, sendo especificada a sua estrutura e funcionamento.

No capítulo 5, são apresentados os experimentos realizados para validação dos algoritmos propostos, através do detalhamento das instâncias e dos parâmetros passíveis de ajuste no algoritmo.

No capítulo 6 são apresentados os experimentos e resultados e, no capítulo 7 as considerações finais. Finalizando, as referências bibliográficas são apresentadas no capítulo 8.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os Sistemas Imunológicos, tanto na visão da Biologia quanto em uma visão Computacional, a Teoria dos Jogos e a Co-evolução.

2.1 SISTEMAS IMUNOLÓGICOS

Segundo DE CASTRO (2001), em sua tese, atualmente existe um grande interesse pelo estudo dos sistemas imunológicos. Pesquisadores da área de saúde, mais especificamente imunologistas e profissionais da área médica, estão tentando compreender melhor o funcionamento do sistema imunológico com o objetivo de combater doenças infecciosas e doenças auto-imunes, como são os casos da AIDS¹ e da Esclerose Múltipla².

Em outra vertente, biólogos e imunologistas teóricos tem interesse em modelar o sistema imunológico, visando a reproduzir fenômenos observados em experimentos e com isso ter a possibilidade de prever comportamentos ainda não obtidos em situações reais.

Outros pesquisadores que também têm demonstrado um interesse acentuado sobre este assunto são profissionais da área de computação. Através de estudos e experimentos, pesquisadores em computação estão tentando simular o comportamento imunológico, e com isso criar sistemas artificiais com o objetivo de resolver problemas de diferentes áreas das ciências.

2.1.1 Sistema Imunológico Humano

O Sistema Imunológico (SI) é um complexo de células, moléculas e organismos, que juntos funcionam como um mecanismo de identificação, capaz de perceber e combater disfunções de suas próprias células e microorganismos invasores (TIZARD, 1995). Ele é responsável pela “limpeza” do organismo, retirando células mortas, renovando determinadas estruturas, entre outras funções. Além disso, também age em células alteradas, as quais diariamente surgem no nosso corpo como resultado de mitoses anormais e, caso não sejam destruídas, estas podem vir a se tornar tumores.

¹ Síndrome da Imunodeficiência Adquirida.

² Doença na qual os anticorpos destroem a mielina, substância que envolve os axônios dos neurônios, fazendo com que o impulso nervoso se de mais vagarosamente, alterando as funções do cérebro e dos nervos.

As células do SI são altamente organizadas. Cada tipo de célula age de acordo com sua função (Figura 2-1). Umas são encarregadas de receber ou enviar mensagens de ataque e, em outros casos, mensagens de inibição. Algumas apresentam os inimigos ao “exército” do Sistema para que possam ser combatidos, e outras liberam substâncias que neutralizam o microorganismo invasor ou alguma substância liberada por ele. Junto com os outros sistemas do corpo humano, o SI permite a regulação do organismo, garantindo o seu funcionamento estável.

Existem dois sistemas que são responsáveis pelo reconhecimento de microorganismos invasores: o Sistema Imunológico Nativo (SIN) e o Sistema Imunológico Adaptativo (SIAd). O SIN é assim chamado porque tem a habilidade de reconhecer certos micróbios e imediatamente destruí-los, e o SIAd por possuir a habilidade de reconhecer e tentar adaptar os anticorpos para que sejam produzidos de forma a inibir o corpo estranho (ABUL, 2011).

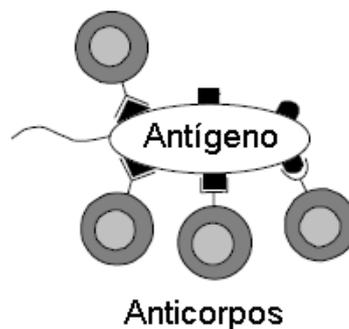


Figura 2-1 Anticorpos atacando um antígeno

Um importante componente desse sistema é uma classe de proteínas do sangue, que tem como função auxiliar os anticorpos no combate aos microorganismos. O aspecto mais importante do reconhecimento do SIN é o fato de induzir a geração de sinais coestimulatórios que irão resultar na ativação das células T, uma das principais células de defesa, promovendo o início da resposta do Sistema Imunológico Adaptativo.

O SIAd usa receptores de antígenos somáticos que são clonados e distribuídos em dois tipos de linfócitos: células T e B. Esses receptores são gerados por processos aleatórios e, assim, o formato geral da resposta do SIAd é baseado no princípio da Seleção Clonal dos linfócitos com particularidades específicas (BURNET, 1978).

2.1.2 Sistema Imunológico Artificial

Segundo DE CASTRO et al. (2002), os Sistemas Imunológicos Artificiais são sistemas adaptativos inspirados na teoria imunológica e na observação das funções imunológicas, sendo seus princípios e modelos aplicados na resolução de problemas. Os AIS emergiram nos anos 90 como uma nova área de pesquisa computacional. Eles seguem outros campos emergentes inspirados pelo comportamento biológico, como as redes neurais artificiais e a vida artificial. Essa nova informática imunológica tem sido investigada e aplicações práticas desenvolvidas, como algumas apresentadas por DASGUPTA (1999) e FORREST et al. (1994).

Segundo DASGUPTA (1999), os AIS permitiram o uso de modelos computacionais para processamento de informações em inter-reações imunológicas, com aplicações práticas em muitos problemas como reconhecimento de padrões, mineração de dados, segurança computacional e detecção de falhas.

Muitos algoritmos inspirados nos princípios do AIS já foram propostos, com destaque para os seguintes algoritmos:

- ClonalG (DE CASTRO et al. 2002);
- Opt-AiNet (DE CASTRO et al. 2002);
- B-Cell Algorithm (KELSEY et al. 2003).

A seguir, são descritos os modelos e princípios do AIS, e sua relevância para o processo de evolução da população de indivíduos.

2.1.2.1 Teoria da Seleção de Clones

O processo de reprodução no Sistema Imunológico é realizado de forma assexuada (clones), dessa maneira se faz necessária uma seleção dos clones para que o sistema possa evoluir. A teoria parte da premissa que para evoluir é preciso reproduzir somente as células que reconhecem os antígenos.

Conforme BURNET (1978), os principais passos da teoria de seleção de clones são:

- Mutacionar as células-clone (cópias dos seus pais) com altas taxas (hipermutação somática);
- Eliminar os linfócitos atualmente diferenciados que apresentem receptores similares;

- Proliferar e diferenciar o contato das células maduras com os antígenos;
- Eliminar os clones que possam provocar a involução do sistema.

A Seleção Clonal é o processo responsável por explorar o espaço de busca a procura de soluções satisfatórias, tendo como objetivo manter a diversidade na população.

2.1.2.2 Reforço de Aprendizado e Memória Imunológica

A Teoria de Seleção de Clones por si só não proporciona um eficiente combate aos microorganismos invasores, pois em caso de re-infecção o processo teria que ser iniciado novamente. Conseqüentemente, passa a ser necessária uma memória auxiliar que armazene as configurações das células que apresentam maior adaptabilidade.

Quando o Sistema Imunológico detecta uma célula estranha, é iniciado um processo de geração de linfócitos que, de acordo com sua afinidade com o antígeno, é ou não clonado. Em caso de alta afinidade, a célula é clonada e os clones se diferenciam em **Células de Memória**, responsáveis por manter as configurações das células com maior adaptabilidade, pois possuem um tempo de vida maior, e **Células de Plasma**, responsáveis pela produção de anticorpos, os quais têm como finalidade o combate às células estranhas.

A primeira resposta do Sistema a um determinado antígeno é lenta, pois os processos de clonagem e mutação ainda serão realizados para geração das células de combate eficientes para aquele determinado antígeno. Já a segunda resposta é mais rápida, pois só é necessário o processo de clonagem nas células de memória as quais resultarão em células de combate eficientes.

Dessa maneira, percebe-se que Aprendizagem e Memória Imunológica é o processo responsável por manter na população as melhores soluções, promovendo a rápida evolução tendo em vista que a qualidade das soluções durante o processo tende sempre a aumentar.

2.1.2.3 Hipermutação Somática

Durante a clonagem, é realizado um processo de mutação com o objetivo de otimizar as células. O processo é denominado hipermutação somática, e provê às células com maior adaptabilidade uma taxa de mutação baixa, enquanto as células com baixa adaptabilidade, as quais são consideradas pobres, são providas com taxas de mutação mais elevadas.

O objetivo do processo é tentar transformar células com baixa adaptabilidade em boas soluções, e não deixar que as células consideradas boas regridam.

Sendo assim, percebe-se que o processo de Hipermutação Somática é o motor de otimização da metaheurística. Nessa etapa, os indivíduos são submetidos a alterações que podem resultar em melhores soluções.

2.1.3 TRABALHOS RELACIONADOS

Uma vez que o trabalho realizado foca no emprego da Seleção Clonal, a seguir serão apresentados trabalhos nessa área de pesquisa do AIS. O princípio da seleção clonal é todo o processo de identificação de antígeno, proliferação de células e diferenciação em células de memória (BURNET, 1959). Vários algoritmos AIS foram desenvolvidos baseando-se nos princípios e conceitos da seleção clonal.

DE CASTRO et al. (2002) propôs um algoritmo de seleção clonal chamado CLONALG para aprendizado e otimização. O algoritmo gera uma população de N anticorpos, cada um representando uma solução no processo de otimização. Em cada iteração, alguns anticorpos são selecionados, clonados e mutacionados, com o objetivo de criar uma nova população de soluções. Os novos anticorpos são avaliados e certa porcentagem dos melhores anticorpos é adicionada à população original. Finalmente, uma porcentagem dos piores anticorpos da geração anterior é substituída por novos anticorpos criados aleatoriamente.

Em RUOCHEN et al. (2003), é proposto um algoritmo de Estratégia Clonal (ICS - *Immunity Clonal Strategy Algorithm*) que inclui um algoritmo de Estratégia Monoclonal Imunológico (IMSA - *Immunity Monoclonal Strategy Algorithms*) e um algoritmo de Estratégia Policlonal Imunológico (IPSA - *Immunity Polyclonal Strategy Algorithm*). O ICS é utilizado para resolver problemas de otimização multi-objetivo.

LI et al. (2003) propõem um algoritmo Imunológico Artificial Caótico (CAIF - *Chaos Artificial Immune Algorithm*) para solução de problemas de otimização de funções. O algoritmo usa variáveis caóticas para realizar uma busca local e explorar o espaço de soluções.

GARRET (2004) apresenta um algoritmo de Seleção Clonal Adaptativo (ACS - *Adaptive Clonal Selection*), uma modificação do algoritmo CLONALG. Ele sugere algumas modificações no algoritmo CLONALG, baseado na análise dos operadores através da seleção

da quantidade de mutação e número de clones de forma a superar as limitações que o algoritmo CLONALG apresenta, como a inclusão de alguns parâmetros e a representação binária.

Um algoritmo de Estratégia Clonal Imunológico Adaptativo (AICSA - *Adaptive Immune Clonal Strategy Algorithm*) é proposto para a solução de problemas de otimização numérica por RUOCHEN et al. (2004). O algoritmo escolhe dinamicamente a unidade de memória imunológica através da afinidade Ab-Ab e Ab-Ag. Ele também integra uma busca local com a busca global.

HOU et al. (2004) apresentam um algoritmo de Seleção Clonal otimizado baseado no algoritmo CLONALG. Um operador de aprendizado é introduzido buscando o aprimoramento do mecanismo de aprendizado do algoritmo CLONALG e para melhorar a eficiência de detecção.

Em (CUTELLO et al., 2005), um algoritmo imunológico é proposto para solução de problemas de otimização global contínua, chamado de OPT-IA. A principal característica do algoritmo proposto é um operador de clonagem que explora a vizinhança em cada ponto do interior do espaço de busca. Um operador de hipermutação proporcional inversa é utilizado no algoritmo, onde o número de mutações é inversamente proporcional ao valor da aptidão (*fitness*). Finalmente, um operador de idade é utilizado para remover as soluções candidatas mais antigas da população atual para introdução de diversidade, evitando o mínimo local durante o processo de busca.

CUTELLO et al. (2006) propõem uma versão melhorada do algoritmo OPT-IA chamada opt-IMMALG. As principais modificações são a troca da representação binária pela representação real e a introdução de um novo operador de hipermutação inversamente proporcional.

Por fim, GONG et al. (2007) apresentam um algoritmo de seleção clonal melhorado, baseado no CLONALG, com um novo método de mutação chamado de mutação caótica parcialmente adaptativa. As principais modificações são a utilização de uma sequência lógica caótica para a geração da população inicial de anticorpos e o novo método de hipermutação.

2.2 TEORIA DOS JOGOS (*GAME THEORY*)

De acordo com Borges (1996) a Teoria dos Jogos lida com situações de conflito de interesses, onde dois ou mais agentes (ou indivíduos) disputam entre si por algum recurso limitado no ambiente.

A Teoria dos Jogos descreve problemas reais, fornecendo uma visão geral da situação. Em alguns casos, a teoria pode indicar a solução para o problema, entretanto, na maioria dos casos, apresenta a cada indivíduo a melhor forma de agir.

Anatol Rapoport (1960) em “*Fights, games and debates*” diz que um agente racional pode ser definido como um indivíduo que age racionalmente. Isso leva à consideração de todas as possíveis consequências de seus atos e, a partir disso, se realiza uma ordem de preferência entre as consequências, baseado em seus próprios atos e na ação que gerou o seu melhor resultado.

Em alguns casos, o resultado não depende unicamente da escolha feita por um único indivíduo, mas sim da ação escolhida pelos outros indivíduos, sobre os quais o primeiro indivíduo não possui qualquer controle.

É interessante notar que um Agente Humano não toma decisões de acordo com o que RAPOPORT (1960) define como racional, isto é, com a intenção de maximizar seus ganhos. Aqui se percebe a diferença entre Agentes Racionais e Agentes Humanos.

A seguir são identificados todos os elementos necessários para a compreensão do objetivo principal da Teoria dos Jogos:

- Jogo: um modelo formal, o que significa que a Teoria dos Jogos envolve descrições e análises técnicas. É importante notar que os únicos tipos de jogos tratados pela Teoria dos Jogos são os estratégicos;
- Interações: as ações individuais de cada agente afetam os outros agentes;
- Agentes: qualquer indivíduo ou grupo de indivíduos tem capacidade de tomar decisões que afetam outros agentes. O agente no contexto da Teoria dos Jogos é denominado “jogador”;
- Racionalidade: assumindo que todos os agentes são racionais, significa que eles utilizam o método mais adequado buscando a satisfação dos seus desejos;

- Comportamento Estratégico: isso significa que cada jogador leva em consideração o fato de que todos os jogadores interagem entre si. Dessa maneira, a decisão tomada irá gerar consequências aos outros jogadores.

Para exemplificar o jogo, destaca-se o Paradigma do Dilema do Prisioneiro, que pode ser definido tradicionalmente como uma situação de conflito de interesses, onde dois indivíduos são presos e colocados em celas diferentes.

Então, foi proposto a cada preso pela polícia o seguinte:

- Se um deles confessasse o crime e o outro não, o que tivesse confessado seria condenado a ficar recluso por três meses por sua cooperação e o outro indivíduo cumpriria pena de dois anos;
- Se ambos confessarem o crime, então a cooperação individual perde força e ambos ficariam presos pelo período de um ano.
- Se não, caso nenhum deles confesse o crime, eles serão presos por apenas seis meses.

A Figura 2-2 mostra a tabela de pagamentos do dilema do prisioneiro, o mais clássico jogo de duas pessoas de soma não zero e não cooperativo na Teoria dos Jogos, que combinam todos os possíveis pares de estratégias correlacionando todos os valores de pagamentos com cada um dos jogadores (RAPOPORT, 1965).

		J2	
		Cooperar (C)	Trair (D)
J1	Cooperar (C)	(R,R)	(S,T)
	Trair (D)	(T,S)	(P,P)

Figura 2-2 Tabela de pagamentos do Jogo Dilema do Prisioneiro (RAPOPORT, 1965)

Observando a Figura 2-2, pode-se ver que cada um dos jogadores tem dois tipos de comportamentos: Cooperar (*Cooperate*) e Trair (*Defeated*). Por trás disso, tem-se quatro variáveis que significam: (1) R é a recompensa (*Reward*) para cada jogador caso ambos venham a cooperar; (2) P é a punição (*Punishment*) para cada jogador caso ambos venham a trair; (3) T é a tentação (*Temptation*) de cada um dos jogadores, caso traíam sozinhos; (4) S é o pagamento do otário (*Sucker*) que coopera sozinho.

A caracterização do Dilema do Prisioneiro é feita de acordo com as seguintes relações (RAPOPORT, 1965):

- $T > R > P > S$
- $R > \frac{T+S}{2}$
- $\frac{T+S}{2} > P$

Essas especificações caracterizam o ambiente onde os indivíduos de uma população irão interagir buscando incrementar o seu ganho.

2.3 CO-EVOLUÇÃO

O termo co-evolução foi utilizado pela primeira vez por ERLICH et al. (1964), para designar adaptações recíprocas entre plantas e borboletas, levando a sua interdependência. A co-evolução ocorre quando a interação entre dois ou mais organismos ocasiona uma resposta evolutiva em cada um deles. Um bom exemplo são os fungos que co-evoluem com representantes de todas as formas de vida, de bactérias ao homem.

É importante enfatizar a atuação das forças evolutivas no processo de co-evolução, gerando variabilidade em todas as populações ou organismos envolvidos. Através de mutações, surgem novos alelos nas populações, os quais sofrem seleção e são rearranjados graças à recombinação genética.

Existem vários modelos de co-evolução computacional, sendo os principais (POTTER, 2000):

- O Modelo Co-evolutivo Tradicional, que parte do princípio “dividir para conquistar”;
- O Modelo Co-evolutivo Hierárquico, onde as populações apresentam níveis de hierarquia, sendo as de níveis mais baixos dependentes de seus superiores;
- O Modelo Co-evolutivo Genético, que foi criado recentemente como uma extensão dos métodos evolutivos tradicionais, apresentam várias populações (subpopulações) evoluindo em conjunto, trocando informações entre si para alcançar um objetivo em comum.

O trabalho abordará o modelo Co-evolutivo Genético sendo este melhor detalhado a seguir.

2.3.1 Modelo Co-evolutivo Genético

Nesse modelo, várias populações evoluem de maneira independente, trocando informações entre si. Esse modelo foi originalmente concebido para utilização em Algoritmos Genéticos (POTTER, 2000) e, por esta razão, é proposto um outro modelo inspirado neste, porém utilizando populações de algoritmos baseados na técnica do Sistema Imunológico Artificial.

O novo modelo utiliza várias populações evoluindo de maneira independente, apresentando características próprias por evoluírem em sistemas diferentes, porém trocando informações em períodos pré-definidos. Esse modelo será detalhado na seção 4.6 onde será abordado o modelo de paralelização baseado em subpopulações.

2.4 RESUMO

Nesse capítulo foram apresentados, em detalhes, os Sistemas Imunológicos Humano e Artificial, a Teoria dos Jogos e a Co-evolução, descrevendo as características de cada um. Os pontos principais são as características do Sistema Imunológico que, aplicadas computacionalmente, tornam a técnica do Sistema Imunológico Artificial uma poderosa técnica de otimização, sendo elas:

- Teoria de Seleção de Clones: Modelo de seleção dos melhores clones e reprodução assexuada. É o processo responsável por explorar o espaço de busca a procura de soluções satisfatórias, tendo como objetivo manter a diversidade na população.
- Reforço de Aprendizado e Memória Imunológica: Armazenamento das melhores células, promovendo assim um rápido processo evolutivo. É o processo responsável por manter na população as melhores soluções, ou seja, o que é aprendido nunca é descartado, sendo sempre utilizado na busca de melhores soluções.
- Hipermutação Somática: Motor de otimização através da mutação proporcional a adaptabilidade. É o processo que promove a evolução em si. Nessa etapa a população é submetida a alterações que tendem a gerar melhores soluções.

No próximo capítulo é apresentado o problema foco desta pesquisa (Problema de Alocação de Berços – PAB), sendo descrito o seu conceito, contexto e terminologia empregada.

3 PROBLEMA DE ALOCAÇÃO DE BERÇOS

Neste capítulo é apresentado o Problema de Alocação de Berços (PAB). São descritos detalhadamente os conceitos e as principais características do problema, juntamente com uma visão geral para melhor entendimento do problema.

3.1 TRANSPORTE MARÍTIMO

Segundo PEIXOTO (1977), o sistema de transportes de um país tem a função de promover a circulação em todos os recantos do território nacional, e dentre todos os meios de transportes, destacando-se o modal marítimo por ser responsável por uma grande parcela deste sistema. Sendo assim, fica clara a importância do Porto no Sistema de Transporte, pois é nele que se movimentam cargas, recebem-se e despacham-se produtos, ocorrem os serviços da alfândega e a organização do transporte. Para SILVA (1999), o porto é visto como uma porta de entrada, uma abertura para a constituição de negócios, criando um amplo campo de oportunidades industriais e comerciais.

Ao longo da história, o transporte marítimo sempre foi um dos pilares fundamentais para o crescimento econômico mundial. Os serviços de transporte marítimo são essenciais para ajudar a economia e as empresas competirem em nível mundial. A maior parte do comércio mundial, cerca de 80%, é realizado por via marítima (CCE, 2009).

O atual crescimento da economia mundial e do comércio internacional de mercadorias aumentou a procura de serviços pelo transporte marítimo e, com isso, o transporte de carga containerizada, 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. Na Tabela 3-1 é apresentada a lista de movimentação de contêineres dos principais portos do mundo, onde se percebe que, antes da crise econômica ocorrida em 2009, o volume de carga vinha aumentando ano a ano.

Tendo isso em vista, os portos devem adotar uma nova função de organizar e gerenciar fluxos contínuos, buscando se aperfeiçoar internamente, sendo capazes de fazer circular fluxos de mercadorias cada vez maiores em um tempo cada vez menor, e com custos sempre mais baixos e competitivos.

Tratando-se de um alto volume movimentado de carga, os portos têm buscado a introdução de equipamentos computadorizados e automatizados. Um exemplo é a utilização de pontes rolantes que são operadas por um único homem, substituindo centenas de estivadores, diminuindo o custo dos serviços de forma drástica e dando agilidade à operação.

TEU – Milhões						
Ranking	Porto	País	2009	2008	2007	2006
1º	Singapura	Singapura	25,866	29,918	27,932	24,792
2º	Shanghai	China	20,983	24,248	23,881	23,539
3º	Hong Kong	China	20,983	24,248	23,881	23,539
4º	Shenzhen	China	18,250	21,414	21,099	18,469
5º	Busan	Coréia do Sul	11,954	13,425	13,270	12,039
6º	Guangzhou	China	11,190	11,001	9,200	6,600
7º	Dubai	Emirados Árabes	11,124	11,827	10,653	8,923
8º	Ningbo	China	10,502	11,226	9,349	7,068
9º	Qingdao	China	10,260	10,320	9,462	7,702
10º	Roterdão	Holanda	9,743	10,784	10,791	9,655
11º	Tianjin	China	8,700	8,500	7,103	5,950
12º	Kaohsiung	Taiwan	8,581	9,677	10,257	9,775
13º	Port Klang	Malásia	7,309	7,970	7,120	6,326
14º	Antuérpia	Bélgica	7,309	8,663	8,176	7,019
15º	Hamburgo	Alemanha	7,007	9,737	9,890	8,862
16º	Los Angeles	Estados Unidos	6,748	7,850	8,355	8,470
17º	Tanjung	Malásia	6,000	5,600	5,500	4,770
18º	Long Beach	Estados Unidos	5,067	6,350	7,316	7,289
19º	Xiamen	China	4,680	5,035	4,627	4,019
20º	Bremen	Alemanha	4,578	5,529	4,912	4,450
44º	Santos	Brasil	2,506	2,675	2,532	2,456

Tabela 3-1 Lista da Movimentação de Contêineres dos principais Portos do Mundo

Fonte: AAPA - *American Association of Port Authorities*

No Brasil, a competitividade dos produtos nacionais há tempos sofre a ação da ineficiência gerencial e operacional, com custos portuários exorbitantes que inviabilizam a movimentação de cargas e representam uma perda de bilhões de dólares nas exportações todos os anos.

O Brasil é um dos países com o maior custo para a movimentação da carga, se comparado a países como Bélgica e Holanda (Tabela 3-2). O valor chega a ser quatro vezes maior, mesmo apresentando condições de atendimento portuário precário (Oliveira, 2000).

Portos	Preço (US\$/TEU)
Antuérpia	107
Roterdã	150
Santos e Rio de Janeiro	600

Tabela 3-2 Lista do preço de movimentação de um container em US\$ em alguns portos do mundo

Fonte: Oliveira (2000)

Sendo assim, fica claro que o melhor gerenciamento das atividades realizadas no porto pode oferecer a melhoria dos serviços e diminuição dos custos, aumentando a capacidade portuária e tornando o transporte marítimo mais eficiente. A ferramenta computacional proposta tem como objetivo otimizar uma parcela da operação portuária, buscando assim, oferecer um sistema capaz de realizar o gerenciamento eficiente dos recursos do porto, representados neste trabalho pelos berços de atracação.

3.2 OPERAÇÃO PORTUÁRIA

A Confederação Nacional de Transportes (CNT) define berço como sendo o “espaço destinado ao navio em um cais no qual ele pode operar com segurança” (CNT, 2006). Na Figura 3-1 pode ser visualizada uma representação dos berços de atracação em um cais portuário.

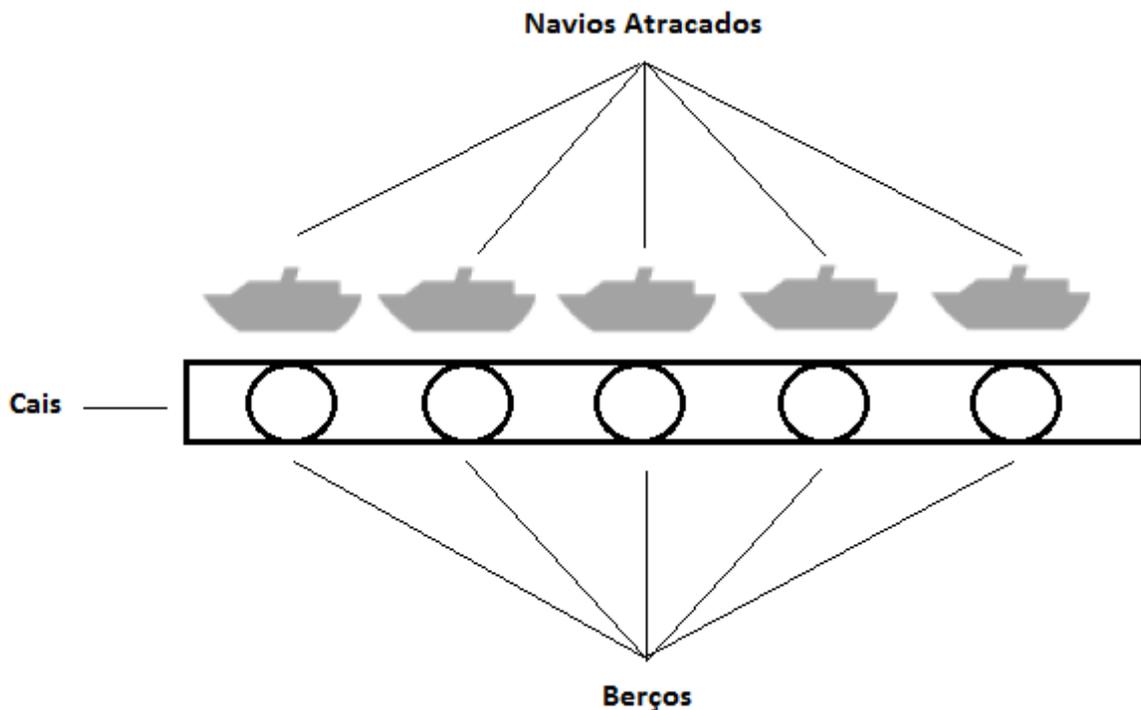


Figura 3-1 Caracterização do Berço em um Porto

De acordo com IMAI et al. (2001), em sua maioria, os berços nos grandes portos são arrendados, ou seja, alugados pelos operadores dos navios para a realização das operações de carga e descarga, a fim de alcançar maior produtividade. Enquanto isso é justificado no caso de uma manipulação de um grande volume de carga, com um grande número de navios aportando, isto pode não resultar em ganhos econômicos se estas quantidades não forem suficientes, pois parte do aumento dos custos é resultante da sobrecapitalização do porto para cargas de volumes relativamente pequenos.

Neste contexto, é interessante limitar o número de berços operantes. A alocação de berços neste sistema, isto é, a atribuição de berços aos navios que aportam para efetuar a manipulação de cargas, torna-se importante na minimização do tempo de realização desta tarefa, devido ao tempo de operação para um navio específico não ser necessariamente o mesmo para cada berço.

Conforme ASHFORD et al. (1982), a taxa de carregamento e descarregamento depende:

- a) do tipo de carga;
- b) do tipo de embarcação e tamanho (especialmente, o número de escotilhas);

- c) da disponibilidade e tamanho da equipe de estivadores;
- d) do grau de mecanização e métodos de manipulação de cargas.

Segundo ASHFORD et al. (1982), no planejamento dos portos, os responsáveis pelas operações do terminal portuário devem obter previsões confiáveis da quantidade de navios a serem recebidos e do tempo necessário para as suas operações, na ordem de antecipar a quantidade de berços requeridos. Portanto, dimensões adequadas para canais de acesso e berços devem ser estipulados para permitir segurança, ativa movimentação dos navios e atracação. Além disso, extensão suficiente para manobra também deve ser estabelecida, a fim de possibilitar as operações de carregamento e descarregamento de navios e também para abastecimento e suprimento dos mesmos.

A profundidade do porto, englobando o berço de atracação e a largura do canal de acesso, deve ser suficiente para permitir o completo carregamento de navios, de modo a navegarem seguramente em águas profundas. Obviamente, a profundidade do porto requerida depende principalmente da distância vertical entre a linha d'água e a quilha dos navios usando o porto. A largura da entrada necessária naturalmente será influenciada pelo tamanho do porto e dos navios que o utilizam. Como regra, a largura da entrada deve ser aproximadamente igual ao comprimento do maior navio que utiliza tal porto (ASHFORD et al. 1982).

Terminais que acomodam diferentes tipos de navios de carga variam amplamente quanto ao berço e ao espaço requeridos para atracação. Não existem padrões para berços e dimensionamento do porto, e os espaços vão depender também dos procedimentos de manipulação de cargas e equipamentos. O comprimento do berço deve ser igual ao comprimento do navio, mais um pequeno espaço entre os navios adjacentes e o espaço para as amarras dos navios.

A forma e extensão da área do atracadouro serão dependentes, principalmente, dos seguintes fatores:

- a) número máximo de navios a serem atendidos;
- b) tamanho dos navios;
- c) método de atracação;
- d) necessidade de mobilidade;
- e) condições topográficas do local proposto.

Uma vez definidas as características físicas do terminal portuário, deve-se ter em mente que o mesmo deverá possibilitar a realização de no mínimo três funções principais, conforme as orientações de ASHFORD et al. (1982): carregar/descarregar cargas de navios com eficiência e rapidez; prover adequado sistema, temporário e de longo prazo, de armazenagem para cargas que chegam e partem e, prover conexões, seja ferroviária ou rodoviária, para movimento do frete dentro e fora da área do porto.

Enquanto a capacidade prática do porto pode ser limitada por alguma dessas funções, a função de carregar/descarregar cargas de navios com eficiência e rapidez é geralmente o fator de controle. Portanto, a capacidade de operação será essencialmente decorrente da taxa de manipulação de cargas (peso/dia/berço ocupado) e do número e grau de utilização dos berços.

Assim, os gerentes na maioria dos portos, buscam reduzir os custos através de uma eficiente utilização dos recursos, incluindo recursos humanos, área de atracação (berços), área destinada à armazenagem da carga, guindastes para movimentação da carga e vários outros equipamentos. Dentre todos estes recursos, os berços são um dos mais importantes e, um planejamento e agendamento adequado deles, melhora a satisfação dos consumidores e aumenta a movimentação nos portos, conduzindo a altos rendimentos do porto.

3.3 DEFINIÇÃO DO PROBLEMA

O problema de alocação de berços (PAB) consiste em atribuir os navios que chegam a um determinado porto para as posições de atracação disponíveis ao longo de um cais, denominadas de berços. As principais decisões a serem tomadas neste processo envolvem a escolha de onde e quando os navios deverão atracar (CORDEAU et al. 2005).

Existem várias restrições possíveis de considerar neste problema, entre elas as relativas à profundidade, à distância máxima em relação ao local mais favorável ao longo do cais, ao tamanho do navio, entre outras. As restrições relativas ao tempo, segundo CORDEAU et al. (2005), são expressas como janelas de tempo para a conclusão do seu atendimento.

O tempo de atendimento de um navio depende do seu ponto de atracação (berço), sendo uma função da distância do berço até a área de carga e descarga no pátio ou armazéns do porto. CORDEAU e colaboradores (2005) afirmam que essa dependência afeta fortemente o desempenho das operações no porto.

O objetivo do PAB, portanto, é minimizar o tempo de serviço total dos navios. Caso os navios que chegam ao porto não apresentem a mesma importância, uma soma dos tempos de serviços dos navios, considerando uma penalização para indicar a sua devida importância, pode refletir melhor a prática do gerenciamento de alguns portos. Os pesos nesta soma ponderada podem indicar um esquema de precedência baseado no valor estimado da carga ou na quantidade de carga movimentada. Em algumas variantes do problema, podem ser incluídas outras condições de penalidade na função objetivo, como prioridade de atendimento, tipo de carga movimentada, entre outras.

O PAB pode ser modelado como um problema discreto se o cais for visto como um conjunto finito de berços. Neste caso, os berços podem ser descritos como segmentos de comprimento fixo, ou se a dimensão de espaço for ignorada, como pontos. Já os modelos contínuos consideram que os navios podem atracar em qualquer posição ao longo do cais (BUHRKAL et al. 2009).

O PAB é um problema da classe NP-Difícil dentro da área de Pesquisa Operacional (GAREY et al., 1979), tendo sido relacionado com o problema de partição de conjuntos (LIM, 1998), com o problema de sequenciamento de uma única máquina com datas de entrega (HANSEN et al., 2003), e com o problema de corte bidimensional (IMAI et al., 2005), todos eles sabidamente classificados como NP-difícil (BIERWIRTH et al., 2010). Essa classe de problemas compreende todos aqueles que não possuem solução determinística polinomial conhecida e que são polinomialmente equivalentes em si (ARORA et al. 2009). Problemas da classe NP-Difícil não necessariamente possuem uma forma eficiente de verificar suas soluções, ou seja, ao contrário dos problemas da classe NP, não necessariamente existem métodos polinomiais de se verificar se uma resposta para um problema de decisão dessa classe é correta. GULDOGAN et al (2012) define o PAB como um problema NP-Difícil, relatando sua complexidade e importância para comunidade científica atual.

3.4 TRABALHOS RELACIONADOS

O primeiro modelo de otimização para o PAB foi proposto por THURMAN (1989). BROWN et al. (1994) aprimoraram este modelo e minimizaram os conflitos entre os navios. Os autores propuseram ainda um planejamento para o uso de berços para submarinos em BROWN et al. (1997).

Usando um método baseado na relaxação lagrangeana, IMAI et al. (2001) abordaram o PAB em sua forma dinâmica e contínua, considerando as dimensões dos berços e navios. IMAI et al. (2003) evoluíram sua abordagem inicial e consideraram prioridades no atendimento dos navios. Para gerar as soluções, os autores ainda propuseram o emprego de um Algoritmo Genético.

Alguns trabalhos apresentam o PAB estendido com prioridade de serviço. Prioridade de serviço significa atribuir uma importância a cada navio. IMAI et al. (2003) adicionaram um termo à função objetivo para incluir diferentes prioridades. Isso também foi feito por IMAI et al. (2001), que empregaram um algoritmo genético. DAI et al. (2004) solucionaram o problema como um problema de empacotamento “retangular”. Os autores implementaram um algoritmo de busca de vizinhanças usando *Simulated Annealing*.

Baseados na Busca Tabu, CORDEAU et al. (2005) propuseram duas heurísticas e duas formulações matemáticas para o PAB. CHEONG et al. (2008) apresentaram uma aplicação baseada em um Algoritmo Evolucionário Multiobjetivo. Já GIALLOMBARDO et al. (2010) utilizaram um modelo de programação quadrática, um de programação linear, uma Busca Tabu e uma técnica de programação matemática, usando instâncias reais do problema.

Mauri et al. (2008a) aplicaram o método *Simulated Annealing* para o caso discreto do PAB e Mauri et al. (2008b) propuseram uma metaheurística híbrida que utiliza o Algoritmo de Treinamento Populacional em conjunto com um modelo de Programação Linear por meio da técnica de Geração de Colunas. Meisel and Bierwirth (2009) usaram a metaheurística “squeaky wheel” em conjunto com a Busca Tabu para solucionar o problema. Oliveira et al. (2010) utilizaram o método *Clustering Search* associado ao *Simulated Annealing* para resolver o PAB.

Por fim, uma recente extensão do PAB é a sua integração com o problema de escalonamento de guindastes (PEG). Contribuições nesta variação do problema incluem os trabalhos de PARK et al. (2003), IMAI et al. (2005), LIANG et al. (2009) e BIERWIRTH et al. (2009). Todos propuseram o uso de metaheurísticas para solução do problema.

3.5 RESUMO

Neste capítulo foi apresentado o PAB, problema abordado neste trabalho e cuja solução será obtida através do emprego dos algoritmos propostos. Os principais pontos abordados foram:

- Contextualização do Problema: O problema é de extrema importância no atual cenário mundial, sendo assim, o seu estudo de grande interesse da comunidade científica;
- Caracterização do Problema: definiu-se o problema de alocação de berços (PAB) como a melhor forma de atribuir os navios que chegam a um determinado porto para as posições de atracação disponíveis ao longo de um cais, denominadas de berços.

Agora, com a caracterização do Sistema Imunológico Artificial e do PAB, será apresentado, no próximo capítulo, o modelo proposto dos algoritmos voltados para a solução do PAB.

4 METODOLOGIA PROPOSTA

Este capítulo apresenta o modelo matemático utilizado na descrição do PAB, contemplando as restrições e função objetivo, os algoritmos AISO, PCAISO e PCAISO-GT desenvolvidos para a solução do problema, descrevendo detalhadamente os conceitos utilizados, os operadores empregados, o modelo de co-evolução, o conceito de Teoria dos Jogos e sua aplicação, e as estratégias de paralelização utilizadas. É apresentado também o pseudocódigo das técnicas desenvolvidas junto com uma descrição detalhada do seu funcionamento por meio de fluxogramas.

4.1 MODELO MATEMÁTICO DO PAB

O PAB, como descrito no capítulo anterior, consiste em selecionar os navios que chegam ao porto e atribuí-los a berços ao longo do cais. As principais decisões a serem tomadas neste processo envolvem a escolha de onde e quando os navios deverão atracar (CORDEAU et al. 2005). O PAB tem como objetivo minimizar o tempo de serviço de cada navio no porto. Esse tempo é dado pela soma do tempo de espera (tempo que o navio aguarda para ser atendido) e do tempo de atendimento.

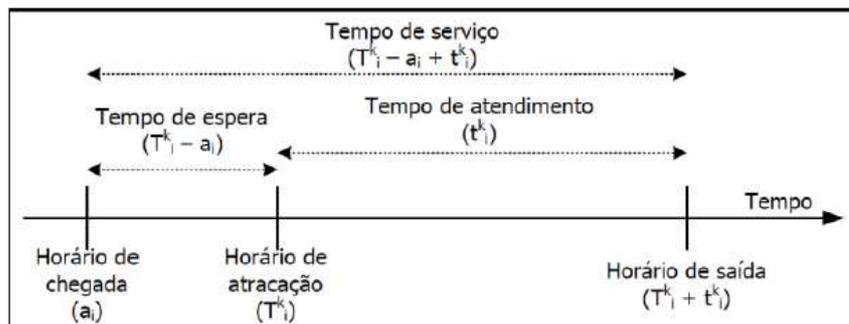


Figura 4-1 Representação das variáveis de tempo (Mauri et al. 2008a)

Considerando então os tempos definidos na Figura 4-1, pode-se definir matematicamente o PAB por meio da seguinte notação:

- N : conjunto de navios, $n = |N|$;
- M : conjunto de berços, $m = |M|$;
- t_i^k : duração do atendimento do navio i no berço k ;
- a_i : horário de chegada do navio i ;
- s^k : horário de abertura do berço k ;
- e^k : horário de fechamento do berço k ;
- b_i : horário de término da janela de tempo para o navio i ;
- v_i : valor do tempo de serviço do navio i ;
- $x_{ij}^k \in \{0,1\} \forall k \in M, \forall (i, j) \in A^k, x_{ij}^k = 1$ se o navio j é atendido pelo berço k após o navio i ;
- $T_i^k \forall k \in M, i \in N$ é o horário que o navio i atracou no berço k ;
- $T_{o(k)}^k \forall k \in M$ é o horário em que o primeiro navio atracou no berço k ;
- $T_{d(k)}^k \forall k \in M$ é o horário em que o último navio saiu do berço k ;
- $M_{ij}^k = \max\{b_i + t_i^k - a_i\}, \forall k \in M, \forall (i, j) \in N$.

Segundo CORDEAU *et al.* (2005), o PAB pode ser descrito matematicamente pelas equações (1)-(11), a seguir. A função objetivo (1) minimiza a soma do tempo de serviço; a restrição (2) garante que cada navio seja atendido por apenas um berço; as restrições (3) e (4) garantem, respectivamente, que um navio será o primeiro a ser atendido em cada berço e outro será o último; a restrição (5) garante a “conservação do fluxo”, ou seja, o atendimento para os demais navios; a restrição (6) faz o cálculo do horário de atracação dos navios, sendo considerados apenas os navios válidos para cada berço, ou seja, alguns navios não podem ser atendidos em determinados berços devido a restrições técnicas; as restrições (7) e (8) garantem, respectivamente, que o horário de atracação seja após a chegada do navio e que o horário do término do atendimento do navio seja anterior ao horário-limite do navio (janela de tempo); as restrições (9) e (10) garantem a não violação das janelas de tempo nos berços; por fim, a restrição (11) garante que as variáveis de decisão sejam binárias. Mais detalhes sobre esse modelo são apresentados em CORDEAU *et al.* (2005).

Minimizar:

$$\sum_{i \in N} \sum_{k \in M} v_i \left[T_i^k - a_i + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{ij}^k \right] \quad (1)$$

Sujeito a:

$$\sum_{k \in M} \sum_{j \in N \cup \{d(k)\}} x_{ij}^k = 1 \quad \forall i \in N \quad (2)$$

$$\sum_{j \in N \cup \{d(k)\}} x_{o(k),j}^k = 1 \quad \forall k \in M \quad (3)$$

$$\sum_{i \in N \cup \{o(k)\}} x_{i,d(k)}^k = 1 \quad \forall k \in M \quad (4)$$

$$\sum_{j \in N \cup \{d(k)\}} x_{i,j}^k - \sum_{j \in N \cup \{o(k)\}} x_{j,i}^k = 0 \quad \forall k \in M, \forall i \in N \quad (5)$$

$$T_i^k + t_i^k - T_j^k \leq (1 - x_{i,j}^k) M_{i,j}^k \quad \forall k \in M, \forall (i, j) \in A^k \quad (6)$$

$$T_i^k \geq a_i \quad \forall k \in M, \forall i \in N \quad (7)$$

$$T_i^k + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{j,i}^k \leq b_i \quad \forall k \in M, \forall i \in N \quad (8)$$

$$T_{o(k)}^k \geq s^k \quad \forall k \in M \quad (9)$$

$$T_{d(k)}^k \leq e^k \quad \forall k \in M \quad (10)$$

$$x_{i,j}^k \in \{0,1\} \quad \forall k \in M, \forall (i, j) \in A^k \quad (11)$$

MAURI *et al.* (2008a) propõem a relaxação das restrições (7), (8), (9) e (10), de tal forma que as restrições (7) e (8) são inseridas na função objetivo original sendo combinadas conforme o termo (13), e as restrições (9) e (10) são inseridas na função objetivo conforme o termo (14). Além disso, coeficientes de penalização ($\omega = [\omega_o, \omega_i, \omega_2]$) são utilizados para ponderar cada termo da função objetivo. Sendo assim, o tempo de serviço, juntamente com seu custo, é representado no termo (12). O termo (13) minimiza as violações nas janelas de tempo dos navios e o termo (14) minimiza as violações nas janelas de tempo dos berços.

Dessa forma, tem-se a seguinte formulação:

Minimizar:

$$\omega_0 \sum_{i \in N} \sum_{k \in M} v_i \left(T_i^k - a_i + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{ij}^k \right) + \quad (12)$$

$$\omega_1 \sum_{i \in N} \sum_{k \in M} \left(\max(0, a_i - T_i^k) + \left(\max \left(0, T_i^k + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{ij}^k - b_i \right) \right) \right) + \quad (13)$$

$$\omega_2 \sum_{k \in M} \left(\max(0, s^k - T_{o(k)}^k) + \max(0, T_{d(k)}^k + e^k) \right) \quad (14)$$

Sujeito a:

$$\sum_{k \in M} \sum_{j \in N \cup \{d(k)\}} x_{ij}^k = 1 \quad \forall i \in N \quad (15)$$

$$\sum_{j \in N \cup \{d(k)\}} x_{o(k)j}^k = 1 \quad \forall k \in M \quad (16)$$

$$\sum_{i \in N \cup \{o(k)\}} x_{i,d(k)}^k = 1 \quad \forall k \in M \quad (17)$$

$$\sum_{j \in N \cup \{d(k)\}} x_{i,j}^k - \sum_{j \in N \cup \{o(k)\}} x_{j,i}^k = 0 \quad \forall k \in M, \forall i \in N \quad (18)$$

$$T_i^k + t_i^k - T_j^k \leq (1 - x_{i,j}^k) M_{i,j}^k \quad \forall k \in M, \forall (i, j) \in A^k \quad (19)$$

$$x_{i,j}^k \in \{0,1\} \quad \forall k \in M, \forall (i, j) \in A^k \quad (20)$$

Deve-se destacar que esse modelo (Equações (12) a (20)) pode resultar em soluções inviáveis para o problema. Porém, as soluções inviáveis são eliminadas durante a execução do algoritmo, por meio da penalização imposta.

4.2 ALGORITMO AISO (*Artificial Immune System Optimization*)

O algoritmo AISO (FERRY et al. 2006) foi desenvolvido baseado no modelo e nos princípios do Sistema Imunológico Humano, sendo voltado especificamente para otimização e inspirado no algoritmo *Opt-AiNet* (*Optimization Artificial Immune Network*) proposto por DE CASTRO et al. (2002), o qual é extensão do algoritmo *AiNet* (*Artificial Immune Network*) voltado para otimização. O AISO consiste em uma população de células (anticorpos) que são representadas por um vetor com valores discretos ou contínuos, dependendo do problema tratado.

Cada célula é uma solução em potencial para o problema a ser resolvido. A população passa por um processo de clonagem, mutação e seleção. O algoritmo utiliza uma memória de células onde são armazenadas as melhores células (soluções) no momento. A estrutura da célula é mostrada na Figura 4-2.

ID	Valor	<i>Fitness</i>
----	-------	----------------

Figura 4-2 Caracterização da célula AISO

Os campos são definidos como:

- ID: identificador único da célula;
- Valor: armazena a solução a qual a célula representa;
- *Fitness*: armazena o valor da solução submetida à função objetivo, ou seja, representa a qualidade da solução.

As principais características deste algoritmo são:

- 1) Determina a localização de múltiplos pontos ótimos;
- 2) Capacidade de manutenção de vários pontos ótimos;
- 3) Demonstra a exploração do espaço de busca.

As principais terminologias utilizadas são:

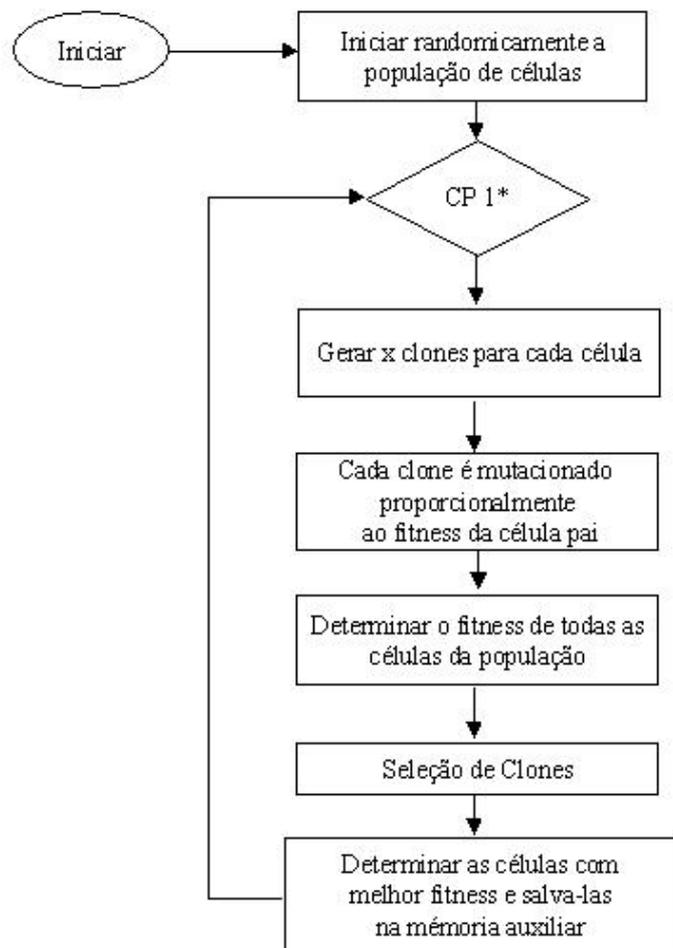
- Células: indivíduos da população. São utilizados valores adequados para a representação de uma solução;
- *Fitness*: representa a adaptabilidade de uma determinada célula em relação à função de avaliação;
- Clone: células que são cópias idênticas de seus pais;
- Clone “Mutacionado”: clone que passou pelo processo de hipermutação somática.

A estrutura do algoritmo:

1. Iniciar a população de células iniciais;
2. Enquanto o critério de parada não for atingido faça:
 - 2.1. Determinar o *fitness* de cada célula com a função avaliação;
 - 2.2. Gerar número de clones proporcional à adaptabilidade da célula;
 - 2.3. Cada clone sofre hipermutação somática com taxa proporcional ao *fitness* das células pais;

- 2.4. Determinar o *fitness* de todas as células da rede;
- 2.5. Para cada grupo de clones, selecionar o melhor Fitness e remover os outros;
- 2.6. Determinar as células com maior *fitness* e gravá-las na memória;
- 2.7. Introduzir uma porcentagem P de células geradas randomicamente.

A Figura 4-3 representa o fluxograma de funcionamento do algoritmo AISO:



CP (Critério de Parada)
 *CP 1: Número de Épocas

Figura 4-3 Fluxograma do Algoritmo AISO

4.3 AISO APLICADO AO PAB

Com o modelo matemático e algoritmo definidos, agora será apresentada a visão do algoritmo aplicado ao problema.

4.3.1 Codificação da Solução

Uma solução, isto é, a célula no caso do AISO, é representada por uma estrutura do tipo matricial, onde cada linha representa um berço, e as colunas a sequência de atendimento dos navios. Os elementos dessa matriz indicam o navio que deverá ser atendido.

Para simplificar a implementação, cada linha da matriz pode apresentar um número diferente de colunas (número de navios atendidos pelo respectivo berço), e todos os navios são distribuídos nessa matriz. A Figura 4-4 apresenta a representação de uma possível solução para um problema hipotético com três berços (linhas) e nove navios.

1	9	5	
8	2	6	4
7	3		

Figura 4-4 Exemplo de uma solução para o problema de 3 berços e 9 navios

A Figura 4-5 é a representação de uma célula do AISO para a possível solução apresentada anteriormente, onde o valor “1234” indica o identificador da célula e o valor “23122” representa o grau de adaptabilidade da célula, isto é o *fitness*.

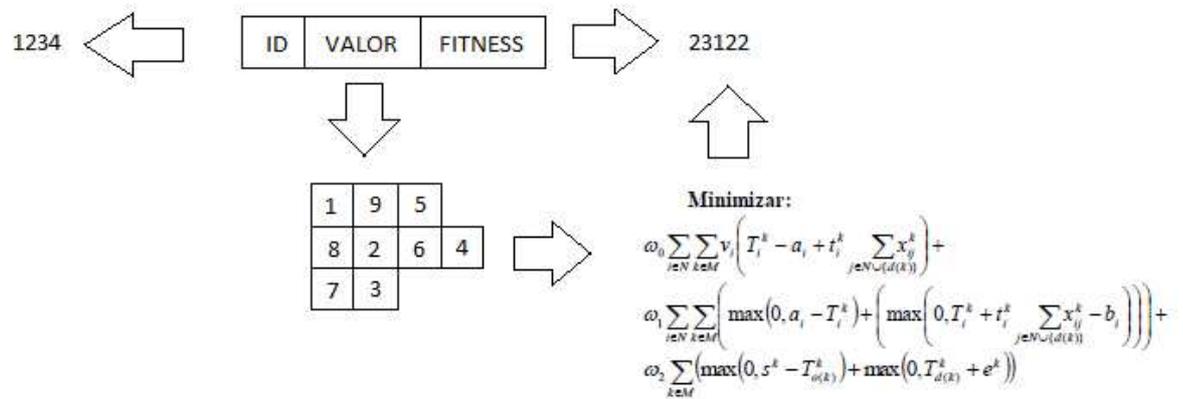


Figura 4-5 Representação da Célula para a solução da figura 4-4

O operador de mutação é uma das partes mais importantes do algoritmo, pois sem ele o algoritmo não evolui na busca por soluções melhores. A mutação tem o objetivo de prevenir que todas as soluções da população caiam em um ponto ótimo local. A forma de aplicação da operação de mutação muda de acordo com as configurações e estrutura da solução.

Toda a população de células passa pelo processo de mutação, sendo que no caso dos clones, a mutação é realizada de acordo com a adaptabilidade da célula pai, caso contrário, a célula sofre mutação de acordo com a sua adaptabilidade. A taxa de mutação é proporcional à aptidão, ou seja, quanto maior a aptidão, menos a célula é modificada pela operação de mutação e, quanto pior a aptidão, mais a célula é modificada por esta operação. A mutação é realizada da seguinte forma:

- 1) Define-se a quantidade de faixas de mutação por variável, ou seja, se houver 30 variáveis, por exemplo, serão criadas faixas de tamanho seis caso a quantidade de faixas seja definida como cinco;
- 2) Para cada faixa são definidos valores para mutação de forma crescente, ou seja, na primeira faixa a taxa de mutação será de um (troca de duas variáveis), na segunda faixa de dois (troca de quatro variáveis) e assim sucessivamente;

- 3) Os valores de comparação são definidos de acordo com o melhor e o pior *fitness* da população, gerando faixas de acordo com a divisão da diferença de valores pela quantidade de faixas;

Um exemplo simples de mutação, utilizando a possível solução apresentada na Figura 4-4, é exemplificada na Figura 4-6:

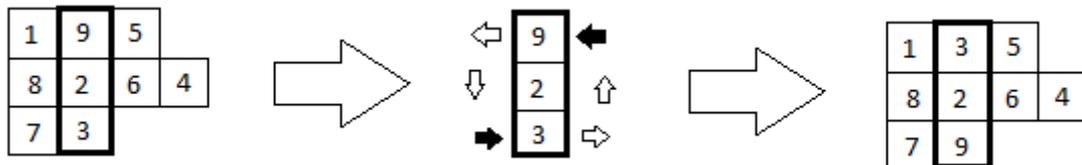


Figura 4-6 Exemplo de mutação na solução AISO

Na figura, é apresentada a mutação de uma possível solução boa, nesse caso apenas uma troca foi realizada. O processo de mutação empregado neste trabalho segue o formato identificado na Figura 4-6, sendo a única diferença a quantidade de trocas realizadas de acordo com a qualidade da célula e da quantidade de faixas de mutação definidas.

4.4 TEORIA DOS JOGOS APLICADA AO AISO

Como forma de possibilitar um aumento do controle da variabilidade da população de células, é proposta a inclusão, no algoritmo AISO, de uma fase de interação social através da Teoria dos Jogos. A nova etapa denominada “interação social”, que foi originalmente proposta por TEIXEIRA (2005), ao utilizar a Teoria dos Jogos como forma de caracterizar fenotipicamente os indivíduos de uma população, que no caso são as células.

Assim, é proposto o uso especificamente do Dilema do Prisioneiro, que possibilita a definição de um ambiente de interação social através da formalização de situações de conflito de interesses, onde os indivíduos da população disputam entre si por recursos nele disponíveis.

A alteração da estrutura da célula apresentada na Figura 4-2 se faz necessária, tendo em vista a necessidade de armazenamento do valor que caracterizará o seu comportamento. A nova estrutura é representada pela Figura 4-7.

ID	Valor	<i>Fitness</i>	<i>Comportamento</i>	<i>Ganho</i>
----	-------	----------------	----------------------	--------------

Figura 4-7 Caracterização da célula do AISO-GT

A nova estrutura do algoritmo é apresentada a seguir, sendo destacadas as modificações realizadas:

1. Iniciar a população de células iniciais;
2. Enquanto o critério de parada não for atingido faça:
 - 2.1. Determinar o *fitness* de cada célula com a função avaliação;
 - 2.2. Gerar número de clones proporcional a adaptabilidade da célula;
 - 2.3. Cada clone sofre hipermutação somática com taxa proporcional ao *fitness* das células pais;
 - 2.4. Determinar o *fitness* de todas as células da rede;
 - 2.5. Para cada grupo de clones repetir até D disputas;
 - 2.5.1. Selecionar duas células aleatoriamente
 - 2.5.2. Obter o comportamento de cada uma das células;
 - 2.5.3. Alterar o *fitness* das células conforme o comportamento adotado por cada uma e a tabela de pagamentos do jogo;
 - 2.6. Para cada grupo de clones, selecionar o melhor *fitness* e remover os outros;
 - 2.7. Determinar as células com maior *fitness* e gravá-las na memória;
 - 2.8. Introduzir uma porcentagem P de células geradas randomicamente.

A Figura 4-8 representa o funcionamento do algoritmo modificado em função da inserção da etapa baseada na Teoria dos Jogos:

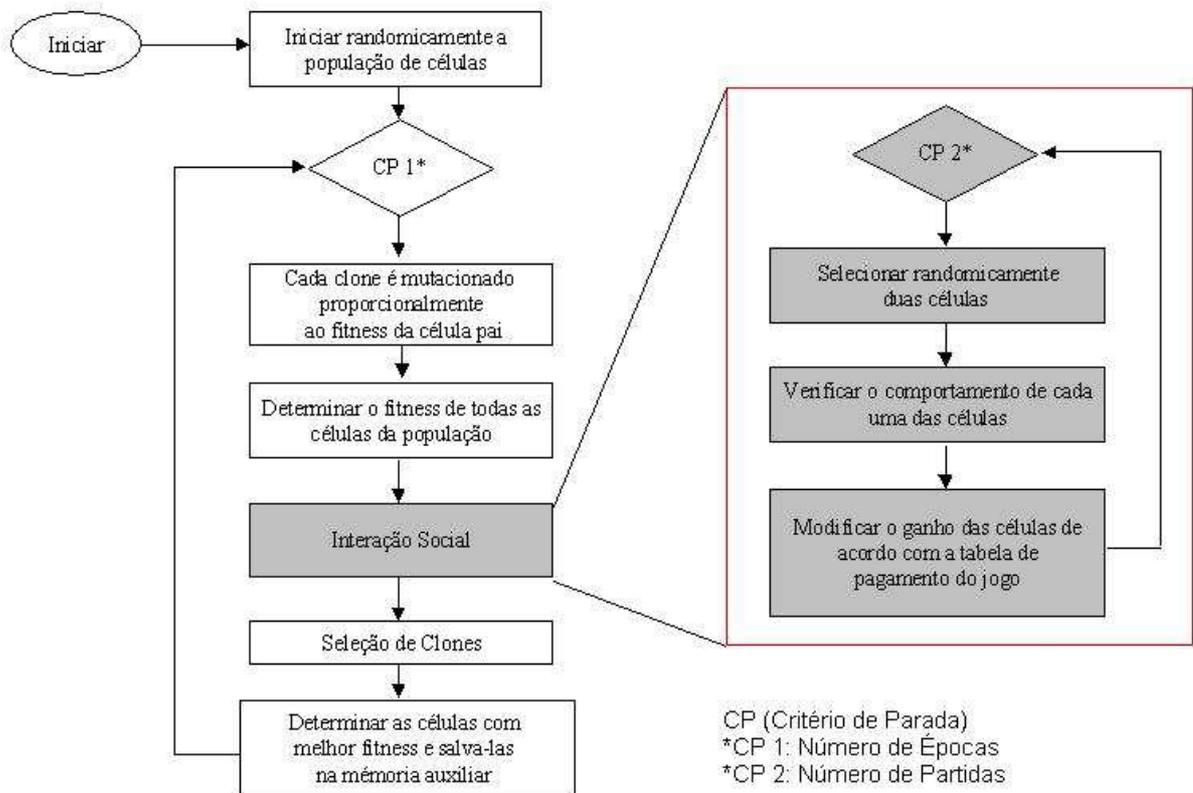


Figura 4-8 Fluxograma do Algoritmo AISO-GT

4.5 ARQUITETURA PARALELA

A arquitetura básica para os computadores modernos foi concebida nos anos 50 pelo matemático John Von Neumann, denominada de Arquitetura de von Neumann, que ainda serve de base para os computadores atuais (DALE et al. 2010). Sua arquitetura era formada basicamente por cinco partes fundamentais: unidade de controle, unidade aritmética, memória e mecanismos de entrada/saída.

Tradicionalmente, os computadores são vistos como máquinas sequenciais que executam instruções específicas contidas em algoritmos, de forma sequencial, uma de cada vez (STALLINGS, 2009). Cada instrução executa uma sequência de operações. Embora a velocidade com que computadores executam essas instruções tenha sido aumentada constantemente nos últimos anos, existem barreiras físicas que podem impedir uma evolução

ainda maior, como a tecnologia de concepção, limitações e conceito de recursos, o aquecimento do circuito e a organização arquitetural, entre outras (TANENBAUM, 2007).

Em meio a essas preocupações, tem-se buscado formas de aumentar o desempenho dos computadores sem estacionar nessas barreiras. Uma solução é organizar os computadores de forma que, múltiplas unidades de processamento possam cooperar para que o trabalho seja feito em paralelo. Uma solução é utilizar um aglomerado de computadores interligados por meio de uma rede de interconexão, os quais funcionam de forma cooperativa, para que o trabalho seja feito em paralelo, formando assim um *cluster* de computadores.

A possibilidade de expandir o poder computacional utilizando um aglomerado de computadores fez com que, nos últimos anos, o uso de *clusters* de computadores aumentasse, sendo utilizados para fins científicos e comerciais. Estudos relacionados a arquiteturas paralelas e *clusters* possibilitam o desenvolvimento de novas tecnologias e técnicas de processamento paralelo.

4.5.1 Paralelização de AIS

Para aumentar o desempenho dos algoritmos de busca na procura por soluções, é interessante o uso de paralelismo. Uma alternativa bastante trivial de paralelização de um algoritmo de busca populacional é a execução deste em vários processadores (ou processos) de forma que, quando um deles encontrar o resultado esperado, todos são interrompidos. Executar várias instâncias ao mesmo tempo, mesmo sem comunicação, é interessante, pois multiplica a chance de ter uma evolução positivamente fora da média, e reduz a probabilidade de uma má evolução.

Estratégias mais elaboradas, propostas para o AIS, são inspiradas nos modelos paralelos tradicionais dos Algoritmos Genéticos (AG) (CANTU-PAZ, 2000), são eles:

- O modelo AIS Mestre-Escravo;
- O modelo AIS utilizando Subpopulações;

O modelo Mestre-Escravo consiste em executar paralelamente, em cada processo escravo, as operações que dependem apenas de cada solução, como a mutação e a avaliação de aptidão, deixando para o processo mestre a execução das tarefas mais simples. Como a comunicação é efetuada a cada nova geração, este modelo só é interessante em arquiteturas de memória compartilhada ou quando o tempo de mutação e avaliação é significativamente

maior que o tempo de comunicação. Uma segunda opção é o emprego de um modelo de subpopulações, que será abordado no próximo item.

4.5.2 Modelo de Paralelização em Subpopulações

Considerando que problemas complexos exigem uma população grande de soluções quando empregado o AISO, o que causa um aumento no custo computacional da execução do algoritmo, torna-se interessante dividir esta população em populações menores. A ideia do modelo de subpopulações é dividir uma população grande em populações menores que evoluirão separadamente, cada uma em uma ilha (CANTU-PAZ, 2000). O ganho pela distribuição se dá inicialmente no tempo de processamento, em virtude da divisão da população em subpopulações e, posteriormente, na migração de indivíduos de uma subpopulação para as outras, através da introdução de nova informação, auxiliando a diversificação da subpopulação de destino.

O modelo de subpopulações possui duas abordagens: o modelo *Stepping Stone* (Figura 4-9) quando a migração é restrita às subpopulações vizinhas (forma de anel) e o modelo Ilha (Figura 4-10) onde a migração é realizada entre todas as subpopulações (CANTU-PAZ, 2000).

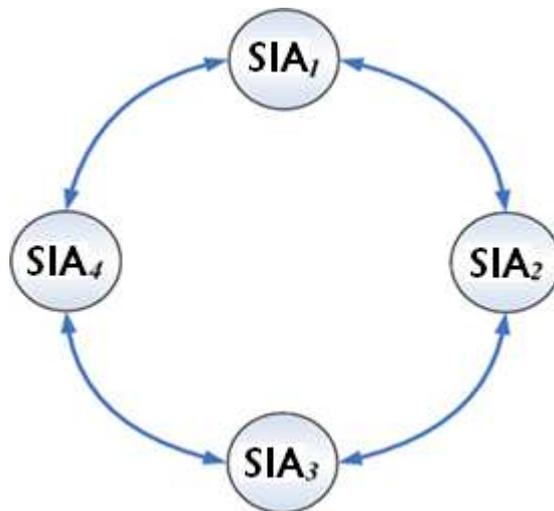


Figura 4-9 Representação do Modelo *Stepping Stone*

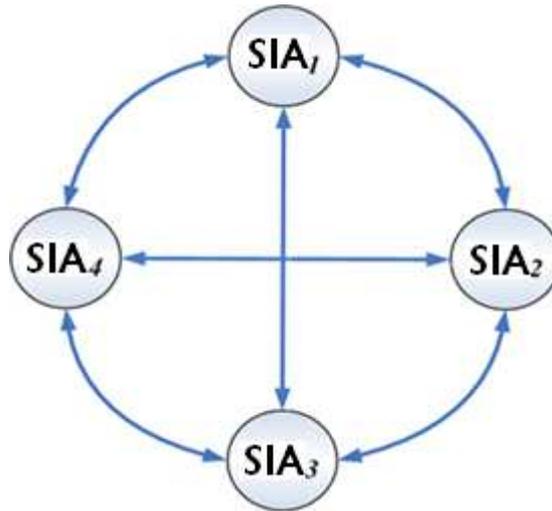


Figura 4-10 Representação do Modelo Ilha

Um problema observado quando se considera uma população única é que uma solução, inicialmente muito boa, pode propagar sua informação por toda a população, eliminando a diversidade e reduzindo a cobertura do campo de possibilidades, isto é, o espaço de soluções. O modelo de ilha, por ter populações separadas que evoluem isoladamente por várias gerações, tem esse efeito reduzido (DABROWSKI, 2008).

No entanto, o problema da uniformização das soluções também pode ocorrer no modelo de ilha, principalmente se a velocidade de evolução das diversas ilhas for muito diferente. Neste caso, uma ilha de execução veloz, que encontre logo um ótimo local no espaço de soluções, transmitirá suas soluções às demais ilhas que, ainda não tendo evoluído, propagarão as características da solução ótima imigrante dentro de suas populações.

A migração é o principal operador do modelo de ilha (CANTU-PAZ, 2000), consistindo na tarefa de transmitir soluções às demais ilhas e receber as soluções enviadas por elas. As principais características a serem decididas acerca da migração referem-se a:

- Quantidade de soluções a serem transmitidas entre as populações;
- Como as soluções serão selecionadas;
- Definição sobre o futuro das soluções migradas, podendo ser excluídas da população de origem ou sendo mantidas e fazendo parte das próximas gerações;
- Definição do momento para realização da troca de informações;
- Definição da forma como as trocas de informações serão realizadas.

Em geral, a transmissão de soluções é feita após um número determinado de iterações, sendo este número escolhido pelo projetista do algoritmo, assim como a quantidade de soluções a serem enviadas. Também é comum a escolha das melhores soluções para a transmissão, no intuito de compartilhar a evolução da ilha. Mesmo parecendo incorreto à primeira vista, é importante para manter a diversidade, que as soluções emigrantes sejam removidas da população em que estavam. Vale lembrar que, mesmo sendo retiradas, uma vez que tenham sido geradas naquele ambiente, estas podem ser geradas novamente.

Também é necessário atentar para que uma solução imigrante, recém chegada, não seja imediatamente transmitida para outra ilha, pois, desta forma, sua informação genética não seria absorvida pela população. Uma forma de resolver este problema é garantir que as soluções imigrantes só sejam incorporadas à população local após a transmissão dos emigrantes, de forma que passem todo o ciclo de espera evoluindo na ilha. O inconveniente desta alternativa é que se perde o tempo em que as soluções ficam esperando, atrasando a evolução.

4.5.3 Técnicas de Paralelização

Para a implementação do algoritmo paralelo será utilizado o modelo *multithreading*, que consiste na utilização de *threads* para a paralelização do algoritmo. O uso de *Threads* é uma forma de um processo dividir-se em duas ou mais tarefas que podem ser executadas simultaneamente (SILBERSCHATZ, 2001). Uma *thread* pode ser vista como um subprocesso de um processo, que permite compartilhar a sua área de dados com o programa ou outras *threads*.

O modelo *Multithreading* oferece simplicidade de programação e, além disso, apresenta outros benefícios como:

- Maior desempenho em ambientes multiprocessados;
- Responsividade em interfaces gráficas.

Todas as *threads* de um processo trabalham com o mesmo espaço de endereçamento, que é a memória lógica do processo que as iniciou. Isto é, quando se tem um conjunto de *threads* dentro de um processo, todas as *threads* executam o código do processo e compartilham as suas variáveis.

O algoritmo desenvolvido nesta dissertação será totalmente paralelizado, onde várias *threads* executarão o mesmo processo sobre os mesmos dados.

Essa situação pode gerar problemas se o controle das variáveis não for bem desenvolvido e, para realizar esse controle, é necessária a utilização de semáforos.

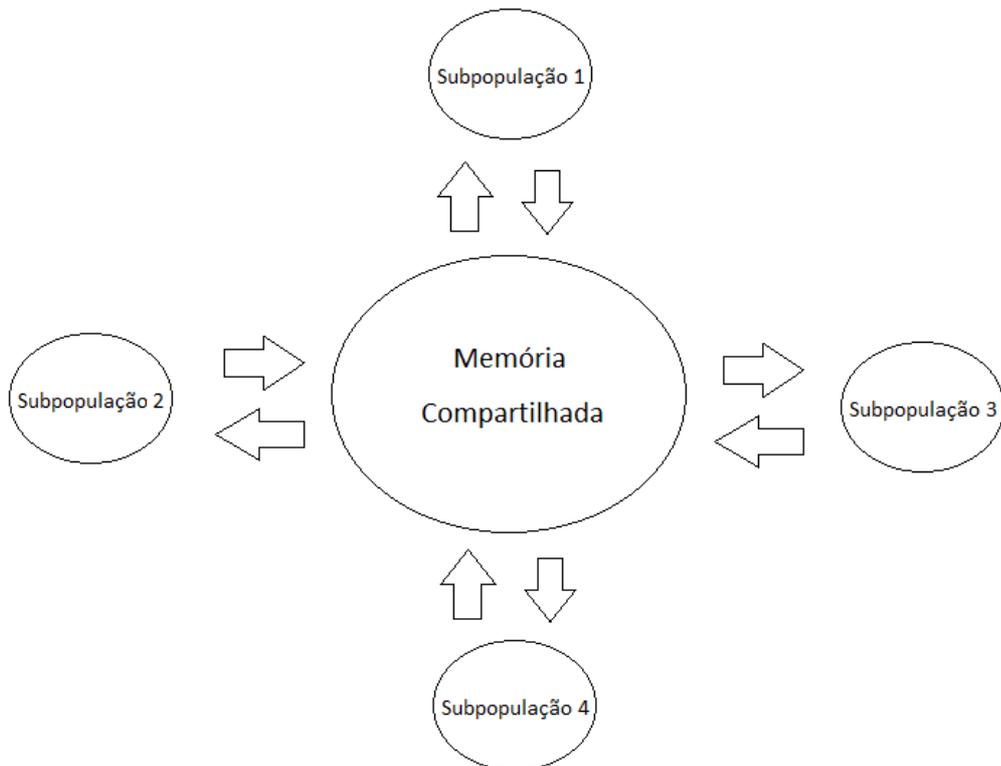


Figura 4-11 Simulação de acesso à memória compartilhada

O semáforo realiza o controle de acesso a determinada região da memória, evitando-se assim, problemas de execução na aplicação, conforme Figura 4-12.

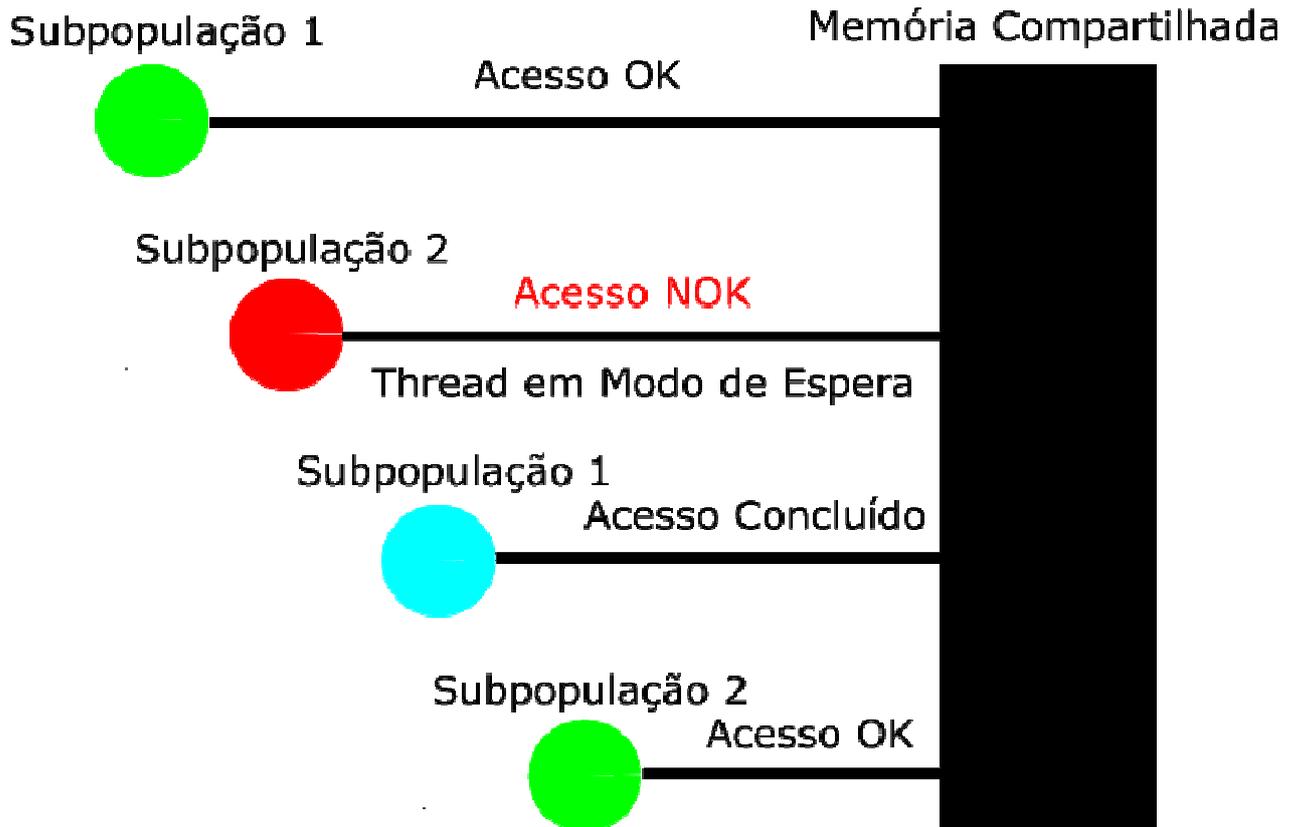


Figura 4-12 Simulação de acesso à memória compartilhada com semáforo

4.6 PROPOSTA DA NOVA METAHEURÍSTICA DE BUSCA

Esta seção apresenta a metaheurística proposta PCAISO-GT (*Parallel Coevolutionary Artificial Immune System Optimization with Game Theory*), descrevendo os ajustes realizados no algoritmo AISO para a inclusão dos conceitos de co-evolução e paralelismo no algoritmo. É apresentado também o pseudocódigo do algoritmo junto com uma descrição detalhada do seu funcionamento por meio de fluxogramas.

4.6.1 Proposta do Algoritmo PCAISO-GT

O algoritmo PCAISO-GT será desenvolvido baseado no modelo e nos princípios do Sistema Imunológico Humano, sendo voltado especificamente para otimização e inspirado no algoritmo AISO-GT proposto por FERRY et al. (2006). O PCAISO-GT possui as mesmas características do algoritmo original, sendo incluídos conceitos de paralelismo agregados a conceitos de co-evolução.

Assim, é proposta a inclusão do modelo co-evolutivo genético e do modelo de paralelização em ilha que possibilitam a divisão da população e evolução através da troca de informações entre as várias subpopulações, criando assim o novo algoritmo.

A alteração da estrutura da célula se faz necessária tendo em vista a necessidade de armazenamento de novas informações. A nova estrutura é representada pela Figura 4-13.

ID	Valor	<i>Fitness</i>	<i>Comportamento</i>	<i>Ganho</i>	<i>ID Origem</i>	<i>Contador</i>
----	-------	----------------	----------------------	--------------	------------------	-----------------

Figura 4-13 Caracterização da Célula do PCAISO-GT

Os campos Id, Valor, *Fitness*, *Comportamento* e *Ganho* são mantidos segundo a estrutura do algoritmo AISO-GT (Figura 4-7). Os novos campos representam:

- Id Origem: armazena os identificadores das populações por onde a célula passou, desde a sua origem até o final do processamento;
- Contador: Indica o tempo que a célula se encontra na população atual. O campo é necessário para realizar o controle de migração do processo, evitando assim a perda de diversidade das populações.

A nova estrutura do algoritmo é apresentada a seguir, sendo destacadas as modificações realizadas:

1. Iniciar as Npop Populações de Células;
2. Para cada População de Célula faça:
 - 2.1. Iniciar a população de células iniciais;
 - 2.2. Enquanto o critério de parada não for atendido faça:
 - 2.2.1. Determinar o *fitness* de cada célula com a função de avaliação;
 - 2.2.2. Gerar X clones para cada célula;
 - 2.2.3. Cada clone sofre hipermutação somática com taxa proporcional ao *fitness* da célula pai;
 - 2.2.4. Determinar o *fitness* de todas as células da rede;
 - 2.2.5. Para cada grupo de Clones repetir até D disputas;
 - 2.2.5.1. selecionar duas células aleatoriamente;
 - 2.2.5.2. obter o comportamento de cada uma das células;
 - 2.2.5.3. Alterar o *fitness* das células conforme o comportamento adotado por cada uma e a tabela de pagamentos do jogo;

- 2.2.6. Para cada grupo de clone selecionar o melhor *fitness* e remover os outros;
- 2.2.7. Determinar as células com maior *fitness* e gravá-las na memória;
- 2.2.8. Introduzir uma porcentagem P de células geradas randomicamente;
- 2.2.9. Se atingida a quantidade T de iterações configuradas para troca entre populações, realizar o envio das B melhores células para área de transferência e receber as células das demais populações.

A Figura 4-14 representa o funcionamento do algoritmo:

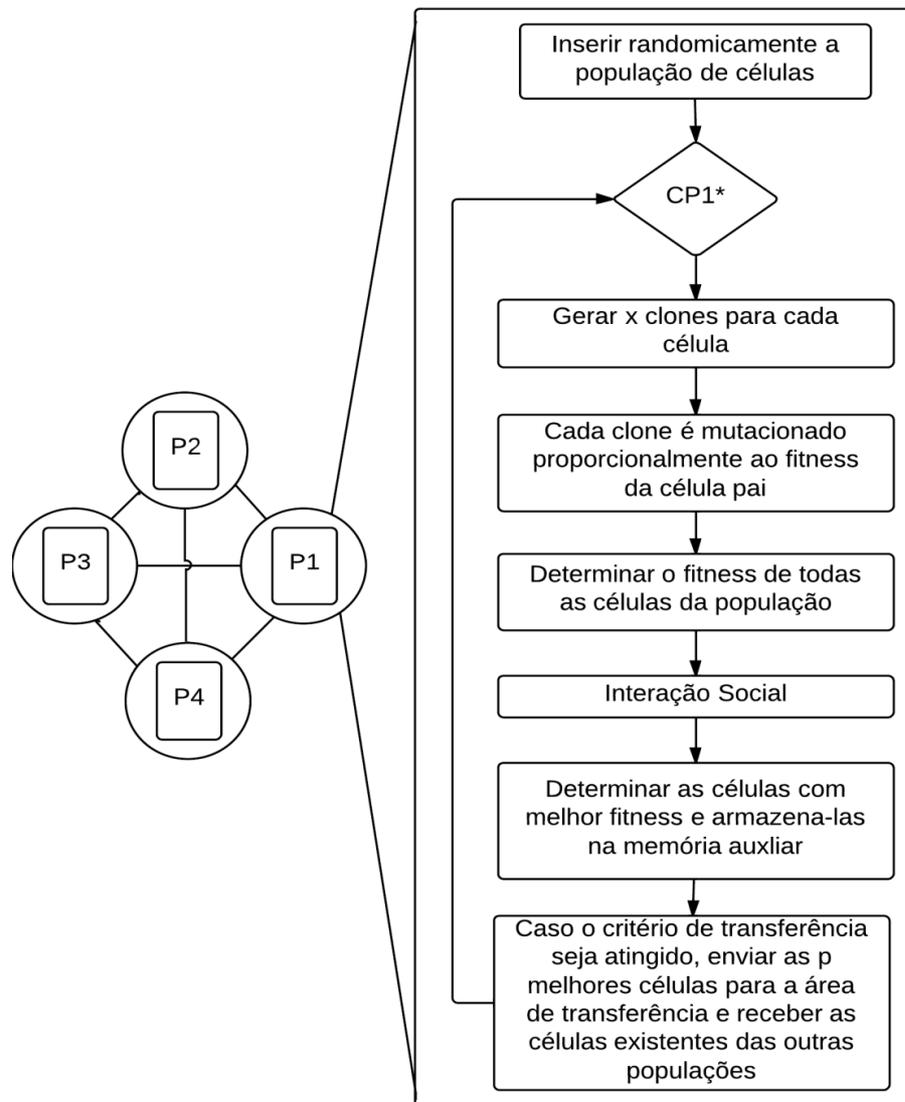


Figura 4-14 Fluxograma do Algoritmo PCAISO-GT

4.7 TRABALHOS RELACIONADOS

Na literatura podem ser encontrados trabalhos propondo estratégias de paralelização da metaheurística AIS e também de outras metaheurísticas populacionais. A seguir são apresentados alguns exemplos de trabalhos que aplicam metaheurísticas paralelas na solução de problemas.

Uma classificação dos tipos de algoritmos genéticos paralelos pode ser encontrada em (NOWOSTAWSKI et al. 1999) e (CANT'U-PAZ, 1995). Com relação à metaheurística PSO (*Particle Swarm Optimization*) também existem trabalhos (CHANG et al., 2005) que propuseram uma abordagem de granularidade grossa que apresenta ganhos, devido à estratégia de comunicação adotada, sobre o PSO tradicional (versão sequencial).

Em JIN et al. (2005) é proposto um PSO Paralelo para projetos de antenas utilizando o modelo mestre-escravo. VENTER et al. (2005) focaram no ganho de velocidade decorrente da utilização de um PSO Paralelo Assíncrono, aplicado na otimização de uma estrutura de asa de uma aeronave de transporte típica.

LI et al. (2007) propuseram um modelo de PSO paralelo de granularidade fina baseado em Unidade de Processamento Gráfico (GPU – *Graphic Processor Unit*). O trabalho aponta para uma nova tendência em computação de alto desempenho.

Na área de pesquisa dos AIS, COELLO et al. (2002) apresentam uma versão paralela do sistema imunológico, utilizando o modelo ilha, para controle dos parâmetros de um algoritmo genético. O objetivo é não obrigar a utilização de fatores de penalização na função objetivo.

Por fim, em (DABROWSKI et al, 2008) é proposto um algoritmo de seleção clonal paralelo para a resolução do problema de coloração de grafos. É utilizado o modelo de ilha onde todos os processadores trabalham com sua própria subpopulação de anticorpos, aumentando o desempenho.

4.8 RESUMO

Neste capítulo foi exposto o novo algoritmo, descrevendo as alterações realizadas e seu funcionamento. Os principais pontos abordados foram:

- Modelo Matemático do PAB: descreveu-se o modelo matemático do PAB que será utilizado nesse trabalho, sendo escolhido o modelo definido por MAURI et al. (2008a);

- Algoritmo AISO: apresentou-se o algoritmo AISO (*Artificial Immune System Optimization*) proposto por FERRY et al. (2006) inspirado no algoritmo *Opt-AiNet* proposto por DE CASTRO et al. (2002) que é a base do algoritmo proposto;
- Co-evolução: definiu-se o modelo de co-evolução que será utilizado, sendo selecionado o modelo co-evolutivo genético;
- Modelo de Paralelização: definiu-se o modelo de paralelização utilizado, sendo selecionado o modelo de subpopulações proposto por CANTU-PAZ (2000);
- Teoria dos Jogos: definiu-se a forma de aplicação dos conceitos de Teoria dos Jogos através da aplicação do clássico jogo Dilema do Prisioneiro, de forma a proporcionar um ambiente de interação social.
- Nova Metaheurística: definiu-se a estrutura do algoritmo proposto aplicado ao problema de alocação de berços, utilizando as características de Teoria dos Jogos, co-evolução e paralelismo definidas.

Com o algoritmo proposto definido, é necessária a realização de experimentos e validação buscando verificar a qualidade da nova solução proposta. Os experimentos e validações serão apresentados no capítulo cinco.

5 VALIDAÇÃO

Neste capítulo são apresentadas todas as informações sobre os testes realizados para a validação dos algoritmos propostos, das instâncias de teste aos parâmetros de configuração dos algoritmos.

5.1 INSTÂNCIAS DE TESTE

Serão utilizados o conjunto de instâncias criadas por CORDEAU et al. (2005). Essas instâncias já foram utilizadas em vários trabalhos como (MAURI et al., 2008a); (MAURI *et al.*, 2008b); (OLIVEIRA et al., 2010), entre outros, possibilitando assim uma maior base de dados para a validação e comparação dos resultados. Serão utilizadas 30 instâncias com 60 navios e 13 berços geradas aleatoriamente.

A Tabela 5-1 demonstra os resultados de vários algoritmos para as referidas instâncias.

Inst.	CPLEX ¹		TS ²		SA+RA ³		PTA/LP ⁴		CS ⁵		GRASP+PR		MELHORAS (%)				
	FO	Tempo (s)	FO	Tempo (s)	FO	Tempo (s)	FO	Tempo (s)	FO	Tempo (s)	FO	Tempo (s)	CPLEX	TS	SA+RA	PTA/LP	CS
i01	-	1415	1409	53,12	1409	74,61	1409	12,47	1409	15,45	-	0,42	0	0	0	0	0
i02	2606	1263	1261	58,94	1261	60,75	1261	12,59	1261	15,81	51,61	0,16	0	0	0	0	0
i03	2565	1139	1129	54,03	1129	135,45	1129	12,64	1129	15,37	55,98	0,88	0	0	0	0	0
i04	4353	1303	1302	67,33	1302	110,17	1302	12,59	1302	15,38	70,09	0,08	0	0	0	0	0
i05	2672	1208	1207	55,38	1207	124,70	1207	12,68	1207	15,45	54,83	0,08	0	0	0	0	0
i06	-	1262	1261	53,88	1261	78,34	1261	12,56	1261	15,64	-	0,08	0	0	0	0	0
i07	2887	1279	1279	60,52	1279	114,20	1279	12,63	1279	15,67	55,70	0	0	0	0	0	0
i08	5177	1299	1299	61,45	1299	57,06	1299	12,57	1299	15,75	74,91	0	0	0	0	0	0
i09	-	1444	1444	57,91	1444	96,47	1444	12,58	1444	15,30	-	0	0	0	0	0	0
i10	-	1213	1213	68,95	1213	99,41	1213	12,61	1213	15,47	-	0	0	0	0	0	0
i11	-	1378	1368	76,77	1369	99,34	1368	12,58	1368	15,24	-	0,73	0	0,07	0	0	0
i12	3206	1325	1325	62,84	1325	80,69	1325	12,56	1325	15,55	58,67	0	0	0	0	0	0
i13	-	1360	1360	68,19	1360	89,94	1360	12,61	1360	15,45	-	0	0	0	0	0	0
i14	-	1233	1233	75,06	1233	73,95	1233	12,67	1233	15,77	-	0	0	0	0	0	0
i15	4672	1295	1295	54,55	1295	74,19	1295	13,80	1295	15,42	72,28	0	0	0	0	0	0
i16	4320	1375	1364	63,91	1365	170,36	1364	14,46	1364	15,60	68,43	0,80	0	0,07	0	0	0
i17	-	1283	1283	56,28	1283	46,58	1283	13,73	1283	15,51	-	0	0	0	0	0	0
i18	3681	1346	1345	53,98	1345	84,02	1345	12,72	1345	15,73	63,46	0,07	0	0	0	0	0
i19	2400	1370	1370	52,83	1367	123,19	1367	13,39	1367	15,45	43,04	0,22	0,22	0	0	0	0
i20	-	1328	1328	53,38	1328	82,30	1328	12,82	1328	14,89	-	0	0	0	0	0	0
i21	-	1346	1341	53,52	1341	108,08	1341	12,68	1341	15,22	-	0,37	0	0	0	0	0
i22	3489	1332	1326	57,97	1326	105,38	1326	12,62	1326	15,45	61,99	0,45	0	0	0	0	0
i23	-	1266	1266	53,75	1266	43,72	1266	12,62	1266	15,64	-	0	0	0	0	0	0
i24	4867	1261	1260	54,09	1260	78,91	1260	12,64	1260	15,50	74,11	0,08	0	0	0	0	0
i25	1993	1379	1377	53,56	1376	96,58	1376	12,62	1376	15,64	30,96	0,22	0,07	0	0	0	0
i26	2520	1330	1318	57,34	1318	101,11	1318	12,62	1318	15,59	47,70	0,90	0	0	0	0	0
i27	3209	1261	1261	69,98	1261	82,86	1261	12,64	1261	15,22	60,70	0	0	0	0	0	0
i28	-	1365	1360	58,47	1360	52,91	1359	12,71	1359	15,28	-	0,44	0,07	0,07	0	0	0
i29	4809	1282	1280	69,09	1280	203,36	1280	12,62	1280	15,67	73,38	0,16	0	0	0	0	0
i30	-	1351	1344	70,67	1344	71,02	1344	12,58	1344	15,44	-	0,52	0	0	0	0	0
Média	3495,65	1309,70	1306,93	60,26	1306,87	93,99	1306,77	12,79	1306,77	15,48	59,873	0,222	0,012	0,007	0,000	0,000	0,000
Melhoras no tempo médio (%)											99,57	87,10	74,31	83,53	-21,03		

¹CPLEX 10.0.1

²Cordeau *et al.* (2005)

³Mauri *et al.* (2008a)

⁴Mauri *et al.* (2008b)

⁵Oliveira *et al.* (2010)

Tabela 5-1 Resultados (Lopes et al. 2011)

5.2 VALIDAÇÃO E SINTONIA DOS ALGORITMOS

Para a execução dos experimentos de avaliação dos algoritmos deste trabalho, primeiramente os métodos serão validados e sintonizados utilizando a instância i01. Decidiu-se utilizar apenas a instância i01, pois não foi encontrado nenhum tipo de classificação de complexidade que mostrasse a diferença entre as instâncias. Sendo assim, não haveria nenhum tipo de garantia de ganho de desempenho na realização da sintonia de parâmetros utilizando todas as instâncias.

Todos os algoritmos propostos foram implementados na linguagem de programação JAVA e os experimentos para validação foram realizados em um computador com processador Core i5 2.00GHz, memória RAM de 4GB e Sistema Operacional Windows 7.

Por se tratar de um problema de minimização, os melhores resultados serão sempre os menores valores encontrados. Na Tabela 5-2, são apresentadas as soluções ótimas para cada instância de teste proposta por CORDEAU et al. (2005).

Instância	Solução Ótima
i01	1409
i02	1261
i03	1129
i04	1302
i05	1207
i06	1261
i07	1279
i08	1299
i09	1444
i10	1213
i11	1368
i12	1325
i13	1360
i14	1233
i15	1295
i16	1364
i17	1283
i18	1345
i19	1367
i20	1328
i21	1341
i22	1326

i23	1266
i24	1260
i25	1376
i26	1318
i27	1261
i28	1359
i29	1280
i30	1344

Tabela 5-2 Soluções Ótimas das Instâncias de Teste

5.3 PARÂMETROS DA METAHEURÍSTICA

Os algoritmos apresentam parâmetros que podem ser ajustados durante os testes de sintonia, tendo como objetivo obter o melhor desempenho do algoritmo. De acordo com a configuração utilizada, o algoritmo pode fornecer melhores resultados. Os parâmetros passíveis de configuração são:

1. Tamanho da População: define a quantidade de células da população;
2. Taxa / Faixa de Hipermutação (Mutaç o): define a taxa ou as faixas de muta o. No caso do PAB, informa a quantidade de trocas que ser o realizadas em cada indiv duo e em cada itera o.
3. Taxa / Faixa de Gera o de Clones: define a quantidade ou as faixas de gera o de clones. O valor   um inteiro.
4. N mero de Itera es: define o crit rio de parada a ser adotada em cada subpopula o;
5. Tamanho da Mem ria: quantidade de c lulas que ser  armazenada na mem ria principal do algoritmo;
6. N mero de Subpopula es: define a quantidade de subpopula es que ser o geradas a partir da popula o inicial informada;
7. Per odo para a migra o de C lulas: define o per odo de troca de informa es entre as subpopula es;
8. N mero de C lulas que ser o migradas: define a quantidade de c lulas que ser o migradas de cada subpopula o no per odo de migra o;

9. Número de Iteração Mínima para mudança de população: define o número mínimo de iterações que a célula precisa ter para poder se candidatar a migração;
10. Número de Partidas: define o número de partidas do jogo Dilema do Prisioneiro que serão realizadas em cada iteração;
11. Número de Rodadas: define o número de rodadas que serão realizadas em cada partida do jogo Dilema do Prisioneiro.

Os parâmetros definidos nos itens de 1 a 5 são empregados em todos os algoritmos descritos (AISO, PCAISO e PCAISO-GT). Os parâmetros definidos nos itens de 6 a 9 são empregados apenas nos algoritmos que empregam a estratégia de co-evolução (PCAISO e PCAISO-GT). Por fim, parâmetros definidos nos itens 10 e 11 são empregados nos algoritmos que utilizem a Teoria dos Jogos (PCAISO-GT).

5.3.1 Parâmetros Gerais (AISO)

Em virtude da grande quantidade de configurações possíveis, definiu-se um conjunto de parâmetros para identificação da melhor sintonia. Foram assumidos diferentes valores para estes parâmetros, sendo avaliada cada uma das combinações de acordo com uma média do valor final da função objetivo, baseada em 10 execuções do algoritmo. Desta forma, a melhor combinação de parâmetros foi selecionada como configuração padrão para a execução das simulações das demais instâncias. Como são parâmetros gerais, a configuração definida fará parte da execução dos testes de todos os algoritmos apresentados nesse trabalho. A Tabela 5-3 exhibe os conjuntos de valores assumidos para os diferentes parâmetros a serem sintonizados, buscando a melhor configuração a fim de validar a ferramenta desenvolvida.

	Conjunto de valores		
	200	400	600
Tamanho da População	200	400	600
Quantidade de Faixas de Hipermutação	3	4	5
Quantidade de Faixas de Clones	3	4	5
Quantidade de Iterações	800	1500	2000
Tamanho da Memória	20%	40%	60%

Tabela 5-3 Parâmetros de Teste

De acordo com os valores definidos, foram definidos 11 conjuntos de teste, tendo sido realizadas cinco execuções do algoritmo sequencial (AISO) utilizando a instância i01. Os resultados obtidos estão detalhados na Tabela 5-4.

Lista de Testes							
	População	Hipermutação	Clones	Iterações	Memória	Média	Desvio Padrão
Sintonia 1	200	3	3	800	20%	1414,00	2,28
Sintonia 2	400	3	3	800	20%	1425,00	4,98
Sintonia 3	600	3	3	800	20%	1429,20	8,16
Sintonia 4	200	4	3	800	20%	1412,00	2,83
Sintonia 5	200	5	3	800	20%	1411,60	2,42
Sintonia 6	200	5	4	800	20%	1411,20	2,64
Sintonia 7	200	5	5	800	20%	1411,00	2,53
Sintonia 8	200	5	5	1500	20%	1411,20	3,12
Sintonia 9	200	5	5	2000	20%	1410,80	2,14
Sintonia 10	200	5	5	1500	40%	1410,60	2,24
Sintonia 11	200	5	5	1500	60%	1410,40	1,96

Tabela 5-4 Conjunto de Parâmetros de Teste

O conjunto de parâmetros de número 11 teve o melhor resultado médio dentre as combinações utilizadas para o algoritmo sequencial. Desta forma, os seguintes parâmetros tiveram definidos os seus valores:

- Tamanho da População: 200;
- Quantidade de Faixas de Hipermutação: 5;
- Quantidade de Faixas de Clone: 5;
- Quantidade de Iteração: 1500;
- Tamanho da Memória: 60%.

5.3.2 Parâmetros do Algoritmo Co-evolutivo (PCAISO)

Além dos parâmetros gerais, o algoritmo co-evolutivo apresenta parâmetros adicionais que também devem ser sintonizados. Seguindo o modelo da avaliação dos parâmetros gerais, foram definidos conjuntos de valores para os parâmetros utilizados pelo algoritmo co-evolutivo. Os conjuntos de valores assumidos para os diferentes parâmetros a serem sintonizados são apresentados na Tabela 5-5.

	Conjunto de valores		
	2	3	4
Quantidade de Subpopulações	2	3	4
Período para Migração	100	200	400
Quantidade de Células a Migrar	1	10	50

Tabela 5-5 Parâmetros de Teste do Algoritmo Co-evolutivo

De acordo com os valores definidos, foram definidos sete conjuntos de teste. Assim como realizado na identificação dos valores para os parâmetros gerais, para cada conjunto de parâmetros foram realizadas cinco execuções do algoritmo utilizando a instância i01. Os resultados obtidos estão detalhados na Tabela 5-6.

Lista de Testes					
	Número de Subpopulações	Período de Migração	Número de Células a Migrar	Média	Desvio Padrão
Sintonia 1	2	100	1	1411,00	1,41
Sintonia 2	3	100	1	1410,60	1,36
Sintonia 3	4	100	1	1410,20	1,17
Sintonia 4	4	200	1	1410,00	1,54
Sintonia 5	4	400	1	1409,80	1,16
Sintonia 6	4	400	10	1410,40	1,02
Sintonia 7	4	400	50	1410,60	1,02

Tabela 5-6 Conjunto de Parâmetros de Teste do Algoritmo Co-evolutivo

O conjunto de parâmetros de número cinco teve o melhor resultado médio dentre as combinações utilizadas para o algoritmo co-evolutivo. Desta forma, os seguintes parâmetros tiveram definidos os seus valores:

- Número de Subpopulações: 4;
- Período para Migração: 400;
- Número de Células a Migrar: 1.

Esta configuração será utilizada para os testes do algoritmo co-evolutivo, e servirá como base para a configuração do algoritmo que emprega teoria dos jogos.

5.3.3 Algoritmo com Teoria de Jogos (PCAISO-GT)

Além dos parâmetros gerais e dos parâmetros do algoritmo co-evolutivo, o algoritmo com Teoria dos Jogos também apresenta parâmetros que devem ser sintonizados. Seguindo o modelo de sintonia dos parâmetros empregado nos casos anteriores, novamente foram definidos conjuntos de valores para os diferentes parâmetros utilizados pelo algoritmo com Teoria dos jogos. Os conjuntos de valores assumidos para os diferentes parâmetros a serem sintonizados são apresentados na Tabela 5-7.

Conjunto de valores			
Quantidade de Partidas	5	15	30
Quantidade de Rodadas	5	15	30

Tabela 5-7 Parâmetros de Teste do Algoritmo GT

De acordo com os valores definidos, foram definidos cinco conjuntos de teste, tendo sido realizadas cinco execuções do algoritmo co-evolutivo com Teoria dos Jogos, aplicado na solução da instância i01. Os resultados obtidos estão detalhados na Tabela 5-8.

Lista de Testes				
	Número de Partidas	Número de Rodadas	Média	Desvio Padrão
Sintonia 1	5	5	1410,80	0,98
Sintonia 2	15	5	1410,80	0,75
Sintonia 3	30	5	1410,00	0,89
Sintonia 4	15	15	1409,40	0,49
Sintonia 5	15	30	1410,00	0,63

Tabela 5-8 Conjunto de Parâmetros de Teste do Algoritmo GT

O conjunto de parâmetros de número quatro teve o melhor resultado médio dentre as combinações utilizadas para o algoritmo com Teoria dos Jogos. Desta forma, os seguintes parâmetros tiveram definidos os seus valores:

- Quantidade de Partidas: 15;
- Quantidade de Rodadas: 15.

5.3.4 Resumo

Nesse capítulo foram apresentados os experimentos de configuração e validação dos algoritmos propostos no trabalho. Em virtude da grande quantidade de instâncias de teste e da inexistência de uma definição de dificuldade para as mesmas, os experimentos foram realizados utilizando a instância i01.

Para cada sintonia foram realizadas cinco execuções e, com base nos resultados, foram calculadas as médias e desvios padrão, os quais foram utilizados para avaliação da sintonia que apresentou o melhor desempenho. A melhor configuração de parâmetros identificada foi utilizada em todas as execuções realizadas para geração dos resultados finais do trabalho, que serão apresentados no capítulo seis.

6 EXPERIMENTOS E RESULTADOS

Com a sintonia dos parâmetros empregados nos algoritmos desenvolvidos e as suas respectivas validações, neste capítulo são apresentados os experimentos e a análise dos resultados obtidos para cada um dos algoritmos quando aplicados em cada uma das instâncias de teste.

Todos os algoritmos apresentaram boas soluções visto que produziram resultados próximos e, na grande maioria dos casos, iguais ao melhor resultado para as instâncias do PAB avaliadas.

Foram realizadas 10 execuções para cada algoritmo em cada uma das instâncias propostas, utilizando os parâmetros definidos no capítulo 5. Os resultados foram consolidados e são apresentados na Tabela 6-1. Na tabela são apresentados o melhor resultado de cada algoritmo em cada um das instâncias (Melhor), a média dos melhores resultados das 10 execuções (Média), e o desvio padrão (D. P.).

Nas execuções dos algoritmos com subpopulações, o número de populações é igual ao número de processadores utilizados, ou seja:

- Algoritmo com duas subpopulações: dois processadores;
- Algoritmo com três subpopulações: três processadores;
- Algoritmo com quatro subpopulações: quatro processadores.

Na Tabela 6-1, a sigla GT identifica o algoritmo que utilizou o conceito de teoria dos jogos e, esse conceito foi somente aplicado ao algoritmo com subpopulações que obteve o melhor resultado, no caso o algoritmo com quatro subpopulações.

O algoritmo com Teoria dos Jogos teve o melhor desempenho dentre todos os algoritmos testados. Ele chegou ao resultado ótimo em 29 das 30 (96,67%) das instâncias avaliadas, sendo a instância i28 a única exceção. Segue a lista de resultados ótimos encontrados pelos algoritmos:

- Algoritmo AISO: 18 das 30 instâncias (60,00%)
- Algoritmo PCAISO (duas subpopulações): 25 das 30 instâncias (83,33%);
- Algoritmo PCAISO (três subpopulações): 27 das 30 instâncias (90,00%);
- Algoritmo PCAISO (quatro subpopulações): 28 das 30 instâncias (93,33%);

- Algoritmo PCAISO (quatro subpopulações) + GT (Teoria dos Jogos): 29 das 30 instâncias (96,67%).

Inst.	AISO			PCAISO (2)			PCAISO (3)			PCAISO (4)			PCAISO (4) + GT		
	Melhor	Media	D. P.	Melhor	Media	D. P.	Melhor	Media	D. P.	Melhor	Media	D. P.	Melhor	Media	D. P.
i01	1409	1412,80	2,04	1409	1412,70	1,62	1409	1411,30	1,95	1409	1411,20	1,40	1409	1409,90	0,83
i02	1261	1262,30	3,03	1261	1261,00	0,00	1261	1261,00	0,00	1261	1261,00	0,00	1261	1261,00	0,00
i03	1129	1132,70	2,33	1129	1130,80	2,18	1129	1130,30	2,05	1129	1130,00	1,73	1129	1129,40	0,49
i04	1302	1305,60	5,00	1302	1304,00	5,02	1302	1302,70	1,19	1302	1302,20	0,40	1302	1302,20	0,40
i05	1207	1207,50	1,50	1207	1207,30	0,90	1207	1207,00	0,00	1207	1207,00	0,00	1207	1207,00	0,00
i06	1261	1263,30	4,27	1261	1261,30	0,64	1261	1261,20	0,60	1261	1261,00	0,00	1261	1261,00	0,00
i07	1280	1281,70	2,19	1279	1279,80	0,40	1279	1279,80	0,40	1279	1279,70	0,46	1279	1279,20	0,40
i08	1300	1306,70	4,73	1299	1302,50	2,73	1299	1301,10	2,88	1299	1299,90	1,37	1299	1299,40	0,49
i09	1444	1445,10	2,02	1444	1444,80	1,17	1444	1444,50	0,50	1444	1444,40	0,49	1444	1444,00	0,00
i10	1213	1214,00	1,67	1213	1213,50	0,67	1213	1213,10	0,30	1213	1213,10	0,30	1213	1213,00	0,00
i11	1370	1380,70	7,21	1368	1374,30	4,00	1368	1372,70	2,37	1368	1372,10	2,30	1368	1369,50	0,67
i12	1327	1333,80	5,25	1325	1333,10	5,43	1325	1329,90	2,98	1325	1327,50	2,42	1325	1326,30	0,90
i13	1360	1360,50	1,20	1360	1360,00	0,00	1360	1360,00	0,00	1360	1360,00	0,00	1360	1360,00	0,00
i14	1233	1233,80	1,4	1233	1233,00	0,00	1233	1233,00	0,00	1233	1233,00	0,00	1233	1233,00	0,00
i15	1295	1297,40	2,94	1295	1297,40	2,46	1295	1296,00	1,79	1295	1295,60	0,80	1295	1295,20	0,40
i16	1364	1366,80	2,64	1364	1366,20	1,08	1364	1365,60	1,56	1364	1365,30	1,00	1364	1364,60	0,80
i17	1283	1284,40	2,84	1283	1283,00	0,00	1283	1283,00	0,00	1283	1283,00	0,00	1283	1283,00	0,00
i18	1345	1345,00	0,00	1345	1345,00	0,00	1345	1345,00	0,00	1345	1345,00	0,00	1345	1345,00	0,00
i19	1368	1373,40	3,23	1368	1373,00	3,29	1368	1370,90	2,98	1367	1369,10	1,87	1367	1367,80	0,87
i20	1329	1330,70	3,80	1329	1330,50	2,91	1328	1329,30	1,35	1328	1329,10	0,54	1328	1328,60	0,66
i21	1343	1346,30	2,45	1343	1345,40	2,54	1341	1344,50	2,62	1341	1344,10	1,87	1341	1341,80	0,98
i22	1326	1332,40	5,46	1326	1331,60	2,94	1326	1330,10	4,23	1326	1329,20	2,71	1326	1327,00	0,89
i23	1266	1270,80	4,42	1266	1268,30	2,41	1266	1266,90	1,58	1266	1266,70	0,78	1266	1266,20	0,40
i24	1260	1262,20	1,83	1260	1262,00	1,84	1260	1261,30	1,79	1260	1260,90	1,58	1260	1260,10	0,30
i25	1378	1383,30	5,35	1377	1380,10	3,05	1377	1379,70	3,61	1377	1378,40	1,56	1376	1377,20	0,75
i26	1324	1327,70	4,78	1318	1323,50	3,44	1318	1321,10	3,18	1318	1320,60	1,69	1318	1318,70	0,46
i27	1261	1261,90	1,37	1261	1261,20	0,60	1261	1261,00	0,00	1261	1261,00	0,00	1261	1261,00	0,00
i28	1361	1364,00	3,79	1360	1363,10	1,64	1360	1362,10	1,45	1360	1361,60	1,56	1360	1360,80	0,98
i29	1281	1284,00	1,18	1281	1283,90	1,58	1281	1282,70	1,27	1280	1281,80	1,47	1280	1280,80	1,17
i30	1346	1354,90	4,09	1344	1351,40	4,45	1344	1350,70	3,58	1344	1350,20	5,58	1344	1345,80	2,23
Média	1307,53	1310,86	3,13	1307,00	1309,46	1,97	1306,90	1308,58	1,54	1306,83	1308,12	1,13	1306,80	1307,28	0,50

Tabela 6-1 Resultados Consolidados dos Testes

Com base nos resultados, a seguir são apresentadas as análises dos resultados de cada um dos algoritmos propostos.

6.1 ALGORITMO AISO

O algoritmo AISO apresentou resultados satisfatórios para a maior parte das instâncias, encontrando o ótimo global em 18 das 30 instâncias de teste (60%).

A Figura 6-1 apresenta o comportamento do melhor resultado do algoritmo da geração 1 até a 350. Percebe-se que, inicialmente, a curva de evolução é bastante acentuada (até aproximadamente a geração 130) em virtude da grande diversidade inicial da população.

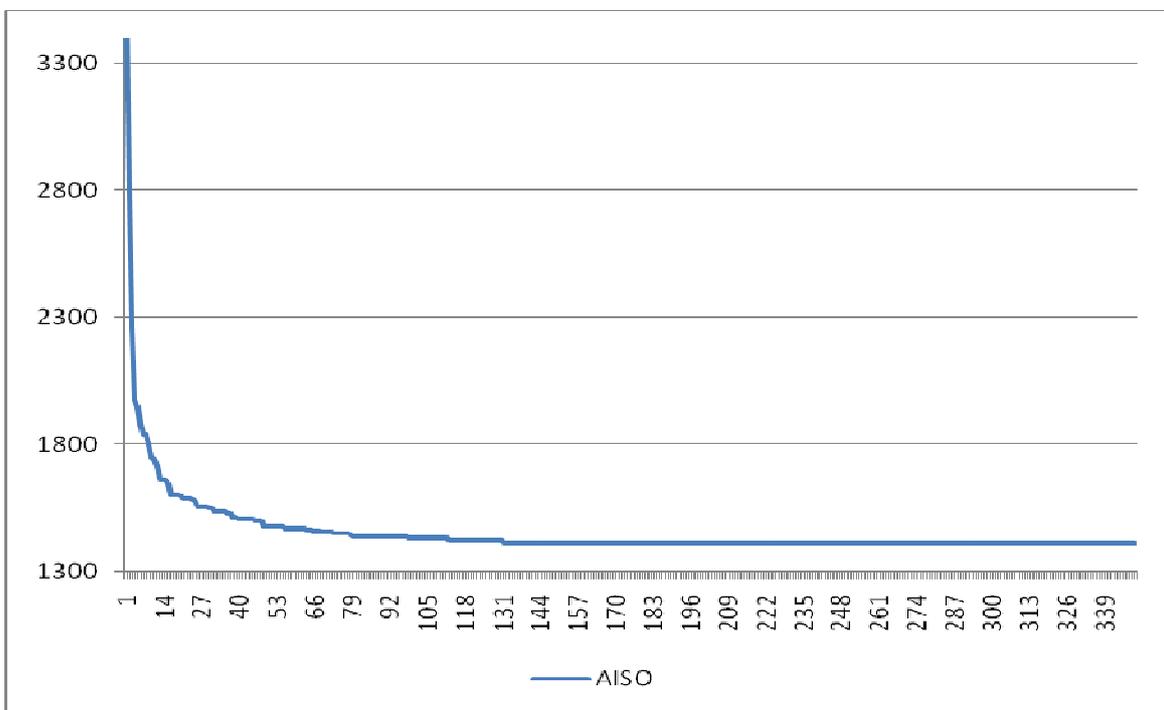


Figura 6-1 Gráfico de Evolução do Algoritmo AISO

Com o passar das gerações, a variabilidade da população tende a diminuir em virtude do elitismo gerado pelo armazenamento das melhores soluções na memória auxiliar, as quais são perpetuadas por todas as gerações. Esse comportamento pode ser visualizado na Figura 6-1, das gerações 131 até a 350.

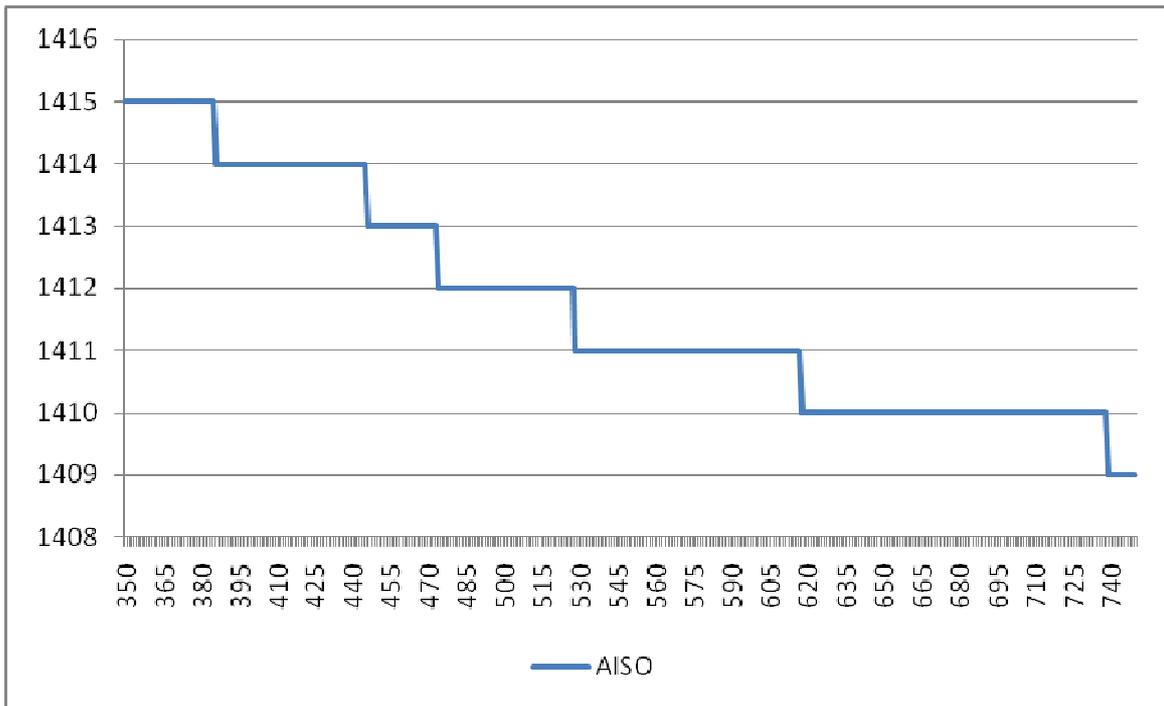


Figura 6-2 Gráfico de Evolução do Algoritmo AISO

A Figura 6-2 apresenta o gráfico de evolução da geração 350 a 750. Nele, é possível perceber que o algoritmo apresenta grandes períodos de tempo sem obter melhora na função objetivo, evidenciando a dificuldade em sair de ótimos locais. Essa característica pode ser observada também na Tabela 6-1, onde o algoritmo AISO apresenta os maiores valores de desvio padrão e média entre todos os algoritmos propostos.

6.2 ALGORITMO PCAISO

O algoritmo PCAISO apresentou resultados satisfatórios, tendo em vista que a qualidade de suas soluções aumentou consideravelmente em relação ao algoritmo original (AISO). O algoritmo, em seu melhor caso (4 subpopulações), atingiu a porcentagem de 93,33% de efetividade em relação ao ótimo global, sendo que nas duas instâncias na qual o resultado não foi atingido, a diferença foi de apenas 1 Unidade.

O objetivo da inclusão de conceitos de co-evolução no algoritmo tem o objetivo de promover uma maior variabilidade, tendo em vista que a evolução de diferentes populações em ambientes diferentes tende a promover maiores diferenciações na população global, promovendo assim uma varredura maior no espaço de busca.

A troca de informações entre as subpopulações tem papel fundamental, pois é a responsável por introduzir características novas, redirecionando o local de busca e, dessa maneira, promovendo, em determinadas situações, a aceleração da evolução de uma determinada população.

As Figuras 6-3, 6-4 e 6-5 apresentam o gráfico com o comportamento das subpopulações na melhor simulação realizada com o algoritmo PCAISO (4) para a instância i01.

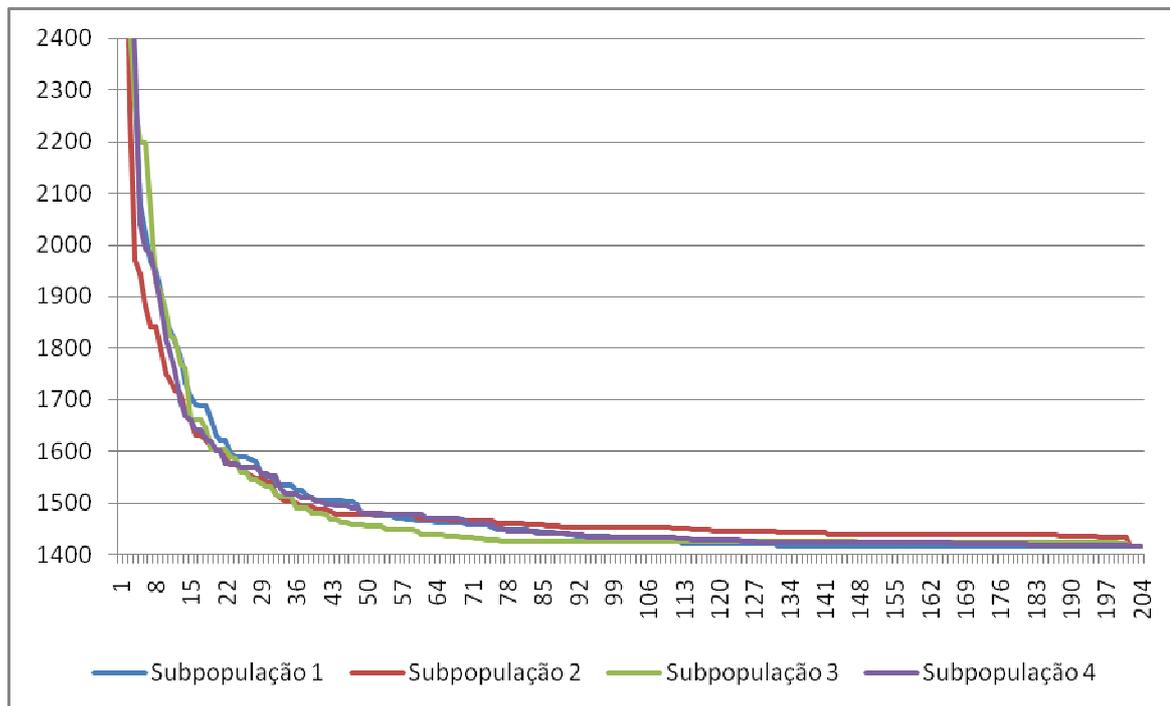


Figura 6-3 Gráfico de Evolução do Algoritmo PCAISO

A Figura 6-3 apresenta o comportamento do melhor resultado de cada uma das 4 subpopulações da geração 1 até a 204. Pode-se perceber que inicialmente são encontrados ótimos locais diferentes para cada uma das subpopulações. Esse comportamento é atribuído à ausência de comunicação entre as subpopulações até a geração 200.

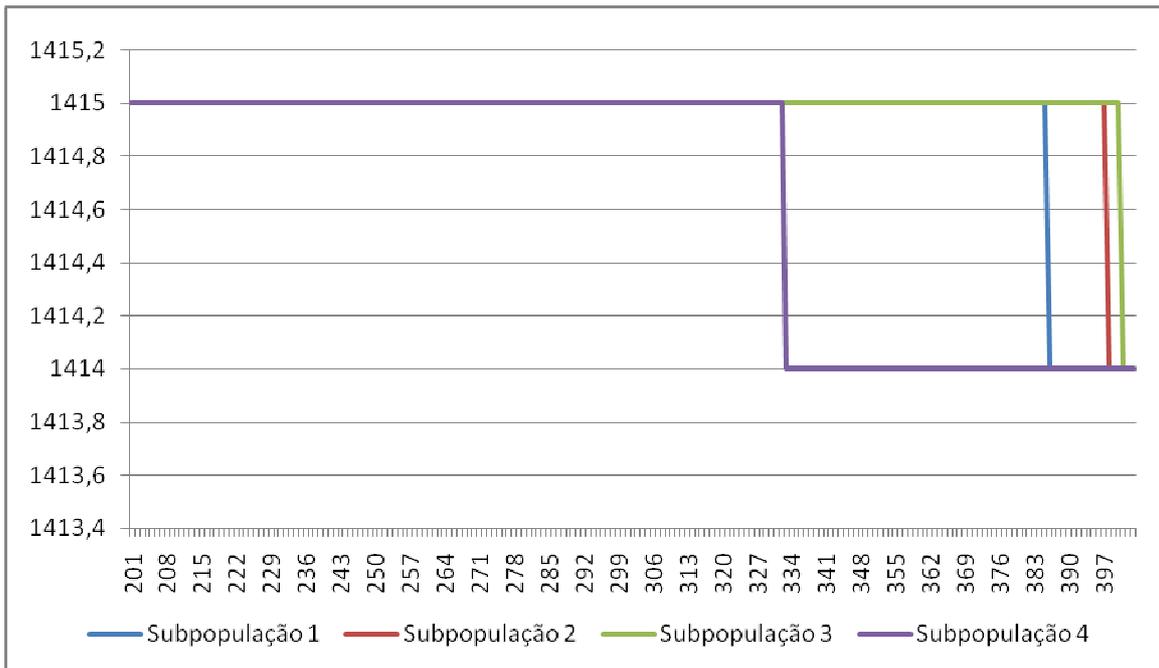


Figura 6-4 Gráfico de Evolução do Algoritmo PCAISO

A Figura 6-4 apresenta o gráfico do comportamento do algoritmo das gerações 201 até a 401. Percebe-se que após a troca de informações (geração 200) todas as subpopulações apresentam a mesma solução ótima atual, ficando o sistema um longo período sem evolução. A alteração do ótimo local ocorreu apenas na geração 333. Isso aconteceu em virtude do redirecionamento do espaço de busca ocasionado pela troca de informação entre as subpopulações e pelo elitismo característico da solução, que faz com que o algoritmo tenha alta tendência em ficar preso nos ótimos locais do problema.

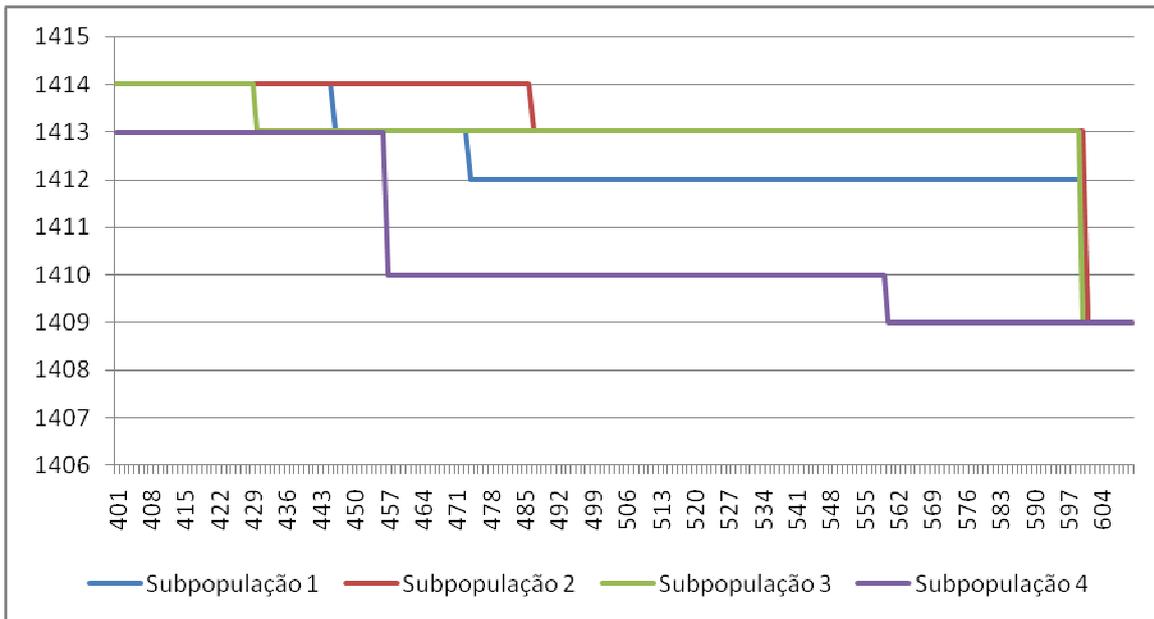


Figura 6-5 Gráfico de Evolução do Algoritmo PCAISO

A Figura 6-5 apresenta o gráfico do comportamento do algoritmo da geração 401 até a 604. Nesse período, o algoritmo encontra ótimo global do problema (1409) na geração 560. Pode-se verificar que o comportamento do algoritmo nesse período é bastante similar ao período anteriormente apresentado na Figura 6-4, apresentando longos períodos sem alteração do ótimo local. Percebe-se que na geração 601 as soluções convergem para a mesma solução ótima, isso se deve a troca de informações realizada na iteração 600.

6.3 ALGORITMO PCAISO-GT

Dentre todos os algoritmos testados, o algoritmo PCAISO-GT foi o que teve o melhor desempenho. O algoritmo encontrou 96,63% dos melhores resultados para instâncias de teste avaliadas, ou seja, em 29 das 30 instâncias.

As Figuras 6-6 e 6-7 apresentam o gráfico com o comportamento das subpopulações na melhor das simulações realizada com o algoritmo PCAISO (4) - GT.

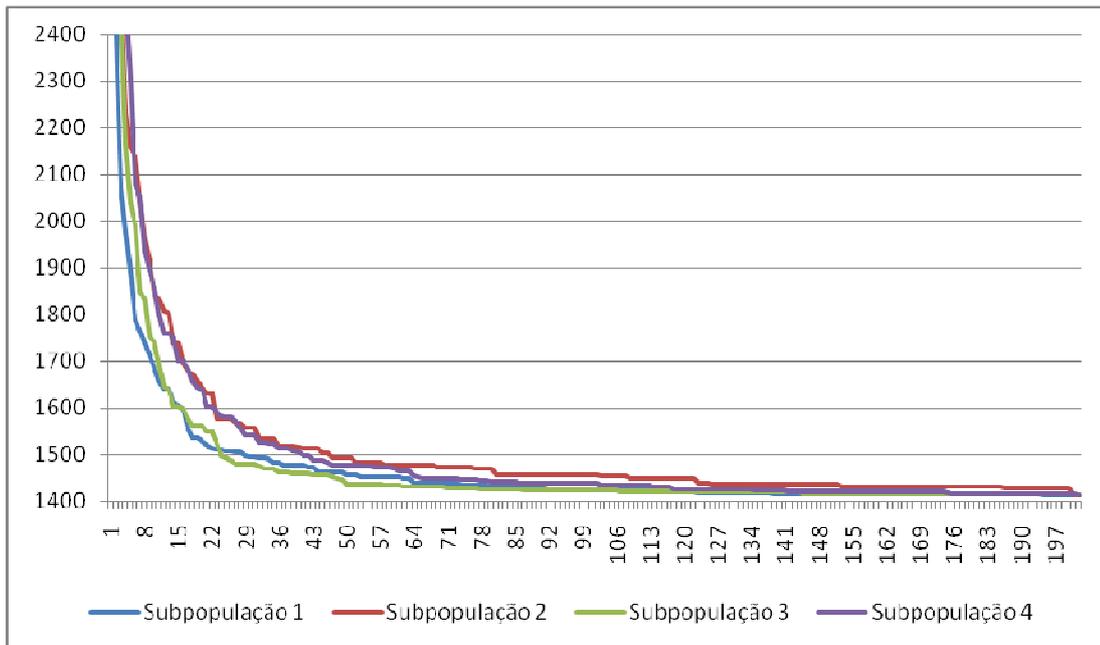


Figura 6-6 Gráfico de Evolução do Algoritmo PCAISO-GT

A Figura 6-6 apresenta o comportamento do melhor resultado de cada uma das 4 subpopulações da geração 1 até a 204. Pode-se perceber que, assim como no algoritmo PCAISO, inicialmente são encontrados ótimos locais diferentes para cada uma das subpopulações.

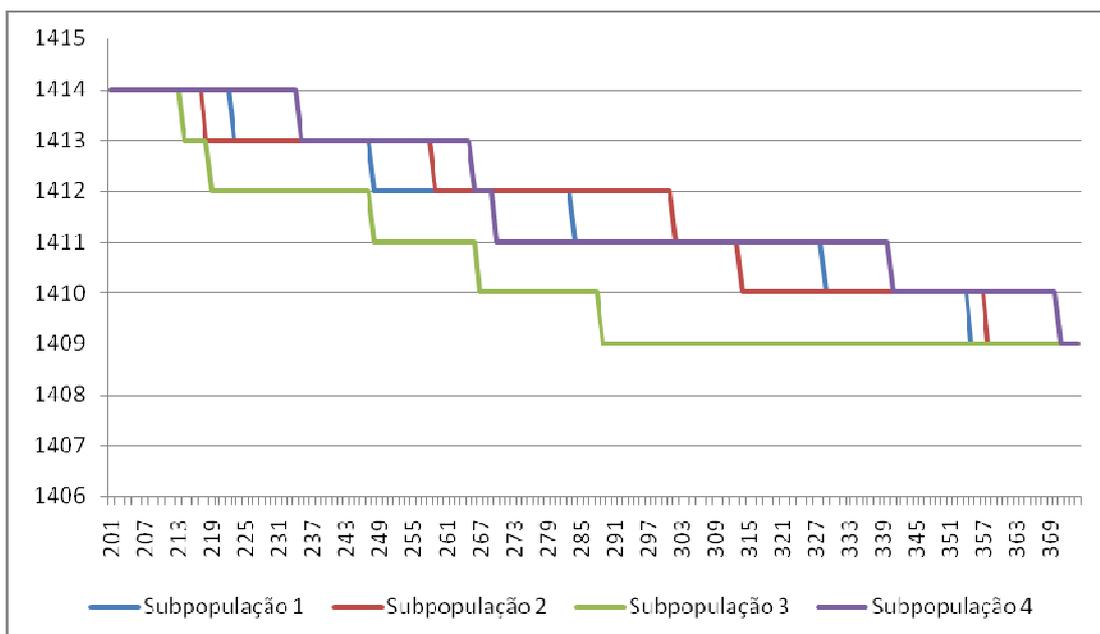


Figura 6-7 Gráfico de Evolução do Algoritmo PCAISO-GT

A Figura 6-7 apresenta o gráfico do comportamento do algoritmo das gerações 201 até a 401. Percebe-se que, diferentemente do algoritmo PCAISO, que o algoritmo com Teoria dos jogos continua apresentando uma constante melhora após a troca de informações (geração 200 o gráfico da figura 6-5 está com o eixo horizontal variando de 1 a 169). Isso se deve à utilização da Teoria dos Jogos como balanceador do elitismo característico do AIS. A atribuição de características fenotípicas fazem com que não apenas as características genéticas sejam consideradas, mas também o resultado do comportamento adotado pela célula nas partidas realizadas.

6.4 COMPARATIVO ENTRE METAHEURÍSTICAS

Com o intuito de promover de maneira mais clara a avaliação final dos resultados obtidos neste capítulo, são apresentados na Tabela 6-2 os resultados comparativos com outras quatro metaheurísticas aplicadas as mesmas instâncias de teste.

Inst.	SA+RA (1)	PATA/LP (2)	CS (3)	GRASP+PR (4)	PCAISO-GT	GAP (%)				
	FO	FO	FO	FO	FO	SA+RA	PTA/LP	CS	GRASP+PR	PCAISO-GT
i01	1409	1409	1409	1409	1409	0,00	0,00	0,00	0,00	0,00
i02	1261	1261	1261	1261	1261	0,00	0,00	0,00	0,00	0,00
i03	1129	1129	1129	1129	1129	0,00	0,00	0,00	0,00	0,00
i04	1302	1302	1302	1302	1302	0,00	0,00	0,00	0,00	0,00
i05	1207	1207	1207	1207	1207	0,00	0,00	0,00	0,00	0,00
i06	1261	1261	1261	1261	1261	0,00	0,00	0,00	0,00	0,00
i07	1279	1279	1279	1279	1279	0,00	0,00	0,00	0,00	0,00
i08	1299	1299	1299	1299	1299	0,00	0,00	0,00	0,00	0,00
i09	1444	1444	1444	1444	1444	0,00	0,00	0,00	0,00	0,00
i10	1213	1213	1213	1213	1213	0,00	0,00	0,00	0,00	0,00
i11	1368	1369	1368	1368	1368	0,00	0,07	0,00	0,00	0,00
i12	1325	1325	1325	1325	1325	0,00	0,00	0,00	0,00	0,00
i13	1360	1360	1360	1360	1360	0,00	0,00	0,00	0,00	0,00
i14	1233	1233	1233	1233	1233	0,00	0,00	0,00	0,00	0,00
i15	1295	1295	1295	1295	1295	0,00	0,00	0,00	0,00	0,00
i16	1364	1365	1364	1364	1364	0,00	0,07	0,00	0,00	0,00
i17	1283	1283	1283	1283	1283	0,00	0,00	0,00	0,00	0,00
i18	1345	1345	1345	1345	1345	0,00	0,00	0,00	0,00	0,00
i19	1370	1367	1367	1367	1367	0,22	0,00	0,00	0,00	0,00
i20	1328	1328	1328	1328	1328	0,00	0,00	0,00	0,00	0,00
i21	1341	1341	1341	1341	1341	0,00	0,00	0,00	0,00	0,00

i22	1326	1326	1326	1326	1326	0,00	0,00	0,00	0,00	0,00
i23	1266	1266	1266	1266	1266	0,00	0,00	0,00	0,00	0,00
i24	1260	1260	1260	1260	1260	0,00	0,00	0,00	0,00	0,00
i25	1377	1376	1376	1376	1376	0,07	0,00	0,00	0,00	0,00
i26	1318	1318	1318	1318	1318	0,00	0,00	0,00	0,00	0,00
i27	1261	1261	1261	1261	1261	0,00	0,00	0,00	0,00	0,00
i28	1360	1360	1359	1359	1360	0,07	0,07	0,00	0,00	0,07
i29	1280	1280	1280	1280	1280	0,00	0,00	0,00	0,00	0,00
i30	1344	1344	1344	1344	1344	0,00	0,00	0,00	0,00	0,00
Média	1306,93	1306,87	1306,77	1306,77	1306,80	0,12	0,07	0,00	0,00	0,07
(1) Mauri et al. (2005)	(2) Mauri et al. (2008b)		(3) Oliveira et al. (2010)			(4) Lopes et al. (2011)				

Tabela 6-2 Comparativo entre Metaheurísticas

A Tabela 6-2 apresenta os resultados obtidos por outras metaheurísticas, apresentadas nos trabalhos de MAURI (2005 e 2008b), OLIVEIRA (2010) e LOPES (2011), bem como o erro percentual em relação à solução ótima (GAP). Os resultados são comparados com os obtidos pelo algoritmo PCAISO-GT, algoritmo que dentre os propostos neste trabalho, obteve os melhores resultados para as 30 instâncias de teste propostas, quando empregadas quatro subpopulações.

O algoritmo PCAISO-GT apenas não obteve o ótimo global para a instância i28, ficando uma unidade atrás, resultado equivalente aos trabalhos de MAURI (2005 e 2008b) e inferior aos trabalhos de OLIVEIRA (2010) e LOPES (2011). Em todas as outras instâncias os resultados encontrados foram os ótimos globais.

Em relação ao tempo de execução, foram utilizadas duas métricas normalmente utilizadas para análise de desempenho, são elas:

- *Speedup*: é uma medida do grau de desempenho. Ela mede a razão entre o tempo de execução sequencial e o tempo de execução em paralelo.
- Eficiência: é uma medida do grau de aproveitamento dos recursos computacionais. Ela mede a razão entre o grau de desempenho e os recursos computacionais

As figuras 6-8 e 6-9 representam graficamente o comparativo de desempenho entre as metaheurísticas PCAISO (2, 3 e 4 subpopulações).

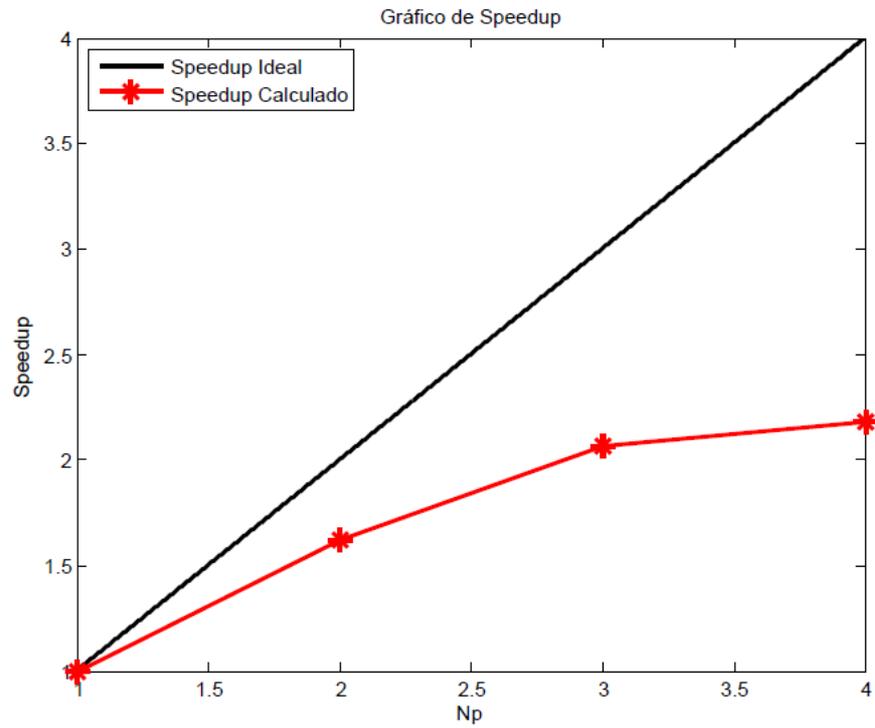
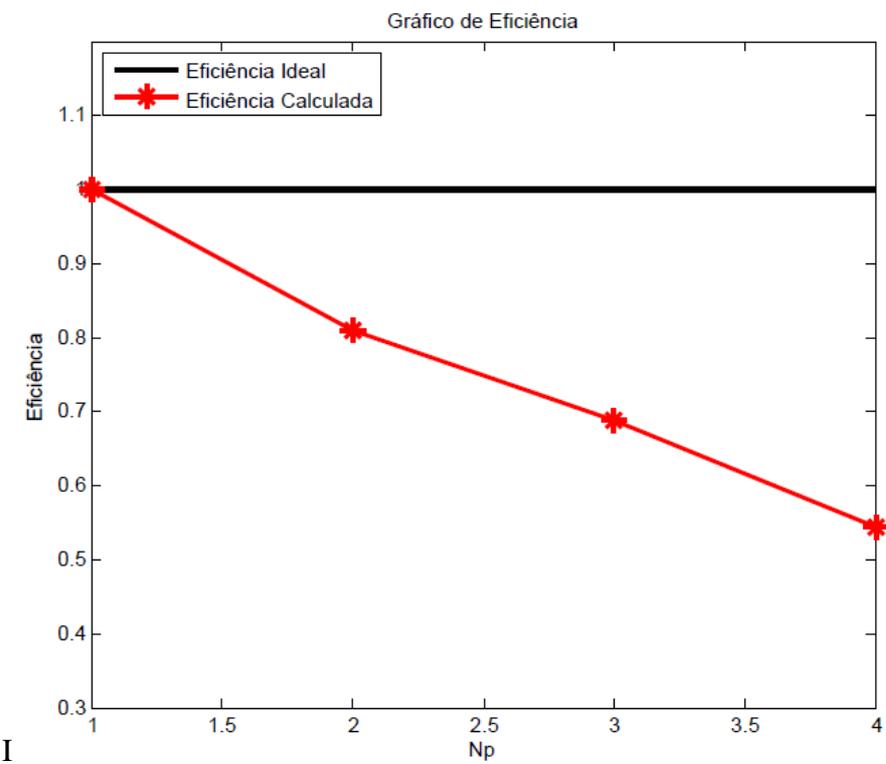


Figura 6-8 Gráfico comparativo de *Speedup*



I

Figura 6-9 Gráfico comparativo de Eficiência

A Tabela 6-3 demonstra os resultados obtidos e representados pelas figuras 6-8 e 6-9.

Tempo de Execução Médio (s)	<i>Speedup</i> Ideal	<i>Speedup</i> Real	Eficiência(%)
169,19			
104,53	2	1,62	0,81
81,91	3	2,07	0,69
77,68	4	2,18	0,54

Tabela 6-3 Tempo médio de execução, eficiência e *speedup*

Através dos gráficos de *Speedup* e Eficiência e pelos valores apresentados na Tabela 6-3, percebe-se uma perda de desempenho computacional a medida que aumenta a quantidade de subpopulações empregadas no processo de solução. Apesar da redução acentuada de eficiência relacionada ao tempo de processamento, observa-se que a qualidade final da solução é melhor quando considerado um número maior de subpopulações, conforme foi demonstrado anteriormente pelos resultados apresentados na Tabela 6-1.

7 CONSIDERAÇÕES FINAIS

Este trabalho teve por finalidade o desenvolvimento de metaheurísticas para solucionar o Problema de Alocação de Berços (PAB). O PAB é um problema complexo e de grande importância, tendo em vista que grande parte do transporte mundial é feito através da via marítima. Dentro desse contexto, buscou-se contribuir no aprimoramento da logística na distribuição dos navios nos berços ao longo cais, minimizando o tempo total de serviço dos navios, diminuindo dessa maneira os gastos com espera do porto.

Para resolver o PAB, foi proposta a aplicação dos seguintes algoritmos baseados na técnica dos Sistemas Imunológicos Artificiais:

- *AISO (Artificial Immune System Optimization)*;
- *PCAISO (Parallel Coevolutive Artificial Immune System Optimization)*: Algoritmo *AISO* com a inclusão de conceitos de co-evolução;
- *PCAISO-GT (Parallel Coevolutive Artificial Immune System Optimization with Game Theory)*: Algoritmo *PCAISO* com a inclusão de conceitos de Teoria dos Jogos.

No processo de construção das metaheurísticas, mais especificamente durante a fase de testes, foram obtidos resultados expressivos para o problema teste em questão, mostrando que a inspiração nos fenômenos naturais para o desenvolvimento de novas técnicas de otimização é um universo ainda a ser bastante explorado.

As novas metaheurísticas (*PCAISO e PCAISO-GT*) apresentaram o comportamento esperado, quando foi considerada que a inclusão de conceitos de Co-evolução e de Teoria dos Jogos à técnica dos Sistemas Imunológicos Artificiais, resultasse na diminuição da vulnerabilidade dos algoritmos a ótimos locais, tendo em vista o aumento da variabilidade da população através da evolução de diferentes populações em ambientes separados (Co-evolução) e da aplicação do clássico jogo Dilema do Prisioneiro, como ambiente (Interação Social) para a realização das disputas por recursos entre as células participantes, fazendo com que a curva de evolução fosse mais expressiva.

O resultado final do algoritmo *PCAISO-GT* quando aplicado às instâncias de teste apresentadas foi bastante expressivo. Das 30 instâncias avaliadas, a nova metaheurística encontrou o resultado ótimo global em 96,66% dos casos, mostrando a qualidade do algoritmo proposto aplicado ao PAB.

A possibilidade de realizar a comparação dos resultados encontrados com outros trabalhos possibilitou uma avaliação efetiva de que a solução proposta teve um resultado final de alta qualidade. É importante observar que ainda devem ser realizados estudos mais aprofundados, no sentido de aprimorar ainda mais o comportamento apresentado pelas técnicas desenvolvidas neste trabalho.

Com o sucesso das metaheurísticas aplicadas ao PAB, pretende-se dar continuidade ao seu processo de desenvolvimento, com objetivo de construir novas técnicas de otimização híbridas e aplicar as metaheurísticas a outros problemas de alta complexidade. Sendo assim, lista-se a seguir algumas dessas possibilidades:

- Utilização de algoritmos sócio-culturais de forma a caracterizar o acúmulo de experiência da população durante o decorrer das gerações;
- Utilizar técnicas de interação política, visando à introdução de um novo comportamento de liderança entre os indivíduos da população;
- Construir um controlador Fuzzy para o ajuste automático dos parâmetros utilizados nos algoritmos e com isso criar a possibilidade de melhorar a adaptação dos indivíduos;
- Implementar e utilizar um Dilema do Prisioneiro Iterado Fuzzy conjuntamente à técnica dos sistemas imunológicos artificiais;
- Aplicar as metaheurísticas ao problema de Alocação de Berços e ao Problema de Carregamento de Navios de maneira simultânea.

Com a implementação dessas técnicas, características e novas aplicações, a expectativa é que possam ser obtidos melhores resultados para o PAB e para outros problemas não só relacionados à atividade portuária, mas também problemas do mundo real enfrentados em outras áreas de interesse.

8 REFERÊNCIAS BIBLIOGRÁFICAS

- ABUL, A. (2011). **“Imunologia Básica”**. Elsevier Brasil, 2011
- ARORA, S.; BARAK, B. **“Computational complexity: a modern approach”**. Nova York: Cambridge University Press, 2009.
- ASHFORD, N. J.; PAQUETTE, R. J.; WRIGHT, P. H. **Transportation engineering: planning and design**. 2. ed. John Wiley & Sons, Inc., 1982.
- BALLARD, D. **An Introduction to Natural Computation**. MIT Press. 1999.
- BIERWIRTH C., MEISEL F. **“Heuristics for the integration of crane productivity in the berth allocation problem”**. Transportation Research Part E 45, 2009: 196 – 209
- BIERWIRTH, C., MEISEL, F. **“A survey of berth allocation and quay crane scheduling problems in container terminals”**. European Journal of Operational Research 202 (2010) 615–627
- BORGES, P. S. S. **“A Model Of Strategy Games Based on The Paradigm of the Iterated Prisoner’s Dilemma Employng Fuzzy Sets”**. 1996. Tese (Doutorado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis.
- BROWN, G.G., LAWPHONGPANICH, S. & THURMAN, K.P. (1994). **“Optimizing ship berthing”**. Naval Research Logistics, 41, 1-15.
- BROWN, G.G., CORMICAN, K.J., LAWPHONGPANICH, S. & WIDDIS, D.B. (1997). **“Optimizing submarine berthing with a persistence incentive”**. Naval Research Logistics, 44, 301-318.
- BURNET, F. M. **Clonal Selection and After**, In Theoretical Immunology, (Eds.) G. I. Bell, A.S. Perelson & G. H. Pimbley Jr., Marcel Dekker Inc. 1978, pp. 63-85.
- BURNET, F.M., 1959. **“The Clonal Selection Theory of Acquired Immunity”**. Cambridge University Press.
- BUHRKAL, K., ZUGLIAN, S., ROPKE, S., LARSEN, J. & LUSBY, R. (2009). **“Models for the discrete berth allocation problem: a computational comparison”**. Technical Report 14/2009 - Technical University of Denmark.
- CANTU-PAZ, E.: **“A Summary of Research on Parallel Genetic Algorithms”**. Technical Report IlliGAL No. 95007, July 1995, University of Illinois at Urbana-Champaign, 1995.

- CANTU-PAZ, E., “**Efficient and Accurate Parallel Genetic Algorithms**”. Dordrecht, The Netherlands: Kluwer Academic Publishers Group, 2000.
- CCE. Comissão das Comunidades Europeias. 2009. “**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.
- CHANG, H. S., LEE, H. G., FU, M. C., MARCUS, S. I. (2005) “**Evolutionary policy iteration for solving Markov decision processes**”. IEEE T Automat Contr 50(11):1804–1808
- CHEONG, C.Y., TAN, K.C., LIU, D.K. & LIN, C.J. (2008). “**Multi-objective and prioritized berth allocation in container ports**”. Annals of Operations Research, 180(1), 63-103.
- CNT (2006). Em <<http://www.sistemacnt.org.br/pesquisacntaquaviaria/2006/glossario.htm>>. Acesso em 30/04/2013
- COELLO, C. A. C.; CORTES, N. C. 2002. “**A Parallel Implementation of an Artificial Immune System to Handle Constraints in Genetic Algorithms: Preliminary Results**”. Evolutionary Computation. CEC 2002. Proceedings of 2002 Congress.
- COOKE, D and HUNT, J. (1995). “**Recognising promoter sequences using an Artificial Immune System**”. 89-97 *Proceedings of Intelligent Systems in Molecular Biology*. AAAI Press.
- CORDEAU, J.F; LAPORTE, G.; LEGATO, P.; MOCCIA, P. (2005). “**Models and tabu search heuristics for the berth allocation problem**”. Transportation Science, 39, 526-538.
- CUTELLO, V. and NICOSIA, G. (2002). “**An Immunological Approach to Combinatorial Optimization Problems**” *Lecture Notes in Computer Science*, Springer vol. 2527, pp. 361–370.
- CUTELLO V.; NARZISI G., NICOSIA, G., and PAVONE, M., 2005. “**An Immunological Algorithm for Global Numerical Optimization**”. Artificial Evolution: 7th Int. Conference, Evolution Artificielle, EA 2005, October 26-28, Lille, France, Springer, LNCS 3871:284-295
- CUTELLO V., NICOSIA G. and PAVONE M., 2006. “**Real Coded Clonal Selection Algorithm for Unconstrained Global Optimization using a Hybrid Inversely Proportional Hyper mutation Operator**”. The 21st Annual ACM Symposium on

- Applied Computing, SAC 2006, April 23 -27, Dijon, France. ACM Press, vol. 2, pp. 950-954.
- DABROWSKI, J. D., and Kubale, M. (2008) **“Computer Experiments With A Parallel Clonal Selection Algorithm For The Graph Coloring Problem”**. *Proceedings of IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2008* p. 1-6.
- DAI A., TRENBERTH K. E., QIAN T. (2004) **“A Global Dataset of Palmer Drought Severity Index for 1870–2002: Relationship with Soil Moisture and Effects of Surface Warming”**. National Center for Atmospheric Research, Boulder, Colorado
- DALE, N.; LEWIS, J. 2010. **“Computer science illuminated”**. Jones and Bartlett Publishers.
- DASGUPTA, D. (1999). **“Artificial Immune Systems and Their Applications”**, Springer-Verlag, Inc. Berlin.
- DE CASTRO, L. N.; VON ZUBEN, F. J. (2002). **“Learning and Optimization Using the Clonal Selection Principle”**. *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems (IEEE)* 6(3): 239–251.
- DE CASTRO, L. N.; TIMMIS, J. (2002). **“Artificial Immune Systems: A New Computational Intelligence Approach”**. Springer. pp. 57–58.
- DE CASTRO, L. N. **Engenharia Imunológica: Desenvolvimento e Aplicação de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais**. Tese de Doutorado, DCA – UNICAMP, 2001.
- ERLICH, P.R.; RAVEN, P.H. **“Butterflies and plants: a study in coevolution”**. *Evolution*, v. 18, p. 586-608, 1964.
- FARMER, J.D.; PACKARD, N.; and PERELSON, A. (1986) **“The immune system, adaptation and machine learning”**, *Physica D*, vol. 2, pp. 187–204
- FERRY, F.; OLIVEIRA, C. E. J. G. 2006. **“EAISO-GT: Uma nova meta-heurística evolucionária de otimização avançada baseada na técnica dos Sistemas Imunológicos Artificiais e na Teoria dos Jogos”**. CESUPA, ACET
- FORREST, S.; PERELSON, A.S.; ALLEN, L.; CHERUKURI, R. (1994). **“Self-nonsel self discrimination in a computer”**. *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*. Los Alamitos, CA. pp. 202–212.
- GARRET S., 2004. **“Parameter-Free Adaptive Clonal Selection”**. Congress on Evolutionary Computation (CEC2004), 19-23 June, Volume: 1, pp.: 1052- 1058.

- GAREY, R. M., JOHNSON, S. D. (1979). “**Computers and Intractability: a guide to the theory of NP-Completeness**” W. H. Freeman and Co. pp. x+338
- GIALLOMBARDO, G., MOCCIA, L., SALANI, M. & VACCA, I. (2010). “**Modeling and solving the tactical berth allocation problem**”. *Transportation Research Part B*, 44, 232-245.
- GONG, M., ZHANG, L., JIAO, L. and Ma W., 2007. “**Differential Immune Clonal Selection Algorithm**”. *Proceedings of 2007 International Symposium on Intelligent Signal Processing and Communication Systems* Nov.28-Dec.1, Xiamen, China.
- GULDOGAN, E.U.; BULUT, O.; TASGETIREN, M. F. (2012) “**A Dynamic Berth Allocation Problem with Priority Considerations under Stochastic Nature**”. Springer , Heidelberg
- HANSEN, P., OGUZ, C. (2003). “**A note on formulations of static and dynamic berth allocation problems**”. *Les Cahiers du GERAD* 30, 1–17.
- HOLLAND, J. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, Ann Arbor, 1975.
- IMAI, A.; NISHIMURA, E.; PAPADIMITRIOU, S. 2001. “**The dynamic berth allocation problem for a container port**”. *Transportation Research Part B*, v. 37, p. 401-417.
- IMAI, A., NISHIMURA, E. PAPADIMITRIOU, S. (2003). “**Berth allocation with service priority**”. *Transportation Research Part B*, 37, 437-457.
- IMAI, A., SUN, X., NISHIMURA, E., PAPADIMITRIOU, S. (2005). “**Berth allocation in a container port: using a continuous location space approach**”. *Transportation Research Part B* 39 (3), 199–221.
- JIN, N., RAHMAT-SAMII, Y. “**Parallel particle swarm optimization and finite-difference time-domain (PSO/FDTD) algorithm for multiband and wide-band patch antenna designs**”. *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 11, pp. 3459–3468, 2005.
- KELSEY, J. and TIMMIS J.; “**Immune Inspired Somatic Contiguous Hypermutation form Function Optimization**”, in *Proceedings of Genetic and Evolutionary Computation Conference (GECCO) 2003*, Cantu-Paz et al. (Eds.), *Lecture Notes in Computer Science* 2727 (2003) 207-218

- LI, X., LUO, W., YAO, X. (2007). "**Evolutionary Learning and Optimization**", Connection Science, Volume 19, Issue 4, December 2007, Taylor & Francis, London, UK.
- LIM, A., 1998. "**The berth planning problem**". Operations Research Letters 22 (2), 105–110.
- 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): 1021-1028 (2009)
- LOPES, A. T., VINICIUS, M. L. S., MAURI, G. R. (2011). "**Grasp com path relinking para o problema de alocação de berços**". PODes.
- RUOCHEN L., JIAO L., DU H. "**Immune Clonal Strategy Algorithm**". Computational Intelligence and Multimedia Applications, 2003. ICCIMA 2003. Proceedings. Fifth International Conference on 27-30 Sept. 2003
- RUOCHEN L., JIAO L., DU H. 2004. "**Adaptive Immune Clonal Strategy Algorithm**". 7th International Conference on Signal Processing (ICSP'04), 31 Aug.-4 Sept., Volume: 2, pp: 1554- 1557.
- MAURI, G. R. "**Novas heurísticas para o problema de escalonamento de tripulações**". Dissertação de Mestrado em Computação Aplicada - Instituto Nacional de Pesquisas Espaciais (INPE), orientada pelo Prof. Luiz Antonio Nogueira Lorena, defendida em 23 de fevereiro de 2005.
- MAURI, G.R. (2008). "**Novas abordagens para representação e obtenção de limitantes e soluções para alguns problemas de otimização combinatória**". Tese (Doutorado em Computação Aplicada), Instituto Nacional de Pesquisas Espaciais (INPE).
- MAURI, G.R., OLIVEIRA, A.C.M. & LORENA, L.A.N. (2008a). "**Heurística baseada no simulated annealing aplicada ao problema de alocação de berços**". GEPROS - Gestão da Produção, Operações e Sistemas, 1, 113-127.
- MAURI, G.R., OLIVEIRA, A.C.M. & LORENA, L.A.N. (2008b). "**A hybrid column generation approach for the berth allocation problem**". Lecture Notes in Computer Science, 4972, 110-122.
- NOWOSTAWSKI, M. Y P. RICARDO. "**Parallel Genetic Algorithm Taxonomy**". Third International Conference of Knowledge-Based Intelligent Information Engineering Systems, 1999.

- OLIVEIRA, C. T. 2000. **“Modernização dos portos”**. 3. ed. São Paulo: Aduaneira.
- OLIVEIRA, R.M., MAURI, G.R. & LORENA, L.A.N. (2010). **“Clustering search aplicado ao problema de alocação de berços”**. In: Anais do XLII SBPO – Simpósio Brasileiro de Pesquisa Operacional.
- PARK, Y.M., K.H. KIM. (2003) **“A Scheduling Method for Berth and Quay Cranes”**. OR Spectrum 25, 1-23.
- PEIXOTO, J. B. 1977. **“Os transportes no atual desenvolvimento do Brasil”**. Rio de Janeiro: Biblioteca do Exército.
- POTTER, M. A.; JONG, K. A. (2000) **“Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents”**. Evolutionary Computation, Vol. 8, p. 1-29.
- RAPOPORT, A. (1960) **“Fights, Games, and Debates”**. University of Michigan Press, 1960, pp. 400.
- RAPOPORT, A; CHAMMAH, A, M. (1965) **“Prisoner's Dilemma: A Study in Conflict and Cooperation”**. University of Michigan Press, 1965.
- SILVA, G; COCCO, G. 1999. **“Cidade e portos: os espaços da globalização”**. Rio de Janeiro: DP&A.
- STALLINGS, W. 2009. **“Computer organization and architecture: designing for performance”**. Prentice Hall.
- SILBERSCHATZ, A. et al. **“Sistemas Operacionais: conceitos e aplicações”**. Campus, 2001. (Cap. 5).
- TANENBAUM, A. S. 2007. **“Organização estruturada de computadores”**. 2 ed. Prentice Hall.
- TEIXEIRA, O. N. **“Proposta de um novo algoritmo genético baseado na teoria dos jogos”**. Programa de Pós-Graduação em Engenharia Elétrica, UFPA. 2005.
- TIZARD, I. R., 1995. **“Immunology An Introduction”**, Saunders College Publishing, 4 ed.
- THURMAN, K.P. (1989). **“Optimal ship berthing plans”**. Dissertation (Masters of Science in Operations Research), Naval Postgraduate School - USA.
- UNITED NATIONS, UNCTAD. 2009. **“Review of maritime transport 2009”**. United Nations Publications, United Nations Conference on Trade and Development.
- VENTER, G. Sobieszczanski-Sobieski, J. A. (2005) **“Parallel Particle Swarm Optimization Algorithm Accelerated by Asynchronous Evaluations”**. NASA.

- YU Y.; HOU C., 2004. **“A Clonal Selection Algorithm By Using Learning Operator”**. Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 Aug.
- ZUO X. and LI S., 2003. **“The Chaos Artificial Immune Algorithm and Its Application to RBF Neuro-Fuzzy Controller Design”**. IEEE International Conference on Systems, Man and Cybernetics, Volume 3, Issue , 5-8 Oct., Page(s): 2809 - 2814.