

UFES – Universidade Federal do Espírito Santo
CT – Centro Tecnológico
PPGI – Programa de Pós Graduação em Informática

**SISTEMA IMUNE ARTIFICIAL PARA O PROBLEMA
DE ESCALONAMENTO *JOB SHOP***

SILDENIR ALVES RIBEIRO

Vitória – ES
Novembro – 2006

UFES – Universidade Federal do Espírito Santo
CT – Centro Tecnológico
PPGI – Programa de Pós Graduação em Informática

**SISTEMA IMUNE ARTIFICIAL PARA O PROBLEMA DE
ESCALONAMENTO *JOB SHOP***

SILDENIR ALVES RIBEIRO

*Dissertação de Mestrado submetida ao PPGI – Programa de
Pós Graduação em Informática, da UFES – Universidade
Federal do Espírito Santo como parte das exigências do curso
de Pós-Graduação Stricto Senso em Informática, para obtenção
do título de Mestre em Informática.*

**ORIENTADORES: ARLINDO GOMES DE ALVARENGA
HANNU TAPIO AHONEN**

Vitória – ES
Novembro de 2006

FICHA CATALOGRÁFICA

RIBEIRO, Sildenir Alves, 1970.

Sistemas Imunes Artificiais para o Problema de Escalonamento *Job Shop*. [Vitória] 2006.

xv, 107 p., 29,7 cm (UFES, M. Sc., Informática, 2006).

Dissertação, Universidade Federal do Espírito Santo, PPGI

Biblioteca Central – UFES.

I. Otimização

II. PPGI/CT/UFES

III. Sistema Imune Artificial para o Problema de Escalonamento *Job Shop*

SISTEMA IMUNE ARTIFICIAL PARA O PROBLEMA DE ESCALONAMENTO *JOB SHOP*

SILDENIR ALVES RIBEIRO

Dissertação submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Informática - Otimização.

Julgada e aprovada pelos professores membros da banca examinadora em 29 de Novembro de 2006.

Prof. D.Sc. Arlindo Gomes de Alvarenga
(PPGI/CT/UFES)
Orientador

Prof. PhD. Hannu Tapio Ahonen
(PPGI/CT/UFES)
Orientador

D.Sc. Attílio Provedel
(PPGI/CT/UFES)
Membro Interno

D.Sc. Samuel Vieira Conceição
(DEP/UFMG)
Membro Externo

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Vitória - ES, Novembro de 2006.

Aos meus pais, Joaquim Ribeiro de Carvalho e Oracy Alves Ribeiro, e a toda minha família e amigos pela força e pelo carinho a mim demonstrado em todos os momentos. À minha esposa, Viviani de Moraes Freitas Ribeiro, pela paciência, pelo carinho e pelo apoio irrestrito e incondicional.

AGRADECIMENTOS

Agradeço a *Deus* pela luz e por ter me mostrado o caminho nos momentos mais difíceis. Agradeço a todos os professores do PPGI/CT/UFES que colaboraram com a realização deste projeto, transmitindo-me os seus conhecimentos e experiências. Agradecimento especial aos professores ***Arlindo Gomes de Alvarenga*** e ***Hannu Tapio Ahonen*** pela orientação, pela ajuda e pelo apoio incondicional. Agradeço a minha família e amigos pelo crédito e pelo incentivo, e a todos que, de forma direta ou indireta contribuíram para a realização deste projeto. Agradeço também, aos colegas da turma 2004/01 do mestrado em informática da UFES, pela troca de conhecimentos, essenciais para superar as dificuldades encontradas no decorrer do curso.

“A pesquisa científica é baseada na idéia de que tudo que acontece é determinado pelas leis da natureza; da mesma forma, a ação humana”.

Albert Einstein: *“O Lado Humano”*
Por: Helen Dukas & Banesh Hoffmann.

RESUMO

Este trabalho apresenta um Sistema Imune Artificial (SIA) para tratar problemas de escalonamento. O Sistema Imunológico Artificial desenvolvido neste projeto baseia-se na estrutura arquitetura e funcionamento dos Sistemas Imunes Biológicos ou Naturais. O uso de Algoritmo Genético (AG) fez-se necessário para gerar os indivíduos a serem escalonados, representando os antígenos e anticorpos do SIA. Cada indivíduo gerado pelo AG representa um conjunto de tarefas processadas em um conjunto de máquinas. Os indivíduos são avaliados por uma função de aptidão que representa o processo de seleção natural. A evolução dos indivíduos e consequentemente das populações são obtidas aplicando-se os operadores genéticos de *crossover* e mutação. As tarefas e as máquinas, utilizadas para o escalonamento, representa o problema de *Job Shop Scheduling* (JSS). Ao problema, foram aplicados alguns testes clássicos da literatura, onde se verificou a viabilidade dos SIA para tratamento de problemas de escalonamento. Ainda com os testes, pode-se observar o comportamento do sistema durante toda a execução, possibilitando assim, uma análise criteriosa das funcionalidades do sistema e dos resultados gerados pela massa de teste, observados durante um período de tempo. A representação dos sistemas imunológicos naturais através de algoritmos computacionais tem inspirado pesquisadores de todo o mundo, a motivação é que os sistemas imunológicos possuem características de paralelismo adaptabilidade e aprendizagem, além da possibilidade de serem aplicados em diversos problemas das mais diversas áreas, devido sua portabilidade.

Palavras Chave: *Algoritmo Genético, Job Shop Scheduling, Otimização, Sistemas Imunes Artificiais, Sistemas Imunes Biológico.*

ABSTRACT

This work presents an Artificial Immune System (AIS) to deal with problems scheduling. The Artificial Immunologic System developed in this project was based on the structure, architecture and functioning of the Biological or Natural Immune Systems. The use of Genetic Algorithm (GA) became necessary to represent the antibodies and antigens of the AIS. Each individual generated for the GA represented a processed task set library in a set of machines. The evaluation of each individual was given by a fitness function that represents the process of natural selection. The evolution of the individuals, and population as a consequence was obtained by applying the genetic operators of crossover e mutation. The machines and the tasks used for the scheduling represent the problem of Job Shop Scheduling (JSS). Some classic tests of the literature where applied to the problem in order to verify the viability of the AIS on the treatment of task of scheduling problems. Those tests also demonstrated the system's behavior its entire execution, therefore, allowing for a detailed analysis of the system's functionalities sets for certain time period. The representation of the natural immunologic systems through computational algorithms inspires from all over world researchers. The motivation is that the immunologic systems possess parallelism characteristics adaptability and learning, which can be applied in several problems found in many areas, had its portability.

Key Words: *Artificial Immune Systems, Biological Immune Systems, Genetic Algorithm, Job Shop Scheduling, Optimization.*

LISTA DE FIGURAS

Figura 2.2.1.1: Mecanismo da Resposta Imune -----	25
Figura 2.2.5.2: Célula B -----	29
Figura 2.2.5.3: Célula T -----	29
Figura 2.2.6.4: Conceitos Básicos da Rede Imunológica -----	31
Figura 2.3.1.5: Diagrama de Níveis para Desenvolvimento dos SIA -----	33
Figura 2.3.3.6: Espaço de Formas S em um Volume V -----	34
Figura 2.3.3.7: Regiões de Complementaridade -----	35
Figura 2.3.3.8: Gráfico de Afinidade Definido Sobre um Espaço de Formas -----	36
Figura 2.3.5.9: Representação do Algoritmo de Seleção Positiva -----	38
Figura 2.3.5.10: Algoritmo de Seleção Negativa -----	39
Figura 3.1.0.11: Representação de Job Shop Scheduling -----	43
Figura 3.3.2.12: Geração de uma Nova População. -----	46
Figura 3.4.0.13: Operadores Genéticos. -----	47
Figura 3.4.2.14: Operador de Crossover de um Ponto. -----	49
Figura 3.4.2.15: Crossover Uniforme. -----	50
Figura 3.4.2.16: Crossover de Dois Pontos. -----	50
Figura 3.4.2.17: Order Based Crossover. -----	51
Figura 3.4.2.18: Overlap Crossover -----	51
Figura 3.4.2.19: Mutação -----	52
Figura 3.5.1.20: Fluxograma Representativo do AG -----	53
Figura 3.5.1.21: Pseudocódigo do AG desenvolvido: -----	54
Figura 4.1.0.22: Codificação Inteira das Moléculas e Função de Matching das Cadeias -----	61
Figura 4.4.0.23: Universo de Antígenos -----	64
Figura 3.4.0.24: Arquitetura do SIA Desenvolvido -----	65
Figura 4.4.3.25: Alinhamento e Cálculo de Match -----	68
Figura 4.5.0.26: Germline. -----	70
Figura 4.5.1.27: Recombinação Somática -----	71

LISTA DE TABELAS

Tabela 3.1.0.1: Um Job Shop Clássico 3X3 -----	43
Tabela 3.2.0.2: Representação de Job's Escalonados-----	44
Tabela 3.2.0.3: Exemplo de Criação de Fenótipos -----	45
Tabela 3.5.4.4: Nomenclatura, Parâmetros e Valores Usados no AG -----	57
Tabela 3.5.5.5 Formato de Entrada de Dados no AG-----	58
Tabela 4.1.0.6: Relação Entre o Problema de Scheduling e o SAI-----	60
Tabela 4.2.0.7: Comparativo Entre Sistema Imune Natural e Sistema Imune Artificial -----	62
Tabela 4.4.1.8: Exemplo de uma Entrada (Instância) 4X4-----	66
Tabela 4.4.1.9: Instância com Tempos Modificados (4X4)-----	66
Tabela 5.1.1.10: Parâmetros Analisados -----	74
Tabela 5.1.2.11: Benchmarks e Instâncias Utilizadas -----	74
Tabela 5.4.1.12: Resultados dos Testes Aplicados à Instância FT06-----	77
Tabela 5.4.2.13: Resultados dos Testes Aplicados à Instância FT10-----	77
Tabela 5.4.3.14: Resultados dos Testes Aplicados à Instância ABZ05 -----	78
Tabela 5.4.4.15: Resultados dos Testes Aplicados à Instância TA02 -----	79
Tabela 5.4.5.16: Resultados dos Testes Aplicados à Instância LA02 -----	79
Tabela 5.4.6.17: Análise dos Resultados – Testes com Todos Operadores-----	80
Tabela 5.4.6.18: Análise dos Resultados – Testes com um Operador -----	80

LISTA DE ABREVIATURAS

Ab	<i>Antibody</i> ou Anticorpo
Ag	<i>Antigen</i> ou Antígeno
AG	Algoritmo Genético
APC	<i>Antigen Presenting Cells</i>
BCR	<i>B Cells Receptors</i>
CLARISA	<i>Classification via Artificial Immune System Analogy</i>
CLONALG	<i>Clonal Selection Algorithm</i>
CX	<i>Cycle Crossover</i>
DDR	<i>Double Data Rate</i>
GB	<i>Giga Byte</i>
IA	Inteligência Artificial
IC	Inteligência Computacional
IDS	<i>Intrusion Detect System</i>
IgG	Imunoglobina G
IgM	Imunoglobina M
IL	Interleucina
INF	<i>Interference Factor</i>
JSS	<i>Job Shop Scheduling</i>
J2SE	<i>Java 2 Standard Edition</i>
MB	<i>Mega Byte</i>
MHC	<i>Major Histocompatibility Complex</i>
MHz	<i>Mega Hertz</i>
NK Cells	<i>Natural Killer Cells</i>
NP	Não Polinomial
OX	<i>Order Based Crossover</i>
PMX	<i>Partial Matched Crossover</i>
PRAIS	<i>Pattern Recognizing Artificial Immune System</i>

RAM	<i>Random Access Memory</i>
SBD	Sistemas de Banco de Dados
SDRAM	<i>Synchronous Dynamic Read Access Memory</i>
SIA	Sistema Imune Artificial
TC	T Citotóxicos
TCR	<i>T Cells Receptors</i>
TH	<i>T Helper</i>
TNF	<i>Tumoral Necroses Factor</i>
TS	T Supressoras
WWW	<i>Wide World Web</i>
2PtX	<i>Two Point Crossover</i>

SUMÁRIO

1 INTRODUÇÃO	16
1.1 Sistemas Imunes Artificiais	16
1.2 Aplicações dos SIA	17
1.2.1 Otimização	17
1.2.2 Inteligência Computacional ou Artificial	17
1.2.3 Redes de Computadores	18
1.2.4 Sistemas de Banco de Dados	18
1.3 Objeto de Estudo	18
1.4 Notação e Formalização	19
1.5 Estrutura do Texto	19
1.5.1 Capítulo 2: Sistemas Imunes Artificiais - Conceitos Gerais	19
1.5.2 Capítulo 3: O Problema de Escalonamento <i>Job Shop</i> .	19
1.5.3 Capítulo 4: Tratamento Imunológico para o Problema de Escalonamento.	20
1.5.4 Capítulo 5: Testes e Resultados	20
1.5.5 Capítulo 6: Considerações Finais	20
1.5.6 Referências Bibliográficas	20
1.5.7 Anexo	20
1.5.8 Apêndice: Glossário	21
2 SISTEMAS IMUNES ARTIFICIAIS (SIA): CONCEITOS GERAIS.	22
2.1 O que é um Sistema Imune	22
2.2 O Sistema Imunológico Biológico	23
2.2.1 Sistemas Imunes Biológicos: Características, Conceitos e Componentes.	24
2.2.2 Características do Sistema Imune Adaptativo	25
2.2.3 Principais Características das Respostas Imunológicas	26
2.2.4 Funcionamento do Sistema Imune Biológico ou Natural	27
2.2.5 Reconhecimento de Padrões e Produção de Anticorpos	28
2.2.6 O Princípio da Seleção Clonal e as Redes Imunológicas	30
2.3 Os Sistemas Imunes Artificiais	31
2.3.1 Definindo um Sistema Imune Artificial	32
2.3.2 Representação dos Componentes Imunes	33
2.3.3 Funções de Afinidade	34
2.3.4 Algoritmos, Modelos e Processos	37
2.3.5 Algoritmo de Seleção Positiva, Negativa e Clonal	38

3 O PROBLEMA DE ESCALONAMENTO <i>JOB SHOP</i>	42
3.1 Apresentação do Problema <i>Job Shop Scheduling</i>	42
3.2 Representação do Problema <i>Job Shop Scheduling</i>	44
3.3 Aplicando Algoritmos Genéticos	45
3.3.1 Representação do Cromossomo	46
3.3.2 Processo de Geração de uma População	46
3.5 Algoritmo Genético para Tratar Problemas de Escalonamento	52
3.5.1 Estrutura do AG Desenvolvido	53
3.5.2 Descrição dos Componentes do AG.	54
3.5.4 Formalização do AG: Nomenclatura, Parâmetros e Valores.	56
3.5.5 Dados de Entrada para o AG	57
3.5.6 Dados de Saída do AG	58
4 TRATAMENTO IMUNOLÓGICO PARA O PROBLEMA DE ESCALONAMENTO	59
4.1 Relação Entre um SIA e o Problema de Escalonamento	59
4.2 Critérios para Desenvolvimento de um SIA	61
4.3 Representação do Ambiente Imunológico	62
4.3.1 Construção do Ambiente Imunológico	63
4.4 O Sistema Imune Desenvolvido	63
4.4.1 Representação dos Antígenos do SIA	66
4.4.2 Representação dos Anticorpos do SIA	67
4.4.3 Avaliação da Aptidão Através do <i>Match Score</i>	67
4.4.4 Produzindo uma Resposta Imune	68
4.5 Construção do <i>Germline</i>	69
4.5.1 Mecanismo de Recombinação Somática	70
4.6 Gerando Escalonamentos	71
4.6.1 Processo de Geração de Escalonamentos	71
5 TESTES E RESULTADOS	73
5.1 Geração de Testes	73
5.1.1 Parâmetros Analisados	73
5.1.2 Cenários de Teste	74
5.2 Recursos Computacionais	75
5.2.1 Recursos de <i>Software</i>	75
5.2.2 Recursos de <i>Hardware</i>	75
5.3 Testes Aplicados	76
5.4 Resultados Obtidos	76

5.4.1 Resultados Obtidos para a Instância FT06 -----	76
5.4.2 Resultados Obtidos para a Instância FT10 -----	77
5.4.3 Resultados Obtidos para a Instância ABZ05 -----	78
5.4.4 Resultados Obtidos para a Instância TA02 -----	78
5.4.5 Resultados Obtidos para a Instância LA02 -----	79
5.4.6 Análise Comparativa dos Resultados -----	80
5.4.7 Considerações Sobre os Testes Realizados e os Resultados Obtidos -----	81
 6 CONSIDERAÇÕES FINAIS-----	83
 7 REFERÊNCIAS BIBLIOGRÁFICAS-----	85
 ANEXOS -----	88
I Interface do Sistema Imune Artificial-----	88
II Arquitetura do Algoritmo Genético-----	89
III Instâncias Utilizadas -----	90
IV LOG dos Testes Realizados com todos Operadores-----	92
V LOG dos Testes Realizados Somente com Um Operador -----	97
 APÊNDICE-----	102
Glossário -----	102

1 INTRODUÇÃO

Neste capítulo serão discutidos e apresentados os elementos principais, estudados e aplicados para a realização deste trabalho. Também será apresentada a estrutura textual desenvolvida nesta dissertação, bem como seus assuntos correlatos.

1.1 Sistemas Imunes Artificiais

Os Sistemas Imunes Artificiais (SIA) são considerados por vários pesquisadores como um novo ramo ligado à área de inteligência computacional. Embora seja uma nova linha de pesquisa, criada em 1996, existe uma grande variedade de pesquisas concluídas e em andamento sobre sistemas imunes artificiais.

Diante desta grande variedade de trabalhos relacionados a esta linha, alguns pesquisadores propuseram a criação de um modelo genérico, mas sem que fosse apresentada uma definição formal para tais modelos, não contribuindo assim, para a criação de uma linguagem genérica com um tratamento unificado de conceitos e ferramentas.

Um Sistema Imune Artificial pode ser definido como um modelo computacional baseado no sistema imunológico natural (humano), representado por um conjunto de mecanismos fundamentados no comportamento e estrutura das células, anticorpos e antígenos.

Na literatura, encontram-se várias definições para sistemas imunes artificiais, embora não exista nenhum diferencial importante em cada definição. Em seu trabalho, De Castro (2001), destaca três definições principais: 1 – Os sistemas imunológicos artificiais são metodologias de manipulação de dados, classificação, representação e raciocínio que seguem um paradigma biológico plausível: o sistema biológico humano; 2 – Um sistema imunológico artificial é um sistema computacional baseado em metáforas do sistema imunológico natural; e, 3 – Os sistemas imunológicos artificiais são compostos por metodologias inteligentes, inspiradas no sistema imunológico biológico, para a solução de problemas do mundo real.

Aliada as outras técnicas, métodos e modelos, os sistemas imunes são bastante poderosos para a solução de problemas complexos. Como por exemplo: o problema de *Job Shop Scheduling* (JSS), que será estudado neste trabalho a fim de propor uma possível solução usando os Sistemas Imunes Artificiais. O Problema de JSS é um problema de otimização classificado como complexo e de difícil solução (NP *Hard*).

1.2 Aplicações dos SIA

Os Sistemas Imunes Artificiais podem ser aplicados em diversas áreas da ciência computacional, para as mais variadas classes e tipos de problemas. Sua abrangência computacional estende-se desde os problemas combinatoriais e de otimização a redes de computadores. As seções a seguir apresentam algumas das áreas de aplicações dos SIA, destacando-se, Otimização, Inteligência Computacional, Redes de Computadores e Banco de Dados.

1.2.1 Otimização

A grande variedade de problemas e as várias possibilidades de aplicação dos SIA fazem da otimização uma potencial área. Várias pesquisas já foram desenvolvidas aplicando SIA em otimização. Na otimização, os sistemas imunes artificiais são largamente aplicados em problemas combinatoriais, de *scheduling*, modelagem computacional, computação paralela e computação distribuída. D'haeseleer *et al.*(1996) propôs um algoritmo imunológico para detecção de mudanças por aproximação, utilizando computação paralela e distribuída.

Neste trabalho, estamos interessados em analisar a viabilidade dos SIA para tratamento de problemas de escalonamento, considerando a característica dinâmica inerente nos ambientes de produção.

1.2.2 Inteligência Computacional ou Artificial

A Inteligência Computacional (IC) ou Inteligência Artificial (IA) aparece como uma grande e oportuna área para aplicação dos Sistemas Imunes Artificiais. Poucos trabalhos têm sido desenvolvidos utilizando os Sistemas Imunes Artificiais na IC. A aplicação dos SIA em IA se dá principalmente pela sua capacidade de aprendizagem, de memória e de

adaptabilidade. Na inteligência computacional as aplicações dos SIA se destacam em: redes neurais, computação evolutiva, biotecnologia e agentes inteligentes. A Robótica também se apresenta como uma das áreas potenciais para estudos e aplicações dos sistemas imunes por estar diretamente ligada a IC. As subáreas da robótica que mais se destacam são: navegação e controle autônomo, aprendizagem de máquinas e reconhecimento de padrões.

1.2.3 Redes de Computadores

Durante a pesquisa, observou-se a realização de alguns trabalhos aplicando Sistemas Imunes Artificiais em Redes de Computadores. As redes de computadores também dispõem de algumas subáreas de aplicação dos SIA. A segurança computacional, com detecção de intrusos (*IDS – Intrusion Detection System*), como visto em Kim e Bentley (1999), processamento paralelo e distribuição de recursos de rede, são as principais linhas de aplicação de SIA em Redes de Computadores.

1.2.4 Sistemas de Banco de Dados

Em seu trabalho Somaiaji *et al.* (1997), destaca aplicação dos SIA também em Sistemas de Bancos de Dados (SBD), sendo as subáreas: análise de dados, *data mining* e classificação de dados. Somaiaji *et al.* (1997), coloca ainda que os SIA podem se tornar uma ferramenta poderosa para a classificação de dados, tratando análise e estatísticas em grandes massas de dados.

1.3 Objeto de Estudo

Neste trabalho, vamos estudar a viabilidade da aplicação de Sistemas Imunes Artificiais em problemas de escalonamento de tarefas. O uso de Algoritmos Genéticos faz-se necessário para a geração e evolução dos indivíduos a serem escalonados. O problema de *Job Shop Scheduling - JSS* será o objeto central de estudo, por apresentar o processo de escalonamento de atividades, os problemas, os conflitos e as restrições no ambiente.

1.4 Notação e Formalização

Para representar alguns termos, elementos e expressões usou-se uma notação formalizada, onde adotou-se os seguintes critérios.

- Palavras ou expressões em outros idiomas são apresentadas no texto em *itálico* (ex. *Job Shop*).
- As expressões ou caracteres em *itálico* com sub-índices representam uma matriz ou um vetor (ex. Ab_i , Ab_j).
- Os caracteres em MAIÚSCULO que aparecem isoladamente representam um termo ou um elemento (ex. C, L).
- Caracteres ou expressões em **negrito** representam um conjunto (ex. **Ab**, **Ag**).
- As palavras ou cadeias de caracteres apresentadas no texto em MAIÚSCULO, representam uma sigla ou abreviatura de algum termo ou expressão (ex. SIA, AG, JSS, MHC).

1.5 Estrutura do Texto

Esta dissertação está formatada em seis capítulos, onde apresenta-se a seguinte estrutura.

1.5.1 Capítulo 2: Sistemas Imunes Artificiais - Conceitos Gerais

O Capítulo 2 apresenta os conceitos dos Sistemas Imunes Artificiais, mostrando suas características principais, bem como suas funcionalidades. O capítulo trará ainda uma abordagem conceitual, fundamentando as características e funcionalidades do Sistema Imunológico Natural ou Biológico e as primeiras pesquisas sobre os sistemas imunes artificiais embasadas no sistema imune biológico.

1.5.2 Capítulo 3: O Problema de Escalonamento *Job Shop*.

O Capítulo 3 apresenta brevemente o problema de escalonamento *job shop*. O capítulo mostrará também a aplicabilidade dos Algoritmos Genéticos (AG) para resolver o problema, destacando o AG desenvolvido, que será utilizado pelo SIA.

1.5.3 Capítulo 4: Tratamento Imunológico para o Problema de Escalonamento.

O objetivo do Capítulo 4 é apresentar o tratamento imunológico para o problema, descrever o ambiente imunológico bem como sua estrutura e componentes. O capítulo mostrará ainda a aplicação do PRAIS¹ e do CLARISA², desenvolvido por Hart e Ross (1999a, 1999b) no sistema imune desenvolvido. Estes sistemas foram largamente utilizados como base para desenvolvimento do SIA apresentado neste trabalho.

1.5.4 Capítulo 5: Testes e Resultados

No Capítulo 5 serão apresentados os testes numéricos realizados e seus respectivos resultados. O capítulo irá apresentar ainda a descrição dos critérios adotados para a aplicação dos testes.

1.5.5 Capítulo 6: Considerações Finais

O Capítulo 6 apresenta as considerações finais sobre o trabalho desenvolvido. Ainda neste capítulo, apresenta-se a experiência adquirida pelo autor e a motivação para trabalhos futuros na área de otimização e áreas correlatas, aplicando os conceitos e paradigmas dos algoritmos evolucionários, problemas de *scheduling* e os sistemas imunológicos artificiais.

1.5.6 Capítulo 7: Referências Bibliográficas

Nesta seção serão apresentadas as fontes consultadas nesta pesquisa. Os textos referenciados foram fundamentais para a realização deste projeto e espera-se que também sirva de fontes para os leitores que desejarem realizar trabalhos nesta linha de pesquisa.

1.5.7 Anexo

O Anexo deste trabalho é composto pela documentação desenvolvida durante a implementação do sistema, provendo assim, uma fonte de pesquisa mais completa para os leitores. Ao anexo inclui os seguintes documentos: a interface, a arquitetura completa do

¹ PRAIS: *Pattern Recognizing Artificial Immune System*

² CLARISA: *CLassification via ARtificial Immune System Analogy*

algoritmo genético, as instâncias utilizadas para alimentar o sistema e os resultados completo dos experimentos.

1.5.8 Apêndice: Glossário

O apêndice traz um glossário apresentando uma descrição rápida de termos poucos comuns utilizados nesta dissertação, destacando-se principalmente terminologias biológicas, computacional, matemática e algorítmica.

2 SISTEMAS IMUNES ARTIFICIAIS (SIA): Conceitos Gerais.

Este Capítulo irá fundamentar os Sistemas Imunes Artificiais, apresentando seus conceitos e características principais, além de suas funcionalidades. O Capítulo trará ainda as primeiras pesquisas sobre os Sistemas Imunes Artificiais e a engenharia imunológica, onde são apresentados os componentes dos sistemas imunes.

Antes da abordagem direta dos Sistemas Imunes Artificiais, faz-se necessário um estudo sobre o comportamento e funcionamento do sistema imunológico biológico. Embora as características imunológicas dos seres vivos (reino animal), de uma forma geral, sejam bastante semelhantes, aqui será estudado e apresentado o sistema imunológico humano, que é a base de sustentação para os estudos dos Sistemas Imunes Artificiais, considerando que os SIA comportam-se basicamente como os modelos do sistema imunológico natural.

2.1 O que é um Sistema Imune

A definição de um sistema imune é muito simples, tão simples quanto a sua função para com o corpo humano. No entanto, entender a funcionalidade e representar as suas características e dos seus agentes funcionais, requer uma análise complexa e dispendiosa.

De forma geral, pode-se definir o sistema imunológico como uma grande biblioteca de informações vitais para defesa do organismo, contra diversas toxinas e agentes patogênicos. Existe uma série de definições na literatura, das quais pode-se destacar algumas, embora todas apresentem o mesmo fim.

“O sistema imune é um complexo sistema de componentes celulares e moleculares, tendo como função primária distinguir os agentes próprios dos não próprios e defender o organismo contra substâncias estranhas.” (Dorland’s Illustrated Medical Dictionary, 2003).

“O sistema imune é um sistema cognitivo, cuja regra principal e primária é prover a manutenção do corpo.” (Cohen, 2002).

“O sistema imune é uma seleção evolucionária de uma consequência das duas primeiras definições, tendo como função primordial prover uma comunicação e um caminho intercelular ideal.” (Stewart, 2003).

Um sistema imune pode ser definido ainda como um composto de estruturas e componentes responsáveis pela proteção do organismo.

2.2 O Sistema Imunológico Biológico

O sistema imunológico biológico é definido por Alonso e Rubio (2005) como: “estado de resistência natural ou adquirido que certos indivíduos ou espécies possuem frente a determinadas ações patogênicas de microorganismos ou substâncias estranhas”.

O sistema imune biológico é um sistema adaptável, altamente paralelo e distribuído. Usa a aprendizagem, a memória, e a recuperação associativa para resolver tarefas de reconhecimento e de classificação (Dasgupta, 1998). Em outras palavras, o sistema imune biológico ou natural, usa a aprendizagem para realizar testes padrões, gravando estes testes em bibliotecas (memória), que são previamente visitados antes da realização de algum tipo de tarefa, e assim, usando as combinações das informações contidas nestas bibliotecas para construir detectores eficientes.

Os sistemas imunológicos podem ser considerados como sofisticados sistemas de processamento de informações, que podem fornecer informações inspiradas em seu funcionamento/modelo para serem aplicadas na ciência computacional, a qual procura uma representação do modelo biológico para a resolução de problemas. Os sistemas de processamento de informações baseados no sistema biológico podem ser classificados, segundo Dasgupta (1998) em: 1 – Sistema Nervoso Cerebral Neural, representados pelas redes neurais (*neural networks*), 2 – Sistemas Genéticos, que são representados pelos algoritmos evolucionários e, 3 – Sistemas Imunológicos, objeto de estudo dos sistemas imunes artificiais. Entre estes, o sistema nervoso e os sistemas genéticos foram aplicados extensamente em vários campos da computação. Já os sistemas imunes, embora sua vasta área de aplicação, têm sido pouco explorados e poucas aplicações foram desenvolvidas até agora. O motivo principal é que, embora existam vários trabalhos publicados sobre o assunto, a maioria destes trabalhos foram pesquisados e desenvolvidos pelos pesquisadores pioneiros, que ainda continuam a trabalhar nesta linha, mas que restringiram suas pesquisas em

aprimoramentos de trabalhos anteriores, não propondo assim grandes variações de pesquisas com a aplicação dos sistemas imunológicos.

Outro motivo, como já foi mencionado, é que os sistemas imunes é uma nova linha de pesquisa, a qual teve início em meados dos anos 90, e possui várias áreas de aplicação. No entanto, mesmo sendo todas largamente estudadas, há uma dificuldade de criação de um modelo ou uma representação padrão, já que, embora a base seja a mesma, existe uma tendência dos pesquisadores em aplicar seus esforços de acordo com a sua linha de pesquisa. Assim, a maior parte dos estudos são direcionados e não diversificados, de acordo com a abrangência das áreas de aplicação.

2.2.1 Sistemas Imunes Biológicos: Características, Conceitos e Componentes.

O sistema imune natural é um sistema muito complexo com diversos mecanismos para a defesa contra os organismos patogênicos. A finalidade principal do sistema imune é reconhecer especificamente e eliminar seletivamente os invasores externos por um processo denominado resposta imune. A resposta imune pode ser natural ou específica e são produzidas pelos mecanismos de imunidade inata (natural) ou adquirida.

Os sistemas imunes são classificados como: 1 - Sistema Imune Inato ou Natural; e 2 - Sistema Imune Específico ou Adquirido (Barraza, 2004).

A imunidade natural consiste em um conjunto de mecanismos que defende o corpo dos microorganismos a partir das barreiras físicas como: a pele, as mucosas, as secreções e os fluidos corporais. Quando os corpos estranhos conseguem ultrapassar essas barreiras, outros mecanismos de defesas são ativados para a eliminação destes corpos. Estão presentes neste mecanismo, as células fagocíticas e as eusínófilas presente nos tecidos, e as células agressoras naturais, ou Células NK (*Natural Killer Cells*) que estão presentes no sangue (Alonso e Rubio, 2005). A resposta inata não discrimina as diferentes substâncias estranhas e não cria memória imunológica, embora seja eficiente para alguns tipos de defesas, não é suficiente para uma defesa eficaz do organismo. Assim, para se ter um sistema de defesa mais eficiente, outros mecanismos de defesas, como a imunidade adquirida ou específica, são desenvolvidos. A resposta imune específica é um sistema integrado de defesa em que participam várias células e moléculas, que atuam cooperativamente, sendo os linfócitos, as citosinas e os anticorpos as mais importantes. Este sistema é conhecido como sistema imune adaptativo. As características da resposta imune adaptativa são: especificidade e memória.

A imunidade específica apresenta duas propriedades importantes, segundo Alonso e Rubio (2005):

1. Capacidade de recordar cada encontro com o antígeno, de tal maneira que os encontros posteriores desenvolvam uma resposta amplificada.
2. Aumenta os mecanismos protetores da imunidade natural, fazendo com que os microorganismos sejam eliminados mais facilmente em seu lugar de entrada.

Apesar de existir claras diferenças entre os componentes da imunidade natural e da específica, como observado na figura 2.2.1.1, na maioria dos casos, existe uma colaboração dos dois mecanismos, obtendo assim, uma maior proteção do corpo.

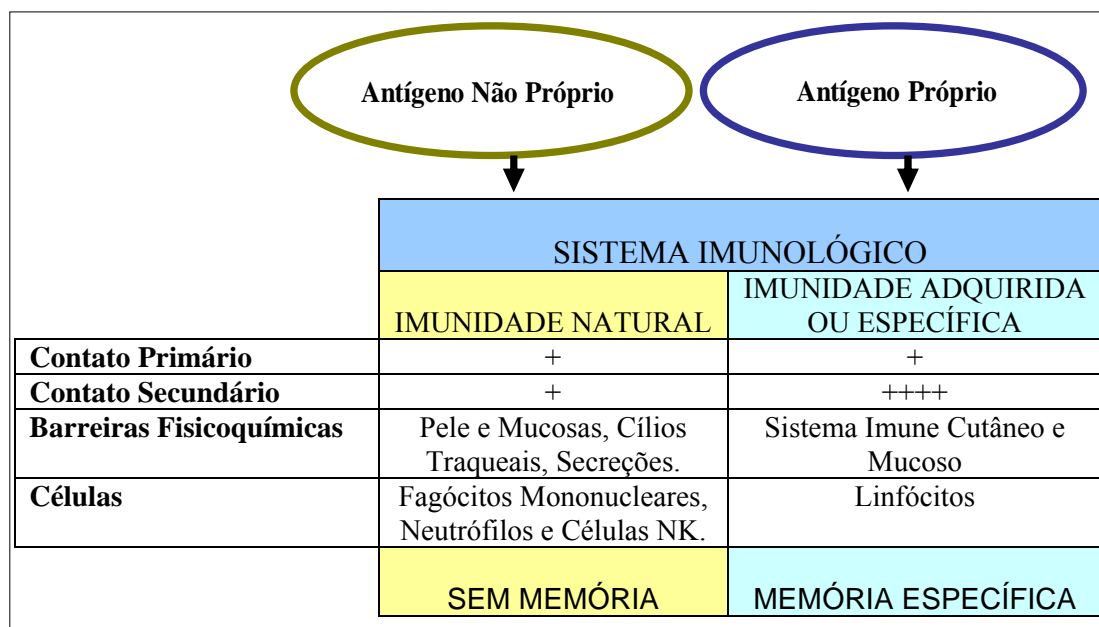


Figura 2.2.1.1: Mecanismo da Resposta Imune (adaptado de Alonso e Rubio, 2005). O sistema imune discrimina entre o próprio e o não próprio, e reagem contra as moléculas não próprias (antígenos estranhos). Após o contato com o antígeno, se é a primeira vez, obtém-se uma fraca resposta específica ou não específica (+), porém, se o antígeno persiste ou o sistema imune o encontra pela segunda vez, é produzida uma resposta específica para este antígeno muito mais intenso (++++), combatendo assim o corpo estranho.

2.2.2 Características do Sistema Imune Adaptativo

O sistema imune adaptativo é responsável por distinguir as moléculas dentro do corpo e categorizá-las como próprias (*self*) ou não próprias (*non self*) (Dasgupta 1999a).

As moléculas próprias são as moléculas do corpo, já as não próprias, são as moléculas adquiridas ou contraídas. As moléculas não próprias são categorizadas para induzir um tipo apropriado de mecanismo defensivo ou uma resposta imune, que pode ser específica e não específica.

As respostas imunológicas específicas são estimuladas quando um indivíduo se expõe a um antígeno estranho e podem ser diferenciadas segundo Alonso e Rubio (2005) em: 1 – Imunidade “*humoral*” ou *Linfática*: que pode ser transferida por meio de plasma ou soro; e 2 – Imunidade celular: que podem ser transferidas por moléculas do sangue, do timo, do baço, dos gânglios linfáticos, etc.

O principal componente da imunidade celular são os linfócitos, que atuam principalmente frente a bactérias e vírus, e são essenciais para produzir a resposta imunológica.

Os linfócitos apresentam-se em diferentes subgrupos, que diferem em suas funções e produtos protéicos que podem distinguir-se fenotipicamente, onde todos parecem morfologicamente similares. As duas principais classes de linfócitos são os linfócitos B, que são as únicas células capazes de produzir anticorpos e os linfócitos T. Os linfócitos T são responsáveis pela detecção e combate dos antígenos estranhos e podem ser divididos em três subclasses: os linfócitos T colaboradores ou **Th** (*T helper*), os linfócitos T citotóxicos ou **Tc** e os T supressores / reguladores ou **Ts**.

2.2.3 Principais Características das Respostas Imunológicas

As respostas imunológicas “*humoral*” e celular possuem várias propriedades fundamentais, como: caráter clonal, é específica para o antígeno, possui diversidade, desenvolve memória, possui autocontrole e distingue entre o próprio, uma célula ou molécula do próprio corpo, e do não próprio, antígenos estranhos e perigosos, como por exemplo: bactérias e vírus.

A especificidade, a diversidade, a clonalidade, a memória imunológica, o autocontrole e a discriminação são as características das respostas imunes. Alonso e Rubio (2005) apresentam as definições destas características, como a seguir.

- Especificidade: as respostas imunes são específicas para diferentes componentes estruturais, antígenos. Os linfócitos reconhecem individualmente porções de antígenos denominados determinantes ou epitopos.
- Diversidade: os linfócitos do antígeno se desenvolvem sem estimulação antigênica, de forma que em diferentes indivíduos imunizados existe clone de células com diferentes receptores para reconhecer e responder a exposição de antígenos estranhos. Este conceito é o princípio básico da seleção clonal.

- **Clonalidade:** quando um linfócito ou grupo de linfócitos é ativado, espalha e diferencia em muitas células derivadas de elos, que possuem receptores idênticos, capazes de reconhecer e unir-se ao mesmo antigênico, constituindo o que é denominado de clone de células.
- **Memória imunológica:** quando o sistema imune é exposto a um antígeno estranho, aumenta a sua capacidade de responder ao mesmo em encontros posteriores.
- **Autocontrole:** a resposta imunológica desenvolve diversos mecanismos internos de controle, como: o autocontrole, a intensidade e a duração da resposta aos antígenos.
- **Discriminação entre o próprio e o não próprio:** Uma das capacidades mais importantes do sistema imune é a capacidade de distinguir entre os antígenos próprios e os antígenos estranhos. A tolerância aos antígenos próprios é um fenômeno adquirido que deve ser aprendido pelos linfócitos de cada indivíduo.

A resposta imune também pode ser primária e secundária. A primária acontece quando um antígeno entra em contato com o corpo pela primeira vez. A secundária ocorre em encontros posteriores com este mesmo antígeno. As respostas primária e secundária são diferentes qualitativa e quantitativamente em três aspectos: 1 – os níveis de anticorpos específicos para o antígeno são alcançados após um longo período de latência da resposta primária. Já a resposta secundária é alcançada rapidamente; 2 – a resposta primária possui menor intensidade que a resposta secundária; e 3 – na resposta primária predomina a IgM (imunoglobina M), já a secundária predomina a imunoglobina G (IgG).

2.2.4 Funcionamento do Sistema Imune Biológico ou Natural

O sistema imune natural é um sistema distribuído com funcionalidades diferentes entre os componentes, que são posicionados em locais estratégicos e em todos os pontos, no nível mais baixo. Assim, a defesa é organizada em vários níveis ou camadas e usada contra os invasores não especificados (natural) ou específicos (contraídos). A organização da defesa, os níveis e as camadas de proteção são os mecanismos imunológicos do sistema.

O linfócito é o principal tipo de célula do sistema imune e a mais participativa. Possui os atributos de especificação, diversidade, memória e percepção. Existem duas subclasses de células linfocíticas – T e B, ou *B-Cells* e *T-Cells*. Cada uma possui uma função, primária e secundária, respectivamente. A primária é o órgão linfocítico, compostos pela medula óssea e pela glândula vascular, local onde um linfócito se desenvolve. Em particular as células – T

desenvolve-se na medula óssea e viaja para a glândula vascular para amadurecer. Já as células “B” são consideradas um linfócito desenvolvido e maduro residente dentro da medula óssea.

O órgão linfócito secundário, composto pela esplenalgia³ e vasos sanguíneos, funciona capturando o antígeno e fornecendo informações sobre os locais de interação do linfócito com o antígeno para simular a resposta para o sistema imunológico.

Uma resposta imune inclui uma série de reações em cadeias, similar ao “efeito dominó”, isto se dá devido à presença do antígeno em viroses, bactérias, extratos, agentes de infecções metabólicas ou vacinas, disparando um primeiro componente, que é o mensageiro molecular linfócito, efetuando uma alteração e ativando o próximo componente, até que a reação seja completa.

2.2.5 Reconhecimento de Padrões e Produção de Anticorpos

As células linfócitas B e T são responsáveis pelo reconhecimento de padrão do sistema imune (Dasgupta,1999b). Estas células carregam moléculas receptoras (BCR e TCR), capazes de reconhecer antígenos de características distintas. O receptor das células B interage com as moléculas antigênicas livres. O receptor das células T reconhece antígenos processados e ligados a uma molécula superficial chamada de complexo de histocompatibilidade maior ou *Major Histocompatibility Complex* – MHC. A função principal das células B é a produção e a liberação de anticorpos em resposta aos agentes patogênicos, enquanto que as células T são responsáveis pelo reconhecimento dos antígenos. (Barraza, 2004).

Cada célula produz um único tipo de anticorpo, propriedade conhecida como mono especificidade. Estes anticorpos são capazes de reconhecer e ligar-se a uma determinada proteína. A liberação e ligação dos anticorpos constituem um sinal para as outras células, fazendo com que estas interajam, processam e removam a substância ligada. A ilustração 2.2.5.2a, a seguir apresenta uma molécula de anticorpo na superfície de uma célula B. O reconhecimento imune ocorre em nível molecular e se baseia na complementaridade entre a região de ligação do receptor a uma porção do antígeno chamado epitopo. Os anticorpos representam somente um tipo de receptor, já os antígenos podem representar vários epitopos. Isto significa que distintos anticorpos podem reconhecer um mesmo antígeno. A figura 2.2.5.2b, ilustra esta representação.

³ Esplenalgia: Dor aguda na região do baço, provocada pelo processo de esplenocêntrização, que tem por objetivo repor os glóbulos vermelhos circulantes.

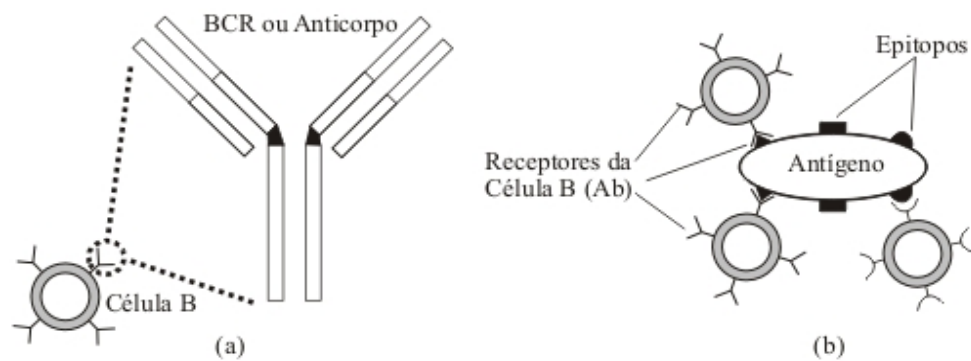


Figura 2.2.5.2: Célula B (De Castro e Timmis, 2002b). (a) A célula B mostra uma molécula de anticorpo em sua superfície. (b) A porção do antígeno que é reconhecida pelo anticorpo denomina-se epítopo. Os anticorpos são específicos, já os antígenos podem representar vários tipos de diferentes epítopos.

As células T podem ser divididas em dois subgrupos: 1 - Células Auxiliadoras (*Th* – *helper T-cells*); e 2 - Células T Assassinas ou Citotóxicas (*Tk* – *Killer T cells*) (Barraza, 2004).

Os antígenos das células receptoras T (TCR) são diferentes em sua estrutura comparando-se com os receptores B (BCR), como ilustrado a figura 2.2.5.2 (a e b), onde o receptor da célula T é ligado com o complexo MHC. As células **Tk** reconhecem os antígenos ligados pela molécula MHC-I, que são classes de moléculas encontradas em todas as células, estas permitem a detecção de células infectadas. Assim as células **Th** ajudam a manter uma revisão das células, de forma que toda vez que estas forem infectadas, o sistema induzirá uma resposta de eliminação do agente patogênico. Uma troca das células **Th** interage com os antígenos ligados a molécula MHC-II, que são moléculas representadas pelas células APC – *Antigen Presentation Cells*. As células APC's (Células Apresentadoras de Antígenos) capturam e processam a proteína antigênica. A figura 2.2.5.3a apresenta uma célula T mostrando ao TCR a superfície da célula. A figura 2.2.5.3b apresenta o processo de ligação de um captura um agente patogênico a uma molécula MHC-II própria.

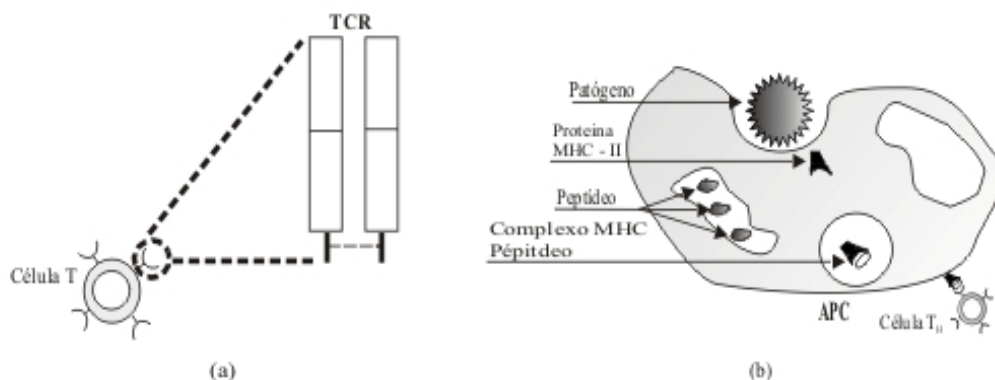


Figura 2.2.5.3: Célula T (De Castro e Timmis, 2002b) (a) Célula T mostrando ao TCR a superfície da célula. (b) O TCR ligando o patógeno ao complexo formando pela molécula MHC própria em um antígeno.

2.2.6 O Princípio da Seleção Clonal e as Redes Imunológicas

O princípio da seleção clonal é descrito por De Castro e Von Zuben (2000b) da seguinte maneira: cada linfócito que ainda não tenha sido envolvido em uma resposta imune, dito ingênuo ou sem memória, que entra na corrente sanguínea, leva receptores de antígeno de uma só especificidade. A especificidade dos receptores é determinada por um mecanismo de reordenamento genético que atua durante o desenvolvimento do linfócito.

A seleção clonal opera tanto sobre células B, quanto sobre células T. A principal diferença entre a expansão clonal de uma célula B e uma célula T, é que a célula B sofre mutações somáticas durante a reprodução, além destas células serem secretoras ativas de anticorpos. As células T por sua vez, não sofrem mutação durante a reprodução e são secretoras ativas principalmente de células Tk. A presença de elementos de mutação e de seleção no processo de expansão clonal das células B permite que os linfócitos incrementem sua diversidade de repositório e também melhore a sua capacidade de reconhecer os antígenos seletivos.

A teoria da seleção clonal apresenta várias propriedades. Em seu trabalho, De Castro (2001) destaca as principais:

- Seleção negativa: eliminação de linfócitos diferenciados recentemente, reativos a padrões de antígenos conduzidos por componentes próprios do organismo, chamados auto-antígenos.
- Expansão clonal: proliferação e diferenciação quando ocorre o contato entre linfócitos maduros e antígenos estranhos no organismo.
- Mono especificidade: restrição fenotípica de um padrão em uma célula diferenciada e retenção deste padrão por descendentes clonais.
- Hipermutação Somática: geração de novas trocas genética aleatórias, expressados por vários padrões diversos de anticorpos, ou uma forma de mutação somática acelerada.
- Auto imunidade: o conceito de um clone proibido resistente à eliminação por auto antígenos, esta é à base das enfermidades.

A descrição da rede imunológica é apresentada por De Castro e Von Zuben (2000a) da seguinte maneira. A porção de uma molécula de anticorpo responsável por reconhecer um epitopo denomina-se paratopo. Um idiotopo é definido como um conjunto de epitopos mostrado pelas regiões viáveis de um conjunto de moléculas de anticorpo, ou seja, um idiotopo é cada epitopo idiotípico. Os padrões determinados pelas mesmas regiões viáveis das

cadeias de polipeptídico do anticorpo, os quais também determinam os paratopos. Por conseguinte, os idiotopos localizam-se ao redor da célula, no lugar de ligação do antígeno, e cada “braço” de uma molécula de anticorpo possui um paratopo e um pequeno conjunto de idiotopo. A figura 2.2.6.4 (a e b) apresenta os conceitos básicos da rede imunológica, através da molécula de um anticorpo, paratopo e idiotopo, interagindo com um antígeno.

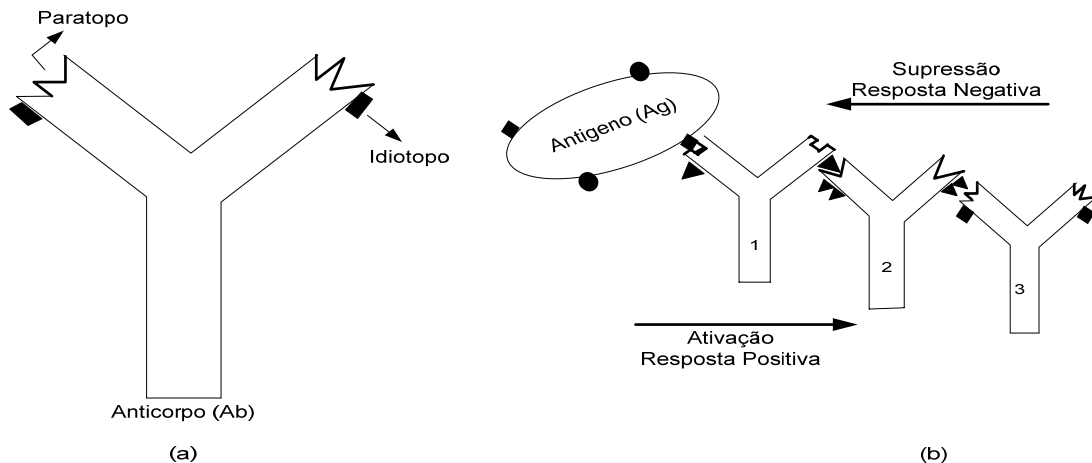


Figura 2.2.6.4: Conceitos Básicos da Rede Imunológica (De Castro, 2001). (a) Molécula de anticorpo, destacando seu paratopo e seu idiotopo. (b) Respostas positiva e negativa. Resultado da interação entre um paratopo e um idiotopo ou um epitopo.

2.3 Os Sistemas Imunes Artificiais

A computação imunológica é uma área emergente da pesquisa computacional, a qual iniciou-se há pouco mais de uma década. Os Sistemas Imunes Artificiais estão fortemente relacionados com os modelos biológicos empregados na biologia computacional ou biotecnologia.

De acordo com Vargas *et al.* (2002), os SIA podem produzir viabilidades e resultados para sistemas complexos, como: modelagem computacional e otimização combinatória. A eficácia dos SIA esta associada à utilização de algoritmos evolucionários, heurísticas e meta-heurísticas.

Um Sistema Imune Artificial (SIA) pode incorporar muitas propriedades dos sistemas imunes naturais, inclusive a diversidade da computação distribuída, da tolerância ao erro, da aprendizagem, da adaptação dinâmica e do autocontrole. Os SIA possuem a estrutura geral de um sistema distribuído, que, em princípio, pode ser aplicado a muitos domínios de problemas. Atualmente, os SIA tem sido amplamente estudados e pesquisados para a solução de problemas das mais variadas áreas de aplicação, que vai desde pátios industriais a

controles de tráfego aéreos e terrestres. Neste trabalho, a aplicação do SIA é voltada para problemas de escalonamentos.

2.3.1 Definindo um Sistema Imune Artificial

Há várias definições para os SIA na literatura, como as apresentadas na seção 1.1, contudo, para que um sistema seja caracterizado como um SIA, este deve, apresentar três requisitos básicos:

1. Representar um componente básico de sistema imune, por exemplo: uma célula, uma molécula ou um órgão;
2. Deve ter sido desenvolvido, incorporando idéias teóricas e/ou experimentais da imunologia; e,
3. Ter como objetivo principal a resolução de problemas.

É importante ressaltar que somente incluir termos imunológicos a um sistema não é suficiente para classificá-lo como um SIA. Além disso, a representação de um SIA requer um nível mínimo de conceitos imunológicos, tais como: um modelo de reconhecimento de padrões, o princípio imune, representados por algoritmos de seleção negativa, clonal e/ou de rede imunológica, técnicas inspiradas em processos biológicos, como as redes neurais e os algoritmos genéticos, entre outros.

Todas estas técnicas possuem os seguintes componentes em comum:

- Uma representação para os componentes do sistema;
- Um conjunto de mecanismos para avaliar a interação dos indivíduos, simulado por um conjunto de estímulos de entradas ou uma ou mais funções de aptidão;
- Procedimentos de adaptação, que inclui dinâmica do sistema e comportamento do sistema;

A partir destes componentes comuns, De Castro (2001) propôs um marco de desenvolvimento dos SIA com as seguintes equivalências:

- Uma representação para modelos abstratos dos órgãos, células e moléculas imunes;
- Um conjunto de funções, chamados de funções de afinidade, para quantificar as interações dos elementos dos SIA, dado pelos modelos dos órgãos, células e moléculas imunes;

- Um conjunto de algoritmos de propósito geral para avaliar e controlar a dinâmica dos SIA.

Este marco pode ser representado por diagrama de níveis, onde a base do sistema é o domínio da aplicação. Para este domínio, deve-se considerar a forma em que os componentes do sistema serão representados. Uma vez definido a representação do SIA, deve-se desenvolver as funções de afinidade, cujo objetivo é quantificar as interações dos elementos do sistemas. Um próximo nível inclui o uso de algoritmos ou processos para controlar e avaliar o comportamento do sistema. Um esquema deste diagrama de níveis pode ser representado pela figura 2.3.1.5.

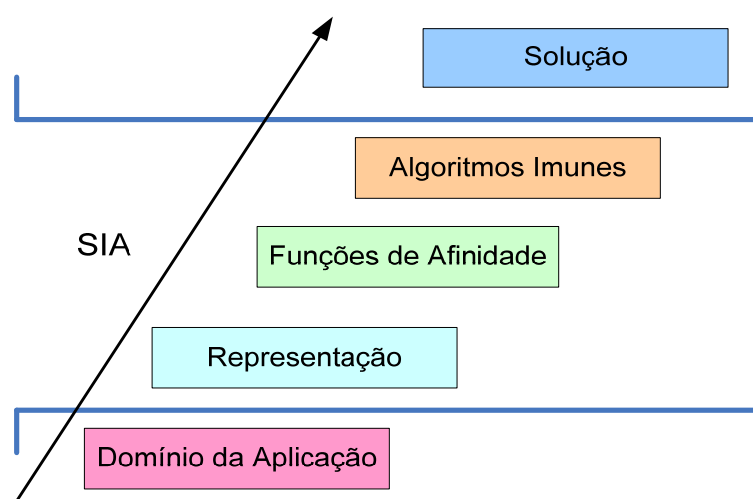


Figura 2.3.1.5: Diagrama de Níveis para Desenvolvimento dos SIA (Adaptado de De Castro e Timmis, 2002a). A figura apresenta as etapas para desenvolvimento dos SIA como um processo crescente, partindo do domínio da aplicação até a solução do problema. As etapas para desenvolvimento de um SIA são apresentadas em três módulos. No nível mais baixo ou a base é a representação do SIA. Um módulo intermediário mostra as funções de aptidão que devem ser desenvolvidas. A ultima etapa apresenta os módulos dos algoritmos imunes.

Os tópicos a seguir apresentam uma breve descrição da representatividade de cada nível, a fim de destacar a importância destes para o desenvolvimento de um SIA.

2.3.2 Representação dos Componentes Imunes

A representação dos componentes imunes obtém-se quando o sistema imune inato e o sistema imune adaptativo possuem moléculas receptoras capazes de reconhecer e identificar padrões de moléculas agressoras. Uma molécula receptora superficial de um leucócito, que reconhece um padrão molecular específico, denomina-se anticorpo, e o padrão molecular reconhecido por um anticorpo se denomina antígeno. Assim, nos linfócitos, um anticorpo corresponde à região viável de um TCR ou BCR e um antígeno é um epítopo ou idiótopo.

2.3.3 Funções de Afinidade

A função de afinidade pode ser representada da seguinte maneira: qualquer molécula m em um espaço de formas⁴ S , como uma cadeia de atributos ou conjunto de coordenadas possui um comprimento L . Assim, uma cadeia de atributos $m = m_1, m_2, m_3, \dots, m_L$, pode ser considerada como um ponto em um espaço de formas de L dimensões de S , ou seja, $m \in S^L$. Esta cadeia pode ser qualquer tipo de atributo, com valores reais, inteiros, cadeias de *bits* e símbolos. Estes atributos geralmente encontram-se determinados pelo domínio do problema do SIA.

A figura 2.3.3.6 apresenta um espaço de formas S e uma região V onde estão localizados porções de anticorpos e antígenos. Qualquer antígeno pode ser reconhecido por um anticorpo dentro da região, desde que seja encontrado o complemento em um determinado volume V_ϵ , ou a sua volta.

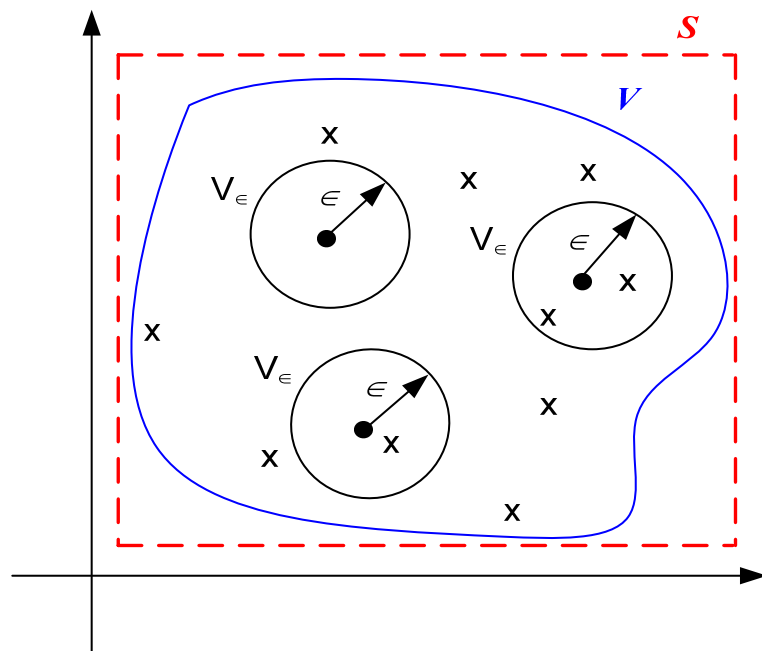


Figura 2.3.3.6: Espaço de Formas S em um Volume V (adaptado em De Castro, 2001). No espaço de formas S , existe um volume V , em que se localiza uma forma de um anticorpo (\bullet) com o complemento do antígeno (x). Um anticorpo pode reconhecer a qualquer antígeno em que o complemento se encontra dentro de um volume V_ϵ ou em volta dele.

O tipo de atributo define o tipo de espaço de formas a ser usado, da seguinte maneira:

- Espaço de formas real: as cadeias de atributos são vetores de reais;

⁴ Espaço de Formas: Local de ligação (*matching* ou reconhecimento) entre um receptor celular e seu ligante (molécula que se liga ao receptor celular). (De Castro, 2001).

- Espaço de formas de *Hamming*: cadeias de atributos construídas a partir de um alfabeto finito de comprimento k .
- Espaço de formas inteiros: As cadeias de atributos são compostas de valores inteiros;
- Espaço de formas simbólico: composto de diferentes tipos de cadeias de atributos, pelo menos uma delas é simbólica; ex. um nome, uma cor, etc.

Num caso geral, onde uma molécula de anticorpo pode ser representada por um conjunto de coordenadas, $Ab = Ab_1, Ab_2, Ab_3, \dots, Ab_L$, e um antígeno por $Ag = Ag_1, Ag_2, Ag_3, \dots, Ag_L$, **Ab** e **Ag** correspondem a uma cadeia sem perda de generalidade, ou seja, os antígenos e anticorpos são do mesmo tamanho.

A baixa perspectiva de reconhecimento de padrões, a interação entre os anticorpos ou entre um anticorpo e um antígeno, avalia por uma medida de distância, conhecida como medida de afinidade entre as cadeias. A medida de afinidade realiza um mapeamento da interação entre as cadeias a um número real não negativo que corresponde à afinidade ou grau de *match*, $S^L \times S^L \rightarrow R^+$. Por conseguinte, a afinidade (Ag, Ab) ou (Ab, Ag) é proporcional ao grau de complementaridade ou distância, entre as moléculas ou cadeias conforme apresenta a figura 2.3.3.7.

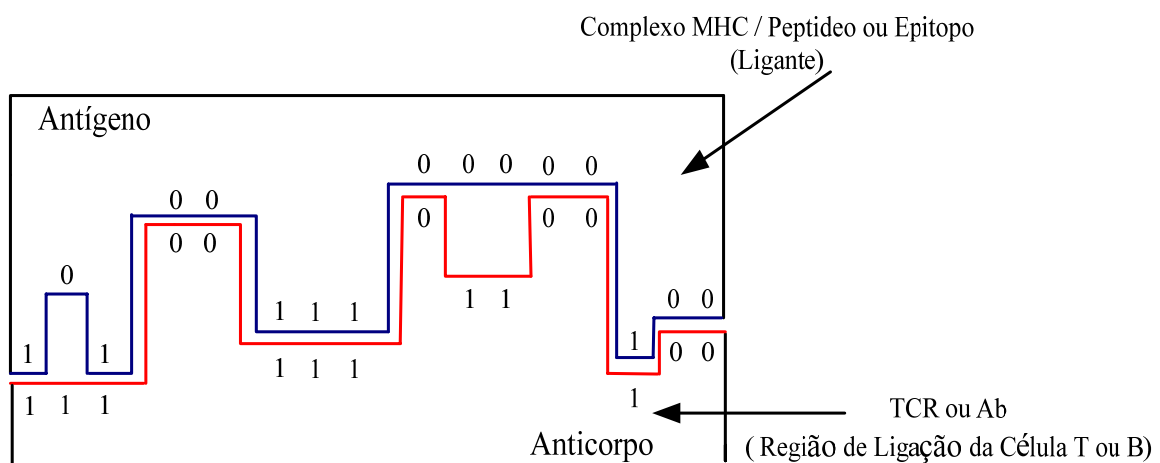


Figura 2.3.3.7: Regiões de Complementaridade (adaptado de De Castro, 2001). Reconhecimento do antígeno pelo anticorpo através de regiões de complementaridade. Representação das funções de afinidade.

A partir de uma representação de cadeias de atributos, na forma de um antígeno e de um conjunto de anticorpos, para cada cadeia que representa um anticorpo, se sua afinidade

pode ser associada a um antígeno, pode-se definir então a curva da afinidade sobre o espaço de formas, como ilustra a figura 2.3.3.8.

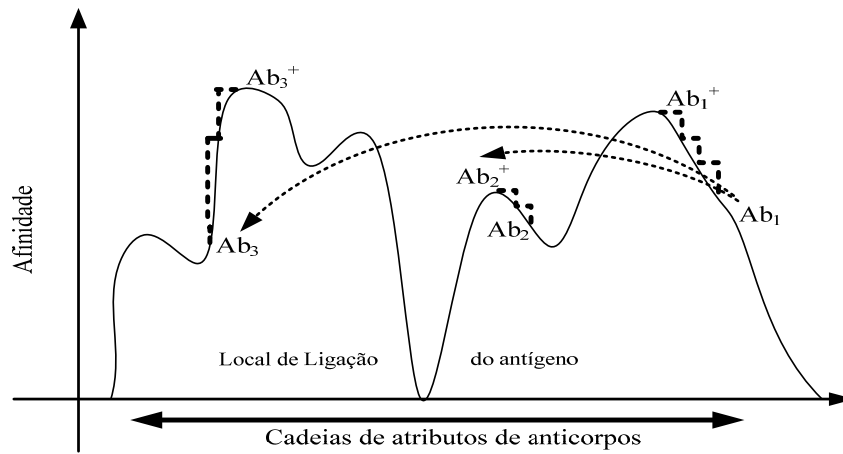


Figura 2.3.3.8: Gráfico de Afinidade Definido Sobre um Espaço de Formas (adaptado de De Castro 2001). A cada forma de anticorpo num espaço de formas é determinado um valor de afinidade relacionado a um antígeno. Formas diferentes produzem como resultados valores de afinidade diferente. Os anticorpos reconhecem os antígenos através dos pontos de ligação, possibilitando alcançar ótimos locais através da mutação somática e da seleção.

Nos espaços de formas reais, as medidas de afinidade podem ser de vários tipos, como por exemplo: a distância Euclidiana e a distância de *Manhattan*. No caso da distância Euclidiana, a afinidade D entre um antígeno (Ag) e um anticorpo (Ab) pode ser definido pela seguinte expressão:

$$D = \sqrt{\sum_{i=1}^L (Ab_i - Ag_i)^2}$$

Os espaços de formas que utilizam estas distâncias recebem o nome de espaços de formas Euclidianas.

A expressão a seguir mostra o caso da distância de *Manhattan* para a afinidade D entre um antígeno (Ag) e um anticorpo (Ab). Os espaços de formas reais que utilizam esta distância são chamados de espaços de formas de *Manhattan*.

$$D = \sum_{i=1}^L |Ab_i - Ag_i|$$

Os espaços de formas reais também podem ser representados por outros espaços de formas, como o espaço de formas de *Hamming*. Neste caso os antígenos (Ag) e os anticorpos (Ab) são representados como seqüências de símbolos sobre um alfabeto finito de tamanho k .

A expressão abaixo representa a medida de *Hamming*, que pode ser utilizada para avaliar a afinidade entre um antígeno (Ag) e um anticorpo (Ab) no espaço de forma *Hamming*.

$$D = \sum_{i=1}^L \delta_i, \quad \text{onde} \quad \delta_i = \begin{cases} 1, & \text{se } Ab_i \neq Ag_i \\ 0, & \text{caso contrário} \end{cases}$$

Os espaços de formas inteiros, onde as cadeias de atributos são compostos por valores inteiros, e os espaços de formas simbólicos, onde as cadeias de atributos incluem ao menos um atributo simbólico, podem ser adicionados aos espaços de formas reais e *Hamming* dependendo do problema, para avaliar o grau de interação dos componentes.

2.3.4 Algoritmos, Modelos e Processos

Após definir uma representação adequada para as células e moléculas imunes, e a forma de avaliar suas interações, é possível apresentar alguns algoritmos de propósito geral que modelam aspectos específicos do sistema imune.

Existem alguns algoritmos e procedimentos capazes de tratar diversos mecanismos e teorias imunes, que podem ser utilizados como unidades básicas para desenvolver um SIA. Estes algoritmos são classificados como genéricos e podem ser utilizados em diversas aplicações. De Castro (2001) destaca estes algoritmos como dois modelos que representam os componentes importantes do sistema imunológico, compostos pelo timo⁵ e pela “medula óssea”, e duas teorias distintas, composta pela “seleção clonal” e as “redes imunes”:

- Modelos da Medula Óssea: utilizados para gerar repertórios de células e moléculas;
- Modelos do Timo: utilizados para gerar repertórios de células e moléculas capazes de realizar a discriminação entre o próprio e o não próprio;
- Algoritmos de Seleção Clonal: utilizados para controlar a interação dos componentes do sistema imune e o meio externo ou antígenos;
- Modelos de Rede Imune: utilizados para simular redes imunes, a partir de sua estrutura dinâmica e meta dinâmica.

A seção a seguir apresenta os algoritmos. Os modelos biológicos, descritos acima, são necessários e importantes para o mapeamento do processo imune biológico e conseqüentemente para os algoritmos imunes computacional, porem estes não serão tratados neste trabalho com o intuito de evitar uma abordagem mais apurada de conceitos biológicos.

⁵ Timo: Órgão linfático desenvolvido no período pré-natal. Involui do nascimento até a puberdade.

2.3.5 Algoritmo de Seleção Positiva, Negativa e Clonal

A seleção positiva de células T estimula e permite o amadurecimento somente das células T capazes de reconhecer moléculas MHC próprias. Este processo gera a restrição MHC, que assegura a propriedade na qual as células T reconhecem antígenos somente se eles se apresentam no contexto de células MHC próprias.

Uma proposta de um modelo celular autômato foi apresentado por Celada e Seiden (1994) para realizar simulações computacionais do sistema imune, baseados no mecanismo de resposta imune. O modelo proposto, não caracteriza um SIA, pois não apresenta as características apontadas na seção 2.3.1.

O algoritmo de seleção positiva possui propriedades interessantes a partir de uma perspectiva computacional. O algoritmo de seleção positiva pode ser ilustrado através da figura 2.3.5.9 e descrito pelos seguintes passos.

1. Início: Gerar o repertório potencial **P** de células T imaturas. Suponha que todas as moléculas receptoras e MHC se apresentam como cadeias binárias de um mesmo tamanho L . Logo é gerado 2^L células distintas.
2. Função de Avaliação da Afinidade: determinar a afinidade de todos os elementos em **P** com todos os elementos do conjunto próprio **S**.
3. Geração do Repertório Disponível: Se a afinidade de um elemento de **P** com pelo menos uma molécula MHC é maior ou igual que uma entrada, então a célula T reconhece esta molécula MHC e seleciona positivamente e introduz ao sistema. Senão a célula T é eliminada.

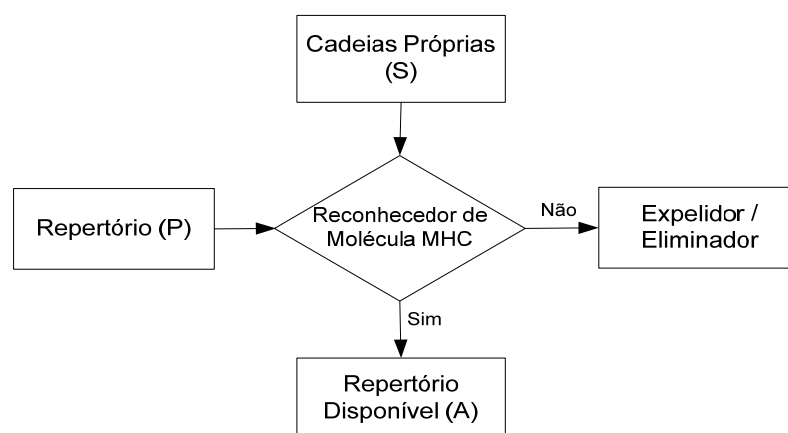


Figura 2.3.5.9: Representação do Algoritmo de Seleção Positiva (adaptado de Forrest *et al.* 1994). O esquema do algoritmo de seleção positiva apresenta um conjunto **S** de cadeias próprias, um repertório potencial **P** e um reconhecedor. Se a molécula for reconhecida, esta vai para o repertório disponível **A**, senão esta será repelida.

A Seleção negativa das células T é responsável pela eliminação de células T, cujos receptores são capazes de ligar-se a “peptídeos” próprios apresentados pela molécula MHC. Este processo garante que as células T que abandonam o “timo” não sejam reconhecidas por nenhuma molécula ou célula própria.

Um algoritmo de detecção de trocas inspirado na seleção negativa de células T no “timo” foi proposto por Forrest *et al.* (1994). O algoritmo é conhecido como algoritmo de seleção negativa, sendo aplicado originalmente em segurança computacional. O algoritmo modela somente um tipo de célula imune: células T, representadas como cadeias de *bits* de tamanho L . Dado um conjunto próprio S , os receptores das células T têm que ser testados para avaliar sua capacidade de reconhecimento e coincidência com os peptídeos próprios. Se uma célula T reconhece um peptídeo próprio, esta é descartada. Caso contrário, a célula T é selecionada como uma célula “imunocompetente” e é incluída no repertório disponível A . O algoritmo de seleção negativa pode ser apresentado através da figura 2.3.5.10 e descrito pelos seguintes passos:

1. **Início:** Gerar aleatoriamente cadeias e associa-las a um conjunto P de células T imaturas. Assuma que todas as moléculas receptoras e MHC apresentam-se como cadeias binárias de um mesmo tamanho L .
2. **Função de Avaliação da Afinidade:** determinar a afinidade de todas as células T em P com todos os elementos do conjunto próprio S .
3. **Geração do Repertório Disponível:** Se a afinidade de uma célula T imatura, dado pelo elemento de P , com pelo menos um peptídeo próprio for maior ou igual que uma entrada, então a célula T reconhece o peptídeo próprio, e este é eliminado. Senão a célula T é incorporada ao repertório disponível A .

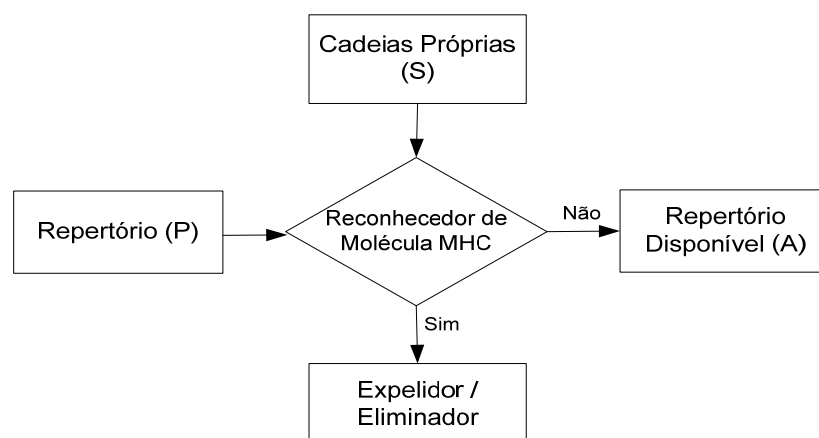


Figura 2.3.5.10: Algoritmo de Seleção Negativa (adaptado de Forrest *et.al* 1994). O diagrama representa o algoritmo de seleção negativa, onde é apresentada um conjunto S de cadeias próprias, um repertório potencial P e um reconhecedor de molécula MHC. Se a célula T reconhece o “peptíco” próprio, este deve ser guiado a eliminação. Senão, a célula T é incorporada ao repertório disponível A .

A seleção clonal é o meio pelo qual o sistema imune adaptativo enfrenta os microorganismos patogênicos. Este conceito é válido tanto para as células B, quanto para as células T, com a diferença que somente as células B sofrem hipermutação somática durante sua proliferação. A capacidade de adaptabilidade das células B possui grande relevância para o desenvolvimento dos SIA. Partindo desse princípio, muitos algoritmos foram propostos na comunidade SIA.

Um algoritmo de seleção clonal chamado CLONALG foi proposto por De Castro e Von Zuben (2000b). Este algoritmo compõe-se basicamente de dois repertórios de cadeias: um conjunto de antígenos Ag e um conjunto de anticorpos Ab . O conjunto Ab pode se decompor em vários subconjuntos de acordo com a aplicação, que pode ser: reconhecimento de padrões ou otimização. Os passos básicos do CLONALG, a partir de um conjunto de padrões a ser reconhecidos, são:

1. **Início:** Criar uma população inicial de indivíduos (Ab) aleatoriamente.
2. **Representação Antigênica:** para cada padrão antigênico, fazer:
 - a. **Função de Avaliação de Afinidade:** apresentar o padrão antigênico a população Ab e determinar sua afinidade com cada elemento de Ab .
 - b. **Seleção e Expansão Clonal:** Selecionar n elementos com a afinidade mais alta e gerar clones destes indivíduos de forma proporcional a sua afinidade, de acordo com o antígeno. Maior afinidade produz o maior número de cópias, e vice e versa.
 - c. **Amadurecimento da Afinidade:** “Mutar” todas estas cópias com uma taxa inversamente proporcional a sua afinidade com o padrão de entrada. Maior afinidade, menor taxa de mutação, e vice versa. Incorporar estes indivíduos “mutados” na população Ab e selecionar novamente o melhor indivíduo a ser guardado como a memória m do antígeno apresentado.
 - d. **Meta Dinâmica:** Substituir um número d de indivíduos com baixa afinidade por indivíduos novos, gerados aleatoriamente.
3. **Ciclo:** Repetir o passo 2 até que um critério de parada seja satisfeito.

O CLONALG usa uma matriz de elementos Ab_m , composta de todos os m , que são parte do conjunto Ab ($Ab_m \subseteq Ab$), com a diferença que eles são substituídos somente por

elementos de maior afinidade. Este processo, junto com a mutação proporcional da afinidade, realiza uma busca “*greedy*” na curva de afinidade.

O CLONALG pode ser aplicado em reconhecimento de padrões e problemas otimização. A literatura dispõe de outros algoritmos baseado na teoria da seleção clonal, que podem ser aplicados em diversos seguimentos computacionais, além de determinar um caminho para novas propostas e estudos na área dos SIA.

3 O PROBLEMA DE ESCALONAMENTO *JOB SHOP*

Neste Capítulo, será introduzido de forma breve o clássico problema JSS. Dentre as várias abordagens existentes na literatura para o tratamento do problema, será destacado a aplicabilidade dos algoritmos genéticos e apresentado um AG específico, a ser utilizado pelo SIA desenvolvido.

3.1 Apresentação do Problema *Job Shop Scheduling*

Os problemas de escalonamento (*scheduling*) são aqueles que envolvem alocação de recursos no tempo com a finalidade de executar uma série de tarefas. Esses problemas exigem um grande esforço computacional para produzir uma solução ótima com um custo/tempo computacional razoável. O problema de *Job-Shop*, é um problema da classe *NP - Hard*. (Blazewicz *et al.* 1996).

Um problema *Job-Shop* é normalmente referenciado como um *n/m JSP*, onde *n* é o número de *jobs* e *m* é o número de máquinas. Um conjunto de *n jobs* $J = \{J_1, J_2, \dots, J_n\}$ deve ser processado em um conjunto de *m* máquinas $M = \{M_1, M_2, \dots, M_m\}$ disponíveis. Cada *job* possui uma ordem de execução específica entre as máquinas, ou seja, um *job* é composto de uma lista ordenada de operações, cada uma das quais definida pela máquina requerida e pelo tempo de processamento na mesma (Klein, 2000).

No modelo clássico de *job shop* tem-se um conjunto *J* de *jobs* e um conjunto *M* de máquinas. Cada *job* $j \in J$ consiste em um conjunto de operações $O_{j\mu_{jk}}$ em que $j \in J, k \in K_{n_j} := \{1, 2, \dots, n_j\}$ e n_j é o número total de operações para cada *job* *j*, e μ_{jk} é o índice da máquina para processar a “*k*-ésima” operação do *job* *j*, ou seja, $\mu_{jk} \in \{1, 2, \dots, m_M\}$.

De acordo com Pinedo (1995), as seguintes condições devem ser consideradas para o problema.

- Todas as máquinas são diferentes e suas velocidades de processamento são constantes.

- Operações não podem ser interrompidas;
- Cada *job* somente é escalonado uma única vez;
- Cada máquina pode processar apenas uma operação em cada instante de tempo;
- Cada *job* só pode estar sendo processado em uma única máquina;
- Um *job* pode ser escalonado em qualquer máquina;
- Cada *job* é produzido por uma seqüência conhecida de operações;
- Não existe restrição de precedência entre operações de diferentes *jobs*.

O objetivo mais comum empregado em um problema *job shop scheduling* é encontrar uma seqüência de *jobs* válida, ou seja, que não viole nenhuma das restrições acima, e que complete todos os *jobs* com um menor tempo possível. Este objetivo é conhecido como *makespan*, e tem por finalidade minimizar o tempo de finalização das operações.

Um exemplo de um *job shop* clássico pode ser apresentado através da tabela 3.1.0.1.

Tabela 3.1.0.1: Um *Job Shop* Clássico 3X3.

REPRESENTAÇÃO DE UM <i>JOB SHOP SCHEDULING</i> 3 X 3			
<i>Job</i>	<i>Máquina Utilizada / Tempo</i>		
1	1(3)	2(3)	3(3)
2	1(2)	3(3)	2(4)
3	2(3)	1(2)	3(1)

A solução pode ser representada por um diagrama de Gantt, como apresentado na figura 3.1.0.11.

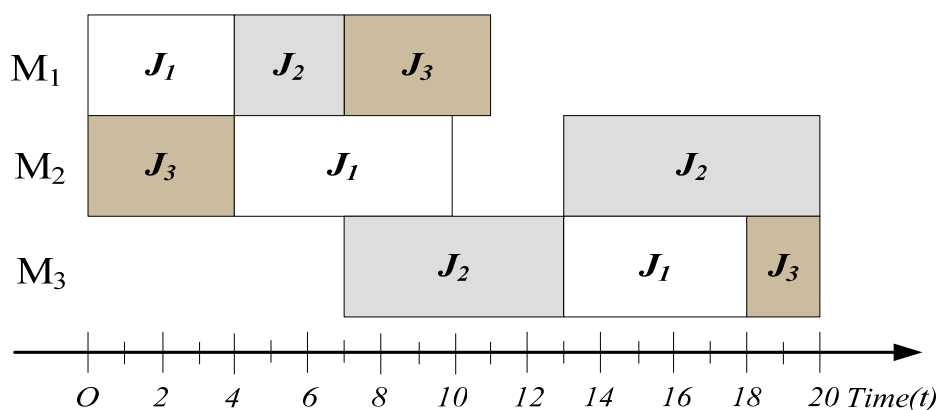


Figura 3.1.0.11: Representação de Job Shop Scheduling. A figura apresenta um *Job shop Scheduling* por meio de um diagrama de *Gantt*, onde M são as máquinas e J são os *job's* que serão processados em um dado tempo (*time*), obedecendo criteriosamente a ordem de processamento das operações e as restrições do problema JSS.

Uma vez que as seqüências de máquinas de cada *job* são fixadas, o problema a ser resolvido consiste em determinar as seqüências dos *job*'s em cada máquina, de forma que o tempo de execução transcorrido, desde o início do primeiro *job* até o término do último seja minimizado. O objetivo é encontrar a solução com menor *makespan*.

3.2 Representação do Problema *Job Shop Scheduling*

O uso de algoritmos genéticos é bastante comum em problemas de otimização, especificamente em problemas de escalonamentos de tarefas. Este trabalho propõe resolver o problema de escalonamento de tarefas através de um SIA.

Será usado como base principal para a realização deste projeto, os trabalhos de Fang (1994), Fang *et al.* (1993), Bean (1994) e Jain e Meeran (1998), que utiliza um algoritmo genético para gerar os escalonamentos procurados.

Para representar o problema de escalonamento de tarefas, faz-se necessário definir uma solução candidata dada por um cromossomo e separá-la entre genótipo – constituição genética de um indivíduo que pode ser representado por um cromossomo – e fenótipo – características de cada indivíduo constituídas a partir do genótipo. Assim, obtêm-se o escalonamento correspondente de cada indivíduo através de um algoritmo de escalonamento aplicado ao genótipo.

O genótipo é uma cadeia de $j \times m$ números inteiros, onde cada gene assume valores entre 0 a $j - 1$, j é o número de *jobs* e m é o número de máquinas. O algoritmo de *schedule* lê seqüencialmente os genes do cromossomo e para cada gene i , insere a próxima tarefa do “ i -ésimo” *job* incompleto, não completada no fenótipo. A lista de *jobs* incompletos mantida pelo algoritmo é circular, ou seja, se não houver “ i -ésimo” *job* incompleto, é escolhido o *job* da lista circular na posição $i \bmod n$ (i modulo n), onde n é o número de *jobs* incompletos. Este procedimento pode ser apresentado a partir da tabela 3.2.0.2.

Tabela 3.2.0.2: Representação de *Job*'s Escalonados

REPRESENTAÇÃO DE ESCALONAMENTOS			
$j \times m$ (3X3)	(m, t)	(m, t)	(m, t)
Job 0	2, 1	0, 3	1, 2
Job 1	1, 2	2, 1	0, 2
Job 2	0, 2	1, 2	2, 2

Uma outra tabela contendo os genótipos, que são as cadeias de inteiros, pode ser associada para representar o processo e exemplificar a criação dos genótipos. Para efeito de

ilustração do exemplo, utiliza-se o seguinte genótipo: 021112210. A tabela 3.2.0.3 apresenta a criação dos genótipos para resolução de um problema de escalonamento.

Tabela 3.2.0.3: Exemplo de Criação de Fenótipos

EXEMPLO DE GERAÇÃO DE FENÓTIPOS	
0 → primeiro <i>job</i> incompleto: <i>job</i> 0	máquina 2
2 → terceiro <i>job</i> incompleto: <i>job</i> 2	máquina 0
1 → segundo <i>job</i> incompleto: <i>job</i> 1	máquina 1
1 → segundo <i>job</i> incompleto: <i>job</i> 1	máquina 2
1 → segundo <i>job</i> incompleto: <i>job</i> 1	máquina 0 (fim do <i>job</i> 1)
2 → segundo <i>job</i> incompleto: <i>job</i> 2	máquina 1 (2° <i>job</i> incompleto)
2 → segundo <i>job</i> incompleto: <i>job</i> 2	máquina 2 (fim do <i>job</i> 2)
1 → terceiro <i>job</i> incompleto: <i>job</i> 1	máquina 0 (único <i>job</i> incompleto)
0 → terceiro <i>job</i> incompleto: <i>job</i> 0	máquina 1 (fim do <i>job</i> 0)

Não existe uma relação direta entre o valor de um gene no cromossomo e um *job* no escalonamento, isto pode implicar em grau elevado de redundâncias, já que vários genótipos podem gerar um mesmo fenótipo.

3.3 Aplicando Algoritmos Genéticos

Os Algoritmos Genéticos (AG), propostos inicialmente por Holland (1975), são muito eficientes para busca de soluções em uma grande variedade de problemas, pois não impõem muitas das limitações encontradas nos métodos de busca tradicionais. Os métodos tradicionais, geralmente iniciam uma busca com um único candidato, que iterativamente, é manipulado utilizando algumas heurísticas (estáticas) diretamente associadas ao problema a ser solucionado. Os Algoritmos Genéticos operam sobre uma população de candidatos, desta forma podem fazer buscas em diferentes áreas do espaço de soluções.

Quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será sua chance de sobreviver e gerar descendentes: este é o conceito básico da evolução genética biológica.

Conforme Gonçalves *et al.* (2002) os AG's são algoritmos de otimização, baseados nos mecanismos de seleção natural. Eles empregam uma estratégia de busca paralela e estruturada, mas aleatória, que é voltada para a busca de pontos de “alta aptidão”, ou seja, pontos em que a função a ser minimizada ou maximizada tem valores relativamente baixos ou altos respectivamente.

Durante cada iteração, os princípios de seleção e reprodução são aplicados a uma população de candidatos que pode variar, dependendo da complexidade do problema e dos recursos computacionais disponíveis. Através da seleção, se determina quais indivíduos

conseguirão reproduzir, gerando um número determinado de descendentes para a próxima geração, com uma probabilidade determinada pelo seu índice de aptidão. Em outras palavras, os indivíduos com maior adaptação relativa têm maiores chances de se reproduzir. .

3.3.1 Representação do Cromossomo

A representação do indivíduo, na forma de cromossomo, é feita normalmente através de um vetor, onde cada posição do vetor representa um gene da cadeia, gerado aleatoriamente, o qual possui um elemento que denota uma característica do indivíduo. Deste modo, um cromossomo pode ser representado por cadeias (*string*) binárias, literal ou numérica, correspondendo a uma lista ordenada.

3.3.2 Processo de Geração de uma População

A geração de uma população inicial na maioria dos casos é feita aleatoriamente. Em seguida esta população é avaliada por uma função de aptidão que elege os melhores indivíduos. Os indivíduos eleitos são selecionados para cruzamento, criando assim uma nova população.

O processo de geração de uma população pode ser dividido em três partes distintas: a geração inicial, a geração intermediária e a próxima geração.

Na primeira parte são selecionados os dados que serão duplicados, criando assim uma geração intermediária. Após esta duplicação, executa-se a recombinação desses dados para a geração da nova população. Este processo é ilustrado na figura 3.3.2.12.

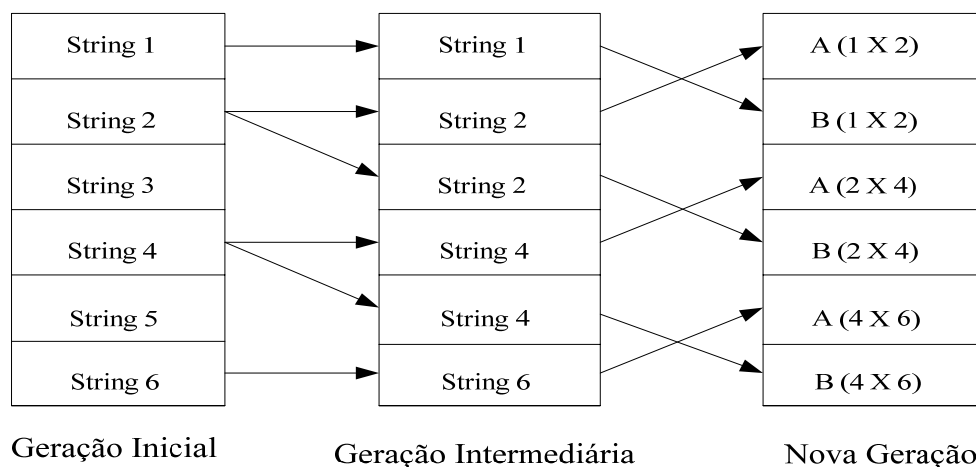


Figura 3.3.2.12: Geração de uma Nova População. O processo de geração de uma nova população inicia-se a partir de uma população inicial. Usando-se operadores de seleção é gerada uma população intermediária, que é recombina para gerar a nova população.

3.4 Operadores Genéticos

Os operadores genéticos transformam a população através de sucessivas gerações, estendendo a busca até chegar a um resultado satisfatório. Um algoritmo genético básico evolui através de gerações consecutivas a partir de três operadores básicos, que são: os operadores de seleção, cruzamento ou *crossover* e mutação, representados pela figura 3.4.0.13.

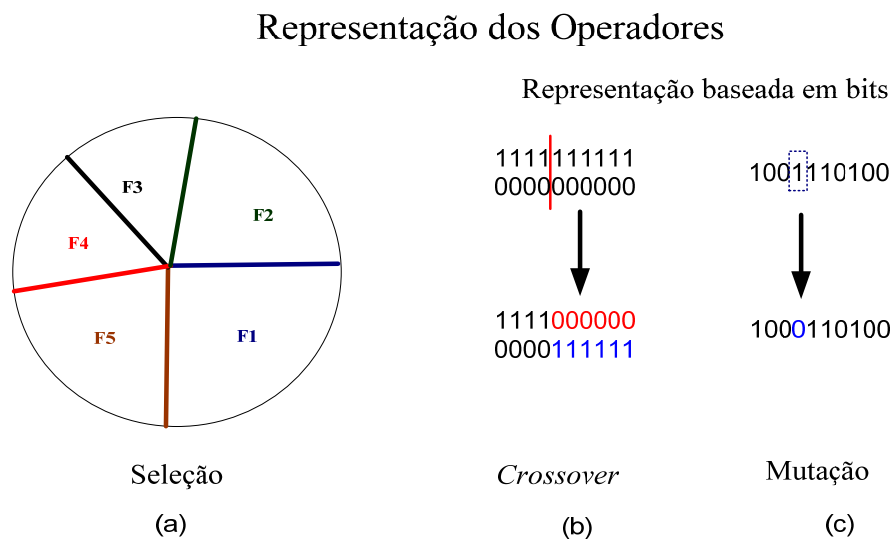


Figura 3.4.0.13: Operadores Genéticos. A figura apresenta os operadores genéticos de seleção (a), representado uma roleta com 5 indivíduos na população. Cada fatia representa a aptidão do indivíduo. O operador de *crossover* (b) é representado por um par de cadeias binárias e o operador de mutação (c), também aparece com uma representação baseada em bits.

3.4.1 Operadores de Seleção

O operador de seleção é o responsável pela oferta dos melhores indivíduos da população corrente, priorizando o processo de reprodução, permitindo assim que os indivíduos forneçam suas características para as gerações futuras. Os operadores de seleção mais comuns são: seleção por roleta, seleção por torneio e seleção por sorteio.

- *Seleção por Método da Roleta (Roulette Well):* os indivíduos são selecionados através de um vetor construído a partir do *fitness* de cada indivíduo. A primeira posição tem o valor do *fitness* do primeiro indivíduo. As posições seguintes têm o valor da posição anterior mais o *fitness* do indivíduo correspondente. O intervalo do “i-ésimo” indivíduo é o seu valor correspondente no vetor menos o anterior. Desta forma, indivíduos com *fitness* maiores, terão um intervalo maior, aumentando assim a probabilidade de escolha dos melhores indivíduos. O

algoritmo genético desenvolvido para o SIA, proposta deste trabalho, utiliza o método da roleta.

- *Seleção por Sorteio (Cara ou Coroa ou Coin Toss)*: este método, utilizado no trabalho de Gonçalves *et al.* (2002), anteriormente proposto por Bean (1994). Assim como o método da roleta, utiliza um vetor construído a partir do *fitness*. A diferença é que neste método um número é gerado a partir de um sorteio entre dois números ou duas posições no vetor, ou entre duas posições de dois vetores distintos, como utilizado por Gonçalves *et al.* (2002). O método consiste em um sorteio baseado em cara ou coroa (*coin toss*) sobre dois elementos dos vetores *a* e *b*, respectivamente. O elemento escolhido é armazenado em um terceiro vetor *c*. O processo continua até que todas as posições do vetor *c* tenham sido preenchidas com os resultados dos sorteios.

- *Seleção por Torneio (Tournament)*: a seleção por torneio consiste em confrontar os indivíduos, neste caso, os indivíduos menos aptos, vão sendo eliminados, e os indivíduos com maior aptidão vão sendo selecionados a cada rodada ou iteração. Cada elemento da amostra é selecionado e comparado, elegendo o melhor indivíduo de um conjunto *S* de elementos obtidos aleatoriamente na população.

3.4.2 Operadores de Cruzamento ou *Crossover*

Pode-se dizer que a principal característica dos AG's em relação a outras técnicas é o cruzamento. O operador de cruzamento é utilizado após a seleção dos indivíduos em uma população corrente. Esta fase é marcada pela troca de características entre os pais de cromossomos selecionados para que novos indivíduos sejam gerados, formando assim a população da próxima geração.

A idéia central do cruzamento é propagar as características dos indivíduos mais aptos da população, mesmo que estes não produzam indivíduos melhores ou mais aptos.

A literatura disponibiliza uma série de cruzamentos aplicados aos algoritmos genéticos para atender uma classe de problemas ou para resolver um problema específico. Além disso, pode-se criar um cruzamento específico para um determinado problema, desde que este obedeça a forma de reprodução “sexuada” dos AG's.

Os cruzamentos mais comuns são: o cruzamento simples ou de um ponto, o cruzamento de dois pontos e o cruzamento uniforme. Outros operadores de cruzamento, como: *Partially Matched Crossover* (PMX), *Order Based Crossover* (OX), *Cycle Crossover*

(CX) e *Overlap Crossover*, apresentados por Goldberg (1989), podem ser usados, principalmente para tratar problemas de escalonamentos.

Para objetivar e dar maior dinamismo ao texto, será apresentado a seguir somente os operadores utilizados na implementação do SIA.

- *Crossover Simples ou de Um ponto*: o *crossover* de um ponto escolhe aleatoriamente uma posição entre 0 e n do indivíduo. O tamanho do indivíduo será o produto entre número de *jobs* e número de máquinas. Este critério foi adotado devido à existência de ambigüidades na representação utilizada: a partir de uma dada posição, os valores dos genes passam a depender da configuração dos genes anteriores. Desta forma, não haveria modificação significativa se apenas a parte final do cromossomo fosse modificada. A figura 3.4.2.14a apresenta um operador de *crossover* de único ponto na sua forma geral, e a figura 3.4.2.14b, um operador de *crossover* simples ou de um ponto para um problema de 3 máquinas e 3 *jobs*:

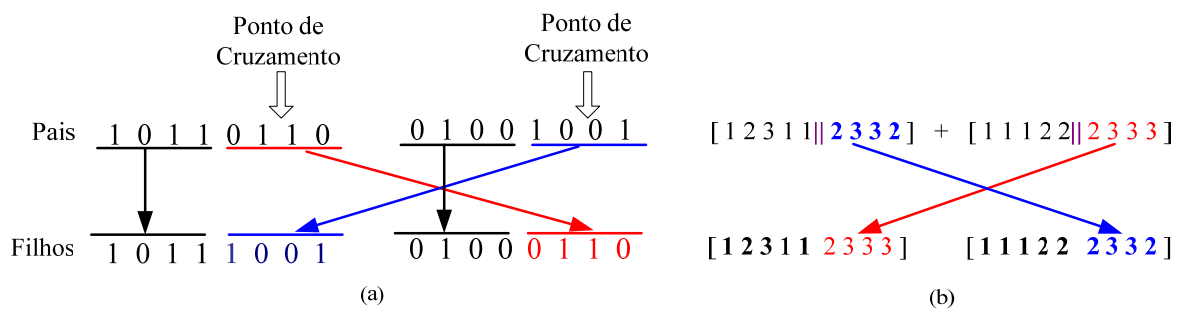


Figura 3.4.2.14: Operador de Crossover de um Ponto. (a) Exemplo genérico de um operador de cruzamento de um ponto. (b) Um operador de *crossover* de um ponto para uma operação de 3 *job's* em 3 maquinas, representando um ambiente do tipo JSS.

- *Crossover Uniforme*: no *crossover* uniforme, cada gene é criado copiando o gene correspondente de um dos pais, escolhidos de acordo com uma máscara de cruzamento gerada aleatoriamente. Onde houver 1 na máscara de cruzamento o gene correspondente será copiado do primeiro pai, caso contrário (0), será copiado do segundo pai. O processo é repetido com os pais trocados para produzir um segundo descendente. Uma nova máscara de cruzamento é criada para cada par de pais. O número de pontos de trocas não é fixo, mas em geral é usado $\ell/2$ (onde ℓ é o tamanho do cromossomo), representando 50% de cada pai. O *crossover* uniforme pode se tornar ineficiente para trocas com taxas superiores a 70%, devido a geração de redundâncias. A figura 3.4.2.15 mostra o processo de cruzamento uniforme através de um exemplo geral e para um problema de 3 máquinas e 3 *jobs*.

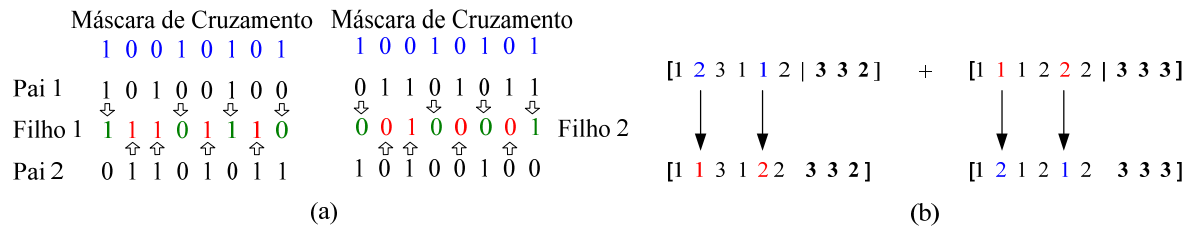


Figura 3.4.2.15: Crossover Uniforme. (a) Exemplo genérico de um operador de *crossover uniforme*. (b) Um operador de *crossover uniforme* para uma operação de 3 máquinas e 3 *job's*, ou seja, problema JSS 3x3 clássico.

Além dos operadores básicos apresentados anteriormente, existem uma segunda classe de operadores, utilizados por Hart e Ross (1999a, 1999b), para produzir uma população de anticorpos. Essa classe é composta de três operadores: 1 – *2Pt Crossover*, usado quando dois indivíduos não possuem genes em comum, 2 – *Order-Based Crossover (OX)*, usado para a permutação de um operador, e 3 – *Overlap Crossover*, usado para duplicar genes já escolhidos de outro indivíduo. Os pontos a seguir apresentam a definição e o exemplo destes operadores.

- *Crossover de Dois Pontos (2 Pt X)*: o *crossover* de dois pontos é similar ao *crossover* de um ponto, diferenciando-se apenas na troca de segmentos, onde é escolhido dois termos de representação ou dois pontos de cruzamento para a troca dos genes. O esquema do *crossover* de dois pontos é ilustrado na figura 3.4.2.16, apresentada abaixo.

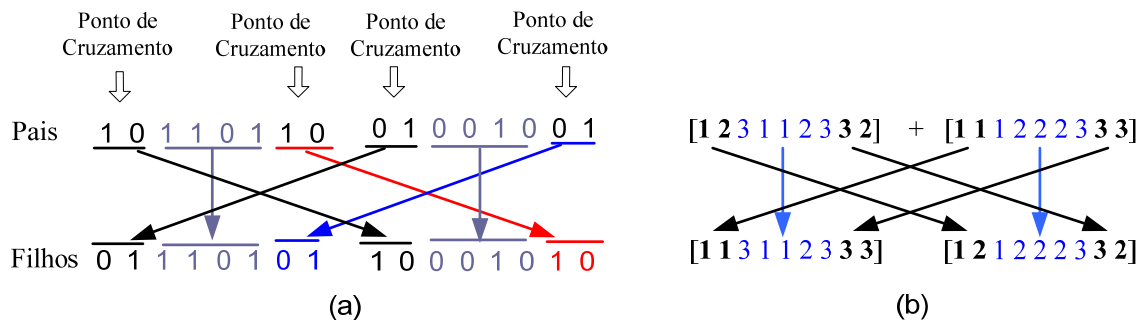


Figura 3.4.2.16: Crossover de Dois Pontos. São escolhidos dois pontos para troca de segmentos dos genes e gerar o descendente. (a) Apresenta um operador de cruzamento de dois pontos genérico e (b) um operador de dois pontos para um *Job Shop Problem 3X3*.

- *Order-Based Crossover (OX)*: o *OX* é usado para permutar operadores através de deslizamentos (*sliding motion*) para preencher as falhas de acordo com as posições marcadas por uma sessão de *matching*. Isto é feito a partir da esquerda, transferindo e mapeando posições. Para ilustrar o funcionamento do *order crossover*, será usado o exemplo apresentado por Goldberg (1989). Usando duas cadeias *A* e *B*, seleciona-se uma sessão de permutação. Quando a cadeia *B* é cruzada com a cadeia *A*, as posições 5, 6, 7 deixarão falhas (*hole*), marcadas com *H* na cadeia. Estas falhas são preenchidas com deslizamento que inicia-

se a partir do segundo cruzamento. As falhas são então preenchidas com os elementos da sessão correspondente (*matching*). Após completar a operação de preenchimento das falhas pela operação de cruzamento, obtêm-se os filhos de A e B , representados por A' e B' . A figura 3.4.2.17 ilustra este procedimento.

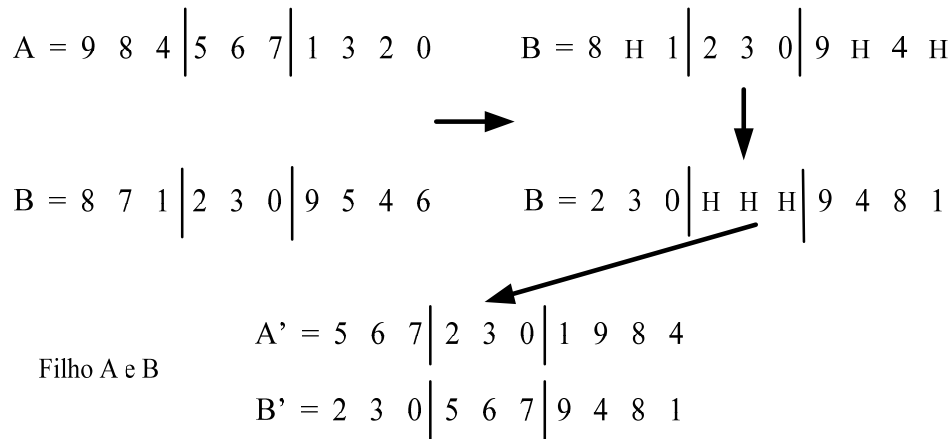


Figura 3.4.2.17: Order Based Crossover. Após selecionar uma região de permutação nas cadeias A e B , e feito um cruzamento deixando a cadeia B com falhas nas posições 5, 6 e 7, que são preenchidas pelo H . O processo de deslizamento é realizado e as falhas ' H ' passam para região de permutação (*matching*). O cruzamento é realizado gerando os filhos A' e B' .

- *Overlap Crossover*: O *overlap crossover* foi proposto no algoritmo genético de Grefenstette (1984) para desenvolver uma população de anticorpos e aplicado no PRAIS (*Pattern Recognizing Artificial Immune System*) de Hart e Ross (1999a) para o mesmo fim. O *overlap crossover* é usado se um pai possui pontos em comum com outro. Neste caso os pais são alinhados para que as linhas das regiões aumentem o grau de comparação. A posição mais a esquerda é lida. Se ambos parentes possuem um gene comum numa posição, seleciona-se “randomicamente” cada pai. O processo se repete, lendo da esquerda para direita até que o tamanho requerido do filho seja alcançado. Um esquema representando o *overlap crossover* pode ser visto na figura 3.4.2.18.

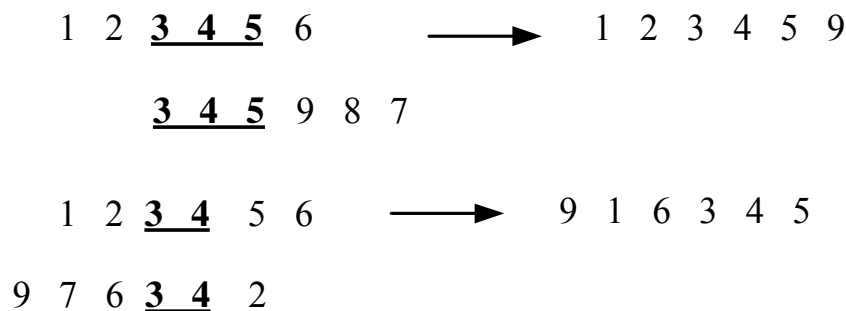


Figura 3.4.2.18: Overlap Crossover (Hart e Ross, 1999a). A partir do alinhamento, dois pais são comparados e detectados 3 genes comuns, gerando assim o descendente após o cruzamento. Na segunda interação, constatam-se dois genes em comum a partir dos pais e novo filho é gerado.

A estrutura, o funcionamento e os componentes do Algoritmo Genético, bem como as definições, serão apresentados nas sessões seguintes deste texto.

3.5.1 Estrutura do AG Desenvolvido

O algoritmo genético implementado neste trabalho utiliza os mecanismos básicos para gerar a população inicial, avaliar e aplicar os operadores de seleção, cruzamento e mutação. Foram desenvolvidos dois grupos de operadores, denominados AG1 e AG2, respectivamente. No primeiro, foram utilizados dois operadores básicos: *crossover* de um ponto e *crossover* uniforme. O segundo grupo é composto pelos seguintes operadores: *crossover* de dois pontos, *order based crossover* (OX) e o *overlap crossover*.

O diagrama de fluxo da figura 3.5.1.20 representa a estrutura simplificada do AG desenvolvido. A estrutura completa do AG pode ser vista no Anexo, sessão I.I.

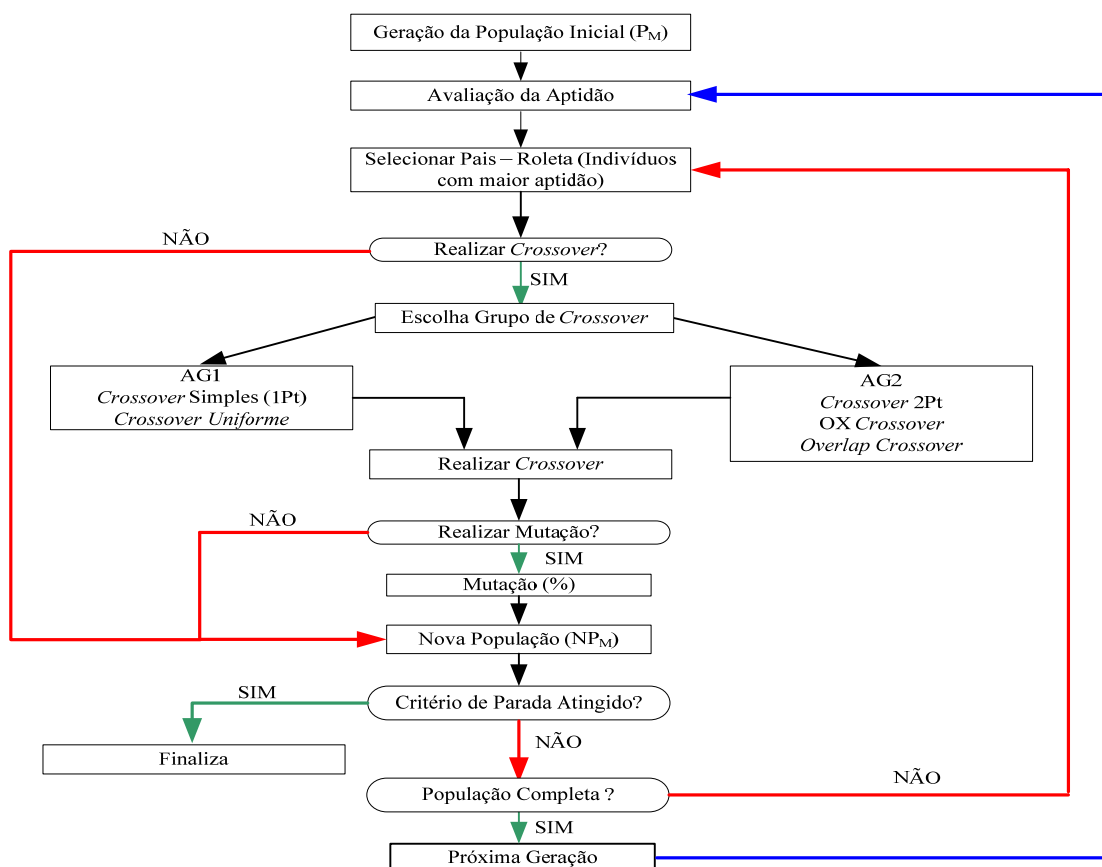


Figura 3.5.1.20: Fluxograma Representativo do AG. O Algoritmo Genético implementado para o problema proposto, gera uma população inicial P , avalia os indivíduos e os seleciona para a operação de *crossover*, podendo haver ou não mutação. Ao fim uma nova população é gerada.

A partir do diagrama apresentado podem ser observados os seguintes passos do AG, apresentado pelo pseudocódigo da figura 3.5.1.21.

```

Início           // t=geração atual, d= critério de parada , P= população M= tamanho da população.
  t := 0;           // Contador.
  Inicia_população (PM, t); // iniciar uma população de n indivíduos.
  Enquanto d = falso faça // Critério de execução do AG.
    Avaliação (PM, t); // avaliar aptidão dos indivíduos da população.
    Repita até (t = d) // testar critérios (tempo, aptidão, etc.).
      Início
        t := t + 1; // incrementar o contador de gerações.
        Seleção_dos_pais (PM, t) // selecionar os pares para cruzamento.
        Crossover (PM, t); // realizar cruzamento dos pares selecionados.
        Escolher_grupo( AG1, AG2) //Escolher grupo de Operadores.
        Mutação (PM, t); = SIM // Mutar o grupo gerado pelo cruzamento.
        Nova_População (PM, t) // Geração da nova população
      Fim;
    Selecionar_novos_individuos // Selecionar os novos indivíduos para cruzamento.
    Avaliação (PM, t); // avaliar as novas aptidões.
  Fim;
Início
  Se d = verdadeiro // teste d critério de parada
    Então finaliza // Fim da execução.
  População completa (PM) // Verifica se a população está completa.
  Nova geração // Início da nova geração.
Fim;
Fim.

```

Figura 3.5.1.21: Pseudocódigo do AG desenvolvido: A figura apresenta os passos do AG desenvolvido. Uma população inicial é gerada e avaliada por uma função de aptidão, direcionando os indivíduos selecionados para o cruzamento e mutação. O processo se repete até que uma nova população seja completada.

3.5.2 Descrição dos Componentes do AG.

O AG proposto para tratar o problema é composto pelos seguintes componentes: gerador randômico da população inicial, função de avaliação da aptidão, operadores de seleção, operadores de cruzamento e operador de mutação.

- Gerador Randômico: a população inicial (tamanho M) é gerada aleatoriamente. O processo é realizado por uma função randômica desenvolvida especificamente para este processo.

- Função de Avaliação e Função de *Fitness*: Para controlar a densidade da população é necessário diferenciar a função de avaliação e a função de aptidão. Assim, foram adotadas as seguintes funções para o problema, baseado em Gonçalves *et al.* (2002): $(F_x) = F - Aval(X)$ $(\forall X \in P)$, onde F é um limitante superior para as avaliações.

- Operador de Seleção: Existem alguns operadores de seleção aplicáveis aos AG's, como apresentado na sessão 3.4.1, neste trabalho foi utilizado a roleta. Este método seleciona os dois indivíduos com maior aptidão obtidos através de um vetor de intervalos construído a

partir do *fitness* de cada indivíduo. O intervalo do “i-ésimo” indivíduo é o seu valor correspondente no vetor. Desta forma, indivíduos com *fitness* maiores, terão um intervalo maior, e consequentemente maior probabilidade de serem escolhidos, ou seja, a seleção é tendenciosa em relação aos indivíduos mais aptos.

- Operadores de *Crossover*: No AG, foram utilizados dois grupos de operadores de cruzamento distintos, como apresentado na sessão 3.5.1. Na sessão a seguir será apresentado o funcionamento e a justificativa da utilização destes dois grupos operadores.

- Operador de Mutação: a mutação pode ou não ocorrer durante uma geração. Caso a opção seja pelo uso, uma probabilidade (%) deve ser estabelecida. A regra básica para estabelecer a probabilidade é que esta seja ($0 \leq Mt \leq 1$). A sugestão é que se use uma taxa pequena para a mutação, variando entre 0 – 1, isto porque a finalidade da mutação é diversificar uma população ou recuperar genes perdidos durante o processo de cruzamento após várias gerações.

- Critério de Parada: Foram estabelecidos os seguintes critérios de parada: 1 – Número máximo de iterações, que é o total de gerações que o AG deve avaliar, e 2 – A tolerância máxima, que é um limite prático do AG. Com este critério, pretende-se garantir que o AG não gere redundâncias ou pare de evoluir após certo tempo em execução.

3.5.3 Grupo de Operadores de *Crossover* (AG1 e AG2)

A idéia de utilizar o AG com grupos de operadores de cruzamento distintos é de testar e verificar quais operadores ou grupo de operadores produz um resultado melhor, i.é, qual grupo é mais eficiente na geração dos indivíduos que serão usados pelo SIA para tratar os escalonamentos. Também será possível verificar se existe diferença no comportamento do AG com relação aos grupos e se estas são relevantes, influenciando direta ou indiretamente nos resultados.

Cada operador de *crossover* presente em cada grupo difere um do outro em sua estrutura e funcionamento. Diante disto, alguns critérios de operação foram estabelecidos para os operadores de cada grupo.

Operadores do Grupo AG1: Aos operadores que compõem o grupo AG1 (*crossover* simples e uniforme) estabeleceu-se o seguinte critério de operação: O grupo AG1 possui somente dois operadores, portanto o critério adotado foi: para cada operador aplicou uma taxa de probabilística de 50%, assim a possibilidade do AG usar cada operador é igual. No caso do

crossover simples, uma escolha aleatória é realizada em uma posição entre 0 e 0,7 do tamanho do indivíduo. O tamanho do indivíduo será o produto entre número de *jobs* e número de máquinas. Já o *crossover* uniforme utiliza uma máscara de cruzamento para criar dois novos indivíduos a partir de dois pais. Cada filho herda o gene de um ou outro pai segundo uma probabilidade de 50%. A troca é feita apenas até 70% do cromossomo, com isso, pretende-se evitar redundâncias ou filhos com baixa aptidão.

Operadores do Grupo AG2: os operadores, *2Pt crossover*, *OX crossover* e *Overlap crossover* do grupo AG2, possuem características estruturais e comportamentais distintas. Os critérios aplicados foram os mesmos usados no PRAIS de Hart e Ross, (1999a e 1999b) e descritos na sessão 3.4.2. Por esse motivo não foi implementado um valor probabilístico fixo para cada operador, como no caso do AG1, e sim, estabelecidos condições de operações de acordo com o comportamento e funcionamento de cada operador, a saber:

- *2Pt Crossover*: Usado quando os pais não possuem genes em comum, exceto possíveis “*wild cards*”⁶;
- *OX Crossover*: Usado se os pais são permutáveis. Uma região de match é definida a partir dos elementos comuns entre os pais para que as trocas possam ser realizadas.
- *Overlap Crossover*: Usado se os pais possuem genes comuns na mesma posição. Os genes são mantidos e um pai é selecionado randomicamente para efetuar as transposições.

3.5.4 Formalização do AG: Nomenclatura, Parâmetros e Valores.

A construção de AG’s requer alguns formalismos que represente seus componentes estruturais. Neste trabalho adotou as notações descritas a seguir.

- População: representada por um conjunto de indivíduos;
- População Inicial: conjunto de indivíduos formados por gerações randômicas;
- Nova População: conjunto de indivíduos gerados a partir da população inicial e evoluídos através dos operadores de seleção, *crossover* e mutação;
- Função de Avaliação de Função e *Fitness*: usada para avaliar e calcular a aptidão de um indivíduo de uma população;

⁶ **Wild Cards:** Representa um gene genérico (comum). Pode ser sobreposto por qualquer outro gene não comum de qualquer um dos pais durante o cruzamento. Geralmente usado em escalonamentos pequenos ou incompletos.

- Tamanho da População: representa o número máximo de indivíduos que uma população possa ter;
- Número de Iterações: total de operações realizadas pelo AG para gerar uma nova população;
- Taxa de *Crossover*: parâmetro passado ao AG para a realização de *crossover*;
- Taxa de Mutação: parâmetro passado ao AG para “mutar” indivíduos e diversificar a população.
- Tolerância Máxima: usado para evitar que o AG produza indivíduos redundantes após certo tempo de execução.

A nomenclatura, os parâmetros e os valores dos componentes de acordo com a definição descrita acima são apresentados na tabela 3.5.4.4.

Tabela 3.5.4.4: Nomenclatura, Parâmetros e Valores Usados no AG.

Nomenclatura	Parâmetro	Valores
População	P	P_M
População Inicial	PI	Randômico
Nova População	NP	T_{Max}
Tamanho da População	T_{Max}	(> 0)
Número de Iterações	Ng_{Max}	(> 0)
Função de Avaliação	$Aval(X)$	$(\forall X \in P)$
Probabilidade de <i>Crossover</i>	$Cx\%$	$(0 \leq Cx \leq 1)$
Probabilidade de Mutação	$Mt\%$	$(0 \leq Mt \leq 1)$
Tolerância Máxima	To_{Max}	(> 0)

3.5.5 Dados de Entrada para o AG

Como parâmetro de entrada para Algoritmo Genético deve-se utilizar instâncias do tipo *job shop*. Existe uma grande variedade de *benchmarks* disponíveis na literatura contendo estas instâncias. Estes *benchmarks* podem ser encontrados na *OR-LIBRARY* (*Operational Research Library*), mantida por Beasley (1990). Para a entrada nos sistema, as instâncias colhidas, devem possuir o formato apresentado na tabela 3.5.5.5.

Tabela 3.5.5.5 Formato de Entrada de Dados no AG.

Linhas	(m, t)	Descrição
L1: (j x m)	3 3	Tamanho da instância
L2: (Job 0)	$(m_1, t_1) (m_2, t_2) (m_3, t_3) \dots (m_n, t_n)$	Lista de tarefas do 1º <i>job</i>
L3: (Job 1)	$(m_1, t_1) (m_2, t_2) (m_3, t_3) \dots (m_n, t_n)$	Lista de tarefas do 2º <i>job</i>
L4: (Job 2)	$(m_1, t_1) (m_2, t_2) (m_3, t_3) \dots (m_n, t_n)$	Lista de tarefas do 3º <i>job</i>
⋮	⋮	⋮
Ln: (Job n)	$(m_1, t_1) (m_2, t_2) (m_3, t_3) \dots (m_n, t_n)$	Lista de tarefas do nº <i>job</i>

A coluna “Linhas” e a coluna “Descrição” mostrada na tabela 3.5.5.5, possuem caráter informativo, assim, estes parâmetros não devem ser passados pela entrada – esta deve seguir criteriosamente os parâmetros da coluna (m, t) . A coluna “Linhas” especificamente, já está fixada na implementação.

3.5.6 Dados de Saída do AG

O resultado do processamento do AG é usado para a geração dos cenários (antígenos), ou seja, a entrada para o sistema imunológico são as seqüências de *job*, que representam os antígenos que deverão ser tratados pelos anticorpos gerados pelo SIA.

4 TRATAMENTO IMUNOLÓGICO PARA O PROBLEMA DE ESCALONAMENTO

O funcionamento dos sistemas imunes artificiais é um processo idêntico ao do sistema imune natural ou biológico, uma vez que este é baseado no comportamento e nas funcionalidades dos sistemas imunes naturais. O Capítulo 4 destaca o ambiente imunológico artificial desenvolvido para solucionar problemas de escalonamento.

4.1 Relação Entre um SIA e o Problema de Escalonamento

Para representar um SIA se faz necessário coletar o material genético para a produção dos anticorpos e conseqüentemente de suas bibliotecas. Hart e Ross (2000b) apontam o modelo apresentado por Hightower *et al.* (1995). Neste modelo, o material genético necessário para produzir uma molécula de anticorpo é armazenado em um conjunto de bibliotecas de anticorpos. Cada biblioteca é composta por conjuntos de anticorpos. Os anticorpos são produzidos por combinação randômica e pela seleção de componente de cada biblioteca.

Os trabalhos de Hart *et al.* (1998), Hart e Ross (1998) e Hart e Ross (2000a) propõem um sistema imunológico artificial capaz de produzir escalonamentos de forma rápida e eficiente para tratar conjuntos particulares de problemas. Fazendo uma analogia com a programação de uma linha de montagem de uma fábrica, os autores levantaram as seguintes questões 1 – o sistema imunológico é capaz de reagir de forma rápida e eficiente aos antígenos previamente encontrados e também a novos antígenos; e, 2 – Uma fábrica em operação é suscetível a variações em seu ambiente, necessitando assim, de alterações nos escalonamentos previamente definidos.

A partir destas questões, nota-se que todas as soluções candidatas não devem ser ignoradas e caminhos e arranjos devem ser construídos e associados a estas soluções a fim de minimizar ou resolver o problema.

A analogia feita por Hart *et al.* (1998), Hart e Ross (1998) e Hart e Ross (2000a) entre uma fábrica e um problema de escalonamento, é bastante natural, pois o processo de uma fábrica representa e exemplifica bem o ambiente de escalonamentos.

A tabela 4.1.0.6 apresenta uma relação entre os sistemas imune artificiais e um ambiente de escalonamento, destacando a nomenclatura imunológica e os arranjos do problema de escalonamento abordado.

Tabela 4.1.0.6: Relação Entre o Problema de Scheduling e o SIA (De Castro, 2001).

Sistema Imunológico	<i>Job Shop Scheduling</i>
Antígeno	Seqüência de tarefas em uma máquina particular, dado um determinado cenário.
Universo antigênico	Conjunto de antígenos que representa os possíveis cenários/contingências que podem ocorrer.
Anticorpo	Partes de uma seqüência de tarefas que são comuns a mais de uma máquina.
Ligação (<i>matching</i>)	A ligação (Ag, Ab) ocorre se os escalonamentos produzidos, utilizando as instruções dadas por um Ab e a informação fornecida por um Ag , resultarem na realização da tarefa antes do prazo final especificado pelo antígeno.
Valor de ligação	Representa o número máximo de unidades de tempo que uma tarefa pode estar atrasada em um planejamento produzido pela interação (Ag, Ab). Um planejamento perfeito possui um valor de ligação 0 e, quanto maior este valor, pior o planejamento.
Resposta Imunológica	Planejamento evoluído por um algoritmo genético.

Como observado, o modelo do sistema imunológico utilizado por Hart *et al.* (1998), Hart e Ross (1999a, 1999b) e Hart e Ross (2000a) foi baseado nos trabalhos de Hightower *et al.* (1995) e Hightower *et al.* (1996), onde o material genético a ser empregado na produção das moléculas de anticorpo está armazenado em um conjunto de bibliotecas. Hightower *et al.* (1995, 1996) utilizou um Algoritmo Genético (AG) para evoluir os componentes das bibliotecas imunológicas e geração dos escalonamentos.

O modelo proposto por Hightower *et al.* (1995, 1996) utiliza um espaço de formas do tipo inteiro, onde as moléculas são representadas por permutações dos L elementos compostos por cadeias de inteiros. Como os anticorpos representam partes de uma seqüência de tarefas, ou seja, componentes de uma das bibliotecas, seu comprimento L_2 é menor ou igual ao comprimento L_1 dos antígenos, ou seqüência de tarefas de uma máquina ($L_2 \leq L_1$). Elementos do tipo “*wild cards*” – que pode se ligar a qualquer elemento/tarefa – podem existir

nos anticorpos objetivando facilitar a ligação de cadeias incompletas. Estes elementos facilitam a ligação, mas por um outro lado, pode gerar cadeias imperfeitas ou irrelevantes.

Em seu trabalho, De Castro (2001) apresentou este modelo de ligação, que pode ser ilustrado através da figura 4.1.0.22.

Antígeno	1 2 3 4 5 6 7 8 9	Valor de Ligação	
	3 4 6 7 8 *	1	
	3 4 6 7 8 *	1	
Anticorpos	<u>3</u> <u>4</u> 6 7 8 *	10 + 1 = 11	Afinidade
	3 4 <u>6</u> <u>7</u> <u>8</u> *	15 + 1 = 16	

Figura 4.1.0.22: Codificação Inteira das Moléculas e Função de *Matching* das Cadeias (De Castro 2001). A figura apresenta os valores de ligação e afinidade dos anticorpos e antígenos. A codificação inteira para as moléculas, juntamente com a função de *matching* das cadeias apresenta uma correspondência entre os números que equivalem a um valor de ligação 5, enquanto um “*wild card*” adiciona 1 ao valor total da afinidade.

A afinidade entre duas moléculas é calculada utilizando-se uma função de afinidade, ou *matching*, que alinha as duas cadeias e verifica a correspondência elemento a elemento (De Castro 2001). Um valor de ligação ainda é calculado para cada alinhamento possível, caso as cadeias possuam comprimentos distintos.

4.2 Critérios para Desenvolvimento de um SIA

Para se construir um sistema imunológico artificial que representa o funcionamento do sistema imunológico humano deve-se criar uma estrutura básica que represente o funcionamento do sistema imunológico biológico e realizar uma análise criteriosa dos seus agentes funcionais para que a implementação do sistema seja realizada.

Um SIA requer certos critérios e cuidados para o seu desenvolvimento, como os apresentados na seção 2.3.1. Para um sistema ser caracterizado como um sistema imune é preciso que existam características marcantes da imunologia, como a representação dos anticorpos e antígenos, e ainda ser aplicado a um problema específico. A tabela 4.2.0.7 apresenta uma relação entre os sistemas imunes biológicos e os sistemas imunes artificiais, tendo como referência o JSS como caso de interesse e destacando os pontos essenciais para o desenvolvimento de um SIA.

Tabela 4.2.0.7: Comparativo Entre Sistema Imune Natural e Sistema Imune Artificial.

SISTEMA IMUNE NATURAL	SISTEMA IMUNE ARTIFICIAL
O material genético necessário para produzir um anticorpo é armazenado em 5 bibliotecas de componentes, separadas.	O material genético para produzir um anticorpo é extraído de algoritmos evolucionários ou genéticos por escolha randômica.
Um anticorpo é produzido por combinações e seleções randômicas de cada componente da biblioteca.	Um anticorpo representa um escalonamento direto ou indireto, dado por um conjunto de <i>strings</i> : ABC... (Gonçalves <i>et. al.</i> , 2002) ou um conjunto de inteiros: 012... (Hart, 1997) de tamanho JM .
Um sistema imune contém L bibliotecas, cada uma com C componentes.	Um anticorpo de um SIA consiste de l bibliotecas, cada uma com c componentes.
Os componentes podem ser usados em C^l formatos diferentes de anticorpos.	Cada componente é uma cadeia de s genes. Cada s gene possui um valor no intervalo de 0 a $j - 1$.
Um conjunto completo de anticorpos pode ser formado com conhecimento igual ao do potencial da lista de antígenos.	Um conjunto de anticorpos é formado a partir de antígenos, através de regiões de complementaridade.
Os anticorpos são produzidos para um antígeno específico ou para uma classe de antígenos, determinado pelo mecanismo de resposta imune.	O valor de (l, c, s) pode ser variável, sujeito a condições $(ls = jm)$ de maneira que o tamanho do anticorpo e expandido igualmente ao número de operações.
Os componentes em cada biblioteca são geneticamente diferentes.	Os componentes em cada biblioteca são geneticamente diferentes.

Existe uma similaridade entre as características do sistema biológico e do sistema artificial mostrados na tabela 4.2.0.7. Entre os pontos mais comuns estão o material genético (cromossomos), as células, os anticorpos e os antígenos. A nomenclatura apresentada, tanto para o sistema biológico quanto para o sistema artificial atende as necessidades básicas para representação dos dois ambientes.

4.3 Representação do Ambiente Imunológico

O ambiente imunológico é composto por um conjunto de L bibliotecas de C componentes combinadas e selecionadas randomicamente, representado por um conjunto de antígenos e anticorpos.

Para representar o ambiente imunológico é preciso seguir os critérios apresentados na sessão 4.2, onde se destaca um modelo baseado em um conjunto de componentes que será a

composição do sistema imunológico. No sistema proposto neste trabalho estas características são evidenciadas através dos escalonamentos das tarefas consistindo em conjuntos de anticorpos e antígenos.

O ambiente imunológico construído por Hart e Ross (1999a , 1999b) é constituído por um *germline*(G) – que é um centro de reprodução de anticorpos/escalonamentos. Assim como Hart e Ross (1999a , 1999b), assumiu-se que existem j *jobs* para ser escalonados. Examinando todas as possíveis combinações dos indivíduos os *jobs* resultariam em $j!$ *schedules* sendo avaliados. Embora os $c!$ genes possam ser $\geq j$, há menos do que c combinações válidas dos genes no *germline*. Como cada gene em G contém mais de um *job* selecionado, isso impossibilita imediatamente selecionar um grande número de genes para a inclusão no mesmo anticorpo.

Atribui-se a diversidade dos anticorpos produzidos pelo sistema imune natural ao mecanismo de recombinação. O mesmo mecanismo pode ser empregado para produzir uma escala diversa de escalonamentos no sistema. Segundo Hart e Ross (1999b) existem seis métodos diferentes de recombinação. Neste trabalho, usou-se somente o método de recombinação somática, sendo este apresentado mais adiante.

4.3.1 Construção do Ambiente Imunológico

A construção do ambiente imunológico requer dois componentes com funções bem definidas: estes componentes representam os antígenos e anticorpos. A definição dos antígenos e anticorpos e suas funções básicas no SIA são:

Antígeno: é construído através do processamento das instâncias de entrada. O antígeno é gerado pela perturbação do sistema.

Anticorpo: o anticorpo é gerado por *schedules* parciais ou pequenos, podendo ter em sua estrutura alelos primitivos (*) capazes de ligar-se a qualquer outro indivíduo.

4.4 O Sistema Imune Desenvolvido

O SIA desenvolvido para tratar o problema de escalonamento, especificamente o problema de JSS, seguiu alguns critérios baseados nos trabalhos encontrados na literatura.

1. Utilização de algoritmo genético para resolver o problema de JSS.

- a. O AG desenvolvido baseou-se no algoritmo genético de Fang (1994) para resolver o problema de JSS. O algoritmo avalia os indivíduos de uma dada população através de uma função de aptidão. Os indivíduos com melhor aptidão são eleitos para cruzamento e assim gerar a próxima população.
- b. Cada indivíduo na população é gerado por atribuição randômica, com o valor de cada gene definido pela tripla (l, c, s) .
2. Os antígenos representam uma sequência de *jobs* encontrados em cada máquina gerados a partir de instâncias *job shop*.
3. Para cada máquina é gerado um conjunto de 10 cenários distintos.
 - a. Um conjunto de antígenos estabelece uma população de antígenos.

O processo de geração e representação dos antígenos e consequentemente da população de antígenos em cada máquina determinam os cenários conforme ilustrados através da figura 4.4.0.23.

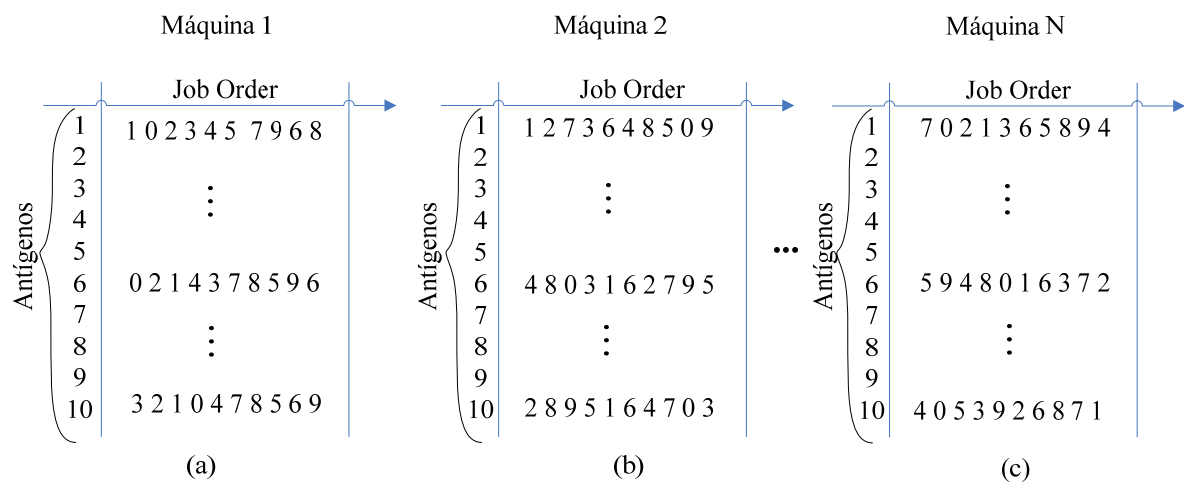


Figura 4.4.0.23: Universo de Antígenos: A figura apresenta um esquema gráfico de N populações de antígenos. Um conjunto de populações será tratado como universo. (a) Apresenta a população 1 gerada na máquina 1 com 10 cenários diferentes a partir do processamento de n jobs. (b) Apresenta a população 2 com as mesmas características da população 1. E (c) apresenta a população N no universo de antígenos, também com as mesmas características da população 1.

4. Os anticorpos são gerados randomicamente no SIA a partir dos escalonamentos obtidos da seguinte maneira:
 - a. Seleciona um antígeno obtido a partir de uma instância JSS.
 - b. Para cada antígeno selecionado.
 - i. Criar escalonamentos parciais ou pequenos.
 - ii. Calcular o *match* a partir da afinidade entre anticorpo e antígeno.

5. O sistema imunológico desenvolvido utiliza um algoritmo genético conforme exposto na sessão 3.5.1.
6. O Sistema imune desenvolvido utilizou os seguintes critérios embasados nos sistemas PRAIS e CLARISA:
 - a. PRAIS: gera um conjunto de cenários que serão usados para perturbar o sistema, estimulando assim a criação de anticorpos parciais. Estes anticorpos são calculados, alinhados aos antígenos e calculando o *match*.
 - b. CLARISA: um *germline* é construído a partir das saídas do PRAIS, utilizando os critérios de classificação e recombinação e recombinação somática, formando assim novos escalonamentos, ditos completos.

A figura 4.4.0.24 apresenta a arquitetura do Sistema Imune Artificial desenvolvido.

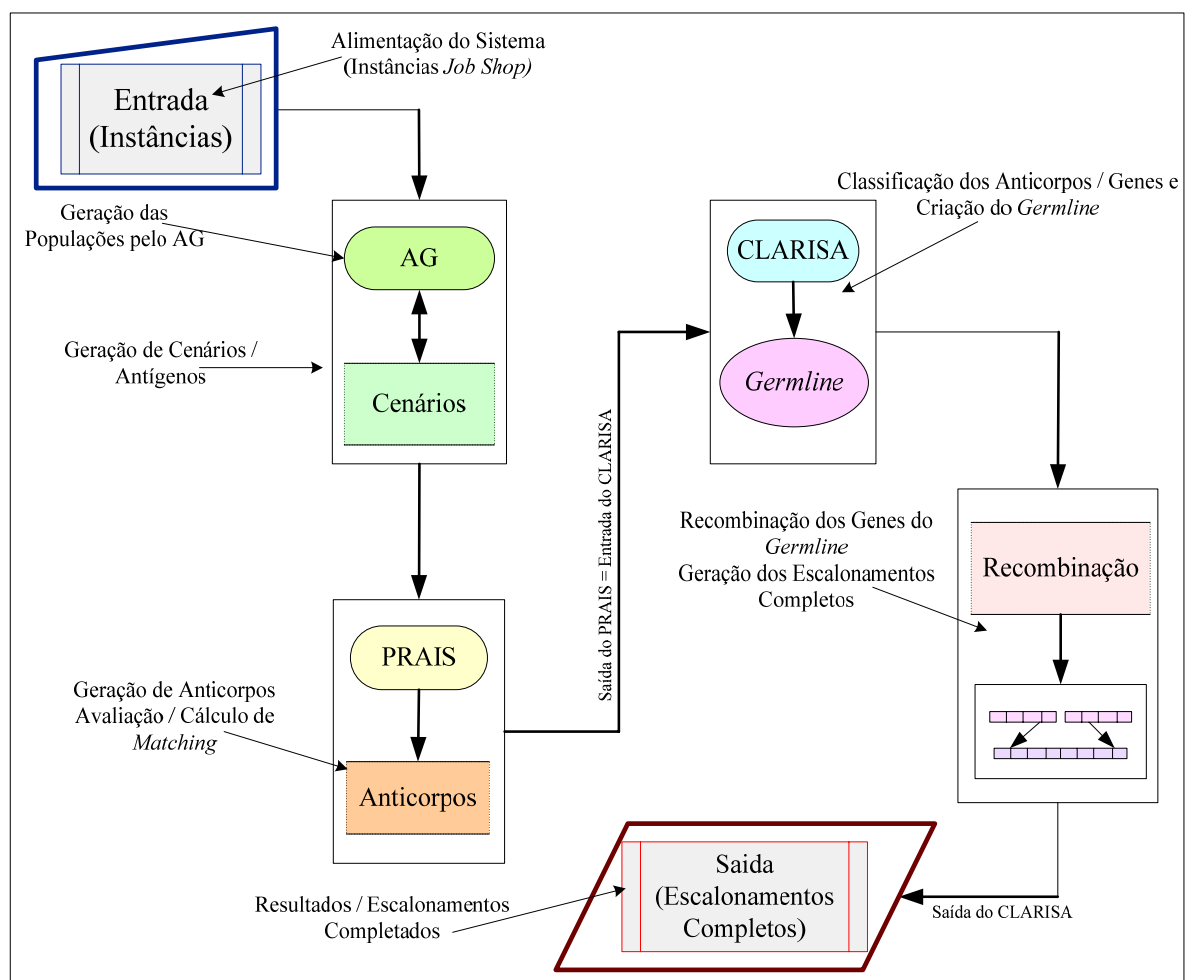


Figura 3.4.0.24: Arquitetura do SIA Desenvolvido: A partir de uma entrada (instância JSS), o AG gera os cenários para o PRAIS, onde inicia a produção de anticorpos. A Saída do PRAIS é a entrada para o CLARISA, onde é formado o *germline* para a recombinação e construção dos escalonamentos completos.

4.4.1 Representação dos Antígenos do SIA

A representação de um antígeno é dada pelo conjunto de tarefas escalonadas no ambiente, ou seja, um antígeno representa uma seqüência de *jobs* em uma dada máquina.

Um antígeno é representado em Hart (2002), como um conjunto de tarefas completadas para um *job*. Em outras palavras, um antígeno é simplesmente uma lista de dados ordenada representada por inteiros. Esta lista é obtida a partir de um conjunto de tarefas escalonadas no ambiente, a partir de uma instância do tipo *job shop*, como apresenta a tabela 4.1.2.8.

Tabela 4.4.1.8: Exemplo de uma Entrada (Instância) 4X4.

Instância de Entrada (4X4)				
4 4	(m,t)	(m,t)	(m,t)	(m,t)
Job 0	(2,88)	(0,68)	(3,94)	(1,99)
Job 1	(0,72)	(3,50)	(2,69)	(3,75)
Job 2	(3,83)	(1,61)	(0,83)	(2,65)
Job 3	(1,94)	(2,68)	(1, 61)	(0, 99)

O sistema imune desenvolvido altera os tempos fornecidos pelas instâncias *job shop* – critério de perturbação, gerando uma nova instância. Esta nova instância é a entrada para o AG. Os melhores indivíduos são então selecionados e extraídos do genótipo de cada indivíduo os *jobs sequence* que serão usados para compor o cenário de cada máquina. A Tabela 4.4.1.9 apresenta um exemplo de alteração dos tempos de processamentos da tabela 4.4.1.8. A partir daí o sistema gera os escalonamentos construindo os cenários (população de antígenos) para uma determinada máquina.

Tabela 4.4.1.9: Instância com Tempos Modificados (4X4)

Instância de Entrada (4X4)				
4 4	(m,t)	(m,t)	(m,t)	(m,t)
Job 0	(2, 85)	(0, 66)	(3, 91)	(1, 92)
Job 1	(0, 68)	(3, 55)	(2, 72)	(3, 71)
Job 2	(3, 84)	(1, 65)	(0, 82)	(2, 60)
Job 3	(1, 95)	(2, 72)	(1, 65)	(0, 92)

Esta estratégia é utilizada com o propósito de viabilizar uma ferramenta que capture a idéia de “ensinar” o sistema a se comportar, face a pequenas mudanças que possam ocorrer no ambiente produtivo.

No PRAIS, cada antígeno é uma cadeia inteira de tamanho j , onde j é o número de *jobs* escalonados em uma dada máquina m e cada elemento j_i da cadeia inteira representa a identidade do *job* escalonado. O conjunto de antígenos desenvolvido a partir de um escalonamento em particular é direcionado para a população de antígenos.

No sistema imune desenvolvido neste trabalho, foram adotados os mesmos critérios de representação de antígenos desenvolvidos no PRAIS.

4.4.2 Representação dos Anticorpos do SIA

Assim como nos trabalhos de Hart e Ross (1999a, 1999b), um anticorpo é representado por uma sequência de inteiros de tamanho l , onde $l < j$. O inteiro l é significativamente menor que j , desta forma, usando-se um anticorpo pequeno, fica mais fácil manter um maior grau de identificações entre os anticorpos e antígenos, estabelecendo assim um padrão constituído por pequenas sequências de *jobs*. A vantagem de se usar pequenas sequências de *jobs* está diretamente relacionada à alta aptidão que um *matching* de um anticorpo parcial possa ter.

Um anticorpo pode conter um conjunto qualquer de alelos primitivos representados por (*) para facilitar a combinação de resultados ou escalonamentos incompletos. Eles também podem ser usados onde *jobs* comuns não são consecutivos ou não completos, sendo possível serem observados padrões na forma $a^{***}b$.

Uma população inicial de anticorpos é gerada aleatoriamente, a partir da recombinação destas sequências, não permitindo a duplicação de *jobs*.

4.4.3 Avaliação da Aptidão Através do Match Score

A avaliação da aptidão dos indivíduos é obtida calculando o *match score*. Para calcular o *match* é preciso alinhar e combinar um anticorpo com um antígeno. Um alinhamento possível é aquele onde cada gene do anticorpo é alinhado com cada gene do antígeno obedecendo a seguinte restrição: o anticorpo selecionado para o alinhamento, deve ser significativamente menor que o antígeno. O processo pode ser representado pelos seguintes passos:

1. Selecionar um antígeno em uma dada população, gerada através de uma instância *job shop*;
2. Selecionar anticorpos obtidos através de escalonamentos parciais e alinhá-los com o antígeno selecionado;
3. Verificar a afinidade (antígeno, anticorpo) comparando cada gene;
4. Verificar sequência (alelos) comuns de cada gene;
5. Atribuir valor 5 para cada posição comum (afinidade) que possua uma afinidade (sequência) ≥ 2 ;
6. Somar cada posição comum, obtendo assim o *match score* do gene;
7. Os *jobs* comuns não consecutivos devem ser substituídos por alelos primitivos ou *wild cards*, representado por (*).
8. Para os genes comuns não consecutivos, atribuir valor 1, evitando que esse genes evolua.

A figura 4.4.3.25 apresenta o processo de calculo de aptidão de um anticorpo alinhado a um antígeno.

Antígeno	1	2	3	4	5	6	7	8	9	Match Score
			3	4	6	7	8			0
Anticorpos			3	4	6	7	8			0
			<u>3</u>	<u>4</u>	6	7	8			10
					3	4	<u>6</u>	<u>7</u>	<u>8</u>	15

Figura 4.4.3.25: Alinhamento e Cálculo de Match (Hart e Ross 1999b). O cálculo da aptidão (*match score*) através se da através da afinidade de um anticorpo com um antígeno. O anticorpo é alinhado a um antígeno e suas posições são comparadas. Cada posição alinhada recebe valor 5. A soma das posições comuns representa a aptidão do anticorpo. O individuo com maior valor de mach é selecionado para compor o *germline*.

4.4.4 Produzindo uma Resposta Imune

A resposta imune pode ser primária ou secundária conforme descrito na sessão 2.2.3. Construiu-se somente o mecanismo de resposta primária, o qual ocorre após a inserção do antígeno no ambiente pela primeira vez.

Assim como Hart e Ross (1999b), usou-se o algoritmo genético para produzir um conjunto de anticorpos (sequência de *jobs*). A inserção de antígenos no ambiente dispara a produção de anticorpos para este antígeno específico, determinando assim o mecanismo de resposta imune primária.

O processo de resposta imunológica é análogo ao mecanismo de resposta imune do sistema natural, onde um conjunto de anticorpos é produzido para um agente estranho inserido no ambiente.

O processo de resposta imune secundária não foi desenvolvido, já que não foi criado um mecanismo de memória imunológica, onde o sistema armazenaria as informações de um determinado antígeno e quando este ocorrer novamente no sistema uma resposta específica e mais intensa é produzida para este antígeno.

4.5 Construção do *Germline*

Construir um antígeno específico é a maneira mais correta para construir um *germline*, pois permite associar os anticorpos do *germline* a somente os anticorpos que correspondem ao antígeno. O processo próprio é mais eficiente, e produz melhores resultados, no entanto é mais difícil produzir estes antígenos diretamente num sistema, como o CLARISA. No trabalho de Hart e Ross (1999b) foi construído um simples e mais geral *germline* usando as saídas do PRAIS, critério também adotado neste trabalho. Em Hart e Ross (1999b) podem ser observados três passos importantes na construção de um *germline*, como a seguir:

Considere um antígeno no qual consiste uma sequência parcial de *jobs* em uma máquina. Considere ainda as informações a respeito dos *jobs* e as operações que devem ser escalonados nessa máquina.

1. Defina um conjunto universal **U** para ser o conjunto de todos os *jobs* que serão processados na máquina.

2. Defina o conjunto **A** para ser o subconjunto dos *jobs* em **U** que já tenha sido processado. Seja **S** o conjunto de anticorpos de comprimento l , gerados pelo PRAIS.

- i. Selecione de **S** todos os anticorpos **Ab** que consistem unicamente em *jobs* ainda não escalonados e adicione-os no *germline* **G**.
- ii. Selecione de **S** todos os anticorpos **Ab** que podem realizar sobreposições (*overlap*) com **A**. Um *overlap* ocorre se os primeiros n *jobs* de **Ab** forem iguais aos últimos n *jobs* em **A**, onde $n \leq l$, e os $(l - n)$ *jobs* restantes em **Ab** não ocorrem em **A**. Adicione **Ab** ao *germline* **G**.
- iii. Adicione em **G** os *jobs* que ainda não foram incluídos, e existe em **U**.

Estes passos alimentam o *germline* com informações suficientes para terminar o escalonamento definido pelo antígeno e pode agora ser usado para produzir anticorpos, *i.é*, *schedules*. Neste caso, cada anticorpo de **S** adicionado em **G**, agora é referido como um gene no *germline*.

A figura 4.5.0.26 apresenta um esquema de um *germline* **G** com c componentes de tamanho ℓ .

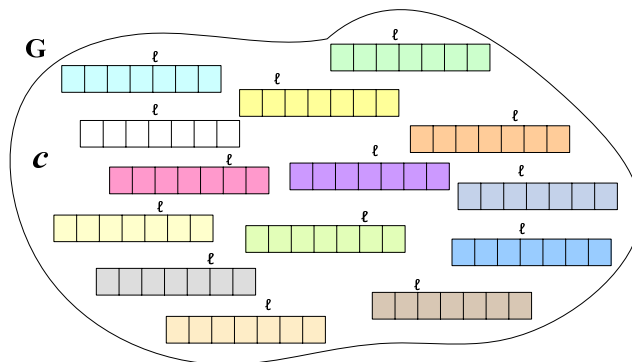


Figura 4.5.0.26: Germline. A figura apresenta um germline (**G**), com c componentes de tamanho ℓ . As diferentes cores representam o processo de diversificação dos anticorpos no *germline*.

4.5.1 Mecanismo de Recombinação Somática

A diversidade dos anticorpos é obtida através de mecanismos de recombinação e mutação somática a partir de um número relativamente pequeno de segmentos gênicos. Além disso, o processo que gera essa diversidade assegura que cada célula B produza imunoglobulinas de uma única especificidade antigênica. Desta forma a diversidade dos anticorpos no *germline* é mantida.

De acordo com Hart e Ross (1999a), a formação do germline pode ser feita por seis diferentes tipos de recombinação. Neste trabalho será usada somente a recombinação somática, que pode ser dividida em: recombinação imperfeita e recombinação por sobreposição (*overlap*), segundo Hart e Ross (1999a).

Recombinação imperfeita: a recombinação dos genes pode resultar em uma recombinação imperfeita a partir dos erros gerados durante o processo de recombinação que são codificados por segmentos.

Recombinação por Sobreposição (*overlap*): os genes combinam com qualquer outro por *overlap* ou com genes faltantes. O resultado é uma recombinação mais curta dos genes combinados.

O processo de recombinação somática pode ser visualizado na figura 4.5.1.27 a e b, apresentada abaixo.

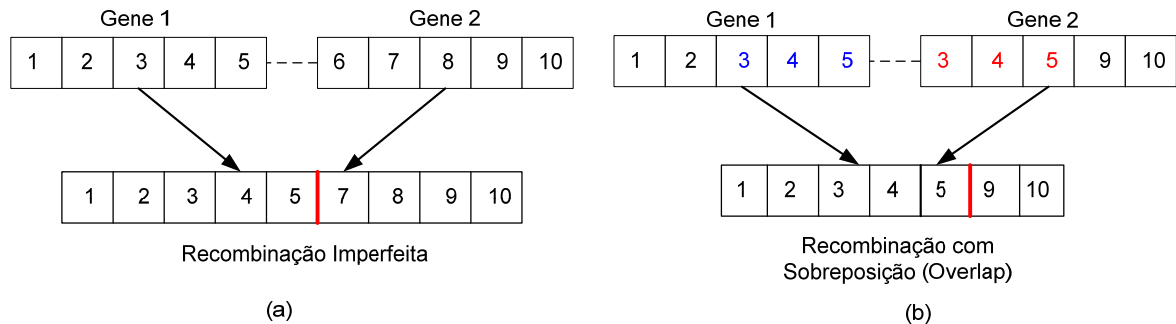


Figura 4.5.1.27: Recombinação Somática (Hart e Ross 1999b). (a) Representa o processo de recombinação somática imperfeita. (b) Representa o processo de recombinação através de *overlap*.

4.6 Gerando Escalonamentos

No sistema imune artificial (SIA), apresentado por Hart *et.al*, (1998), os anticorpos são descritos representando um escalonamento indireto, que consiste em uma cadeia $1,2,3,...n$ de tamanho jm .

No SIA desenvolvido, estas características também estão presentes tendo em vista que o sistema imune proposto é fundamentado com base no PRAIS e no CLARISA desenvolvidos por Hart e Ross (1999b). Algumas características do Algoritmo Genético Híbrido para Escalonamentos desenvolvido por Gonçalves *et al.*(2002) e do *Immune System for Scheduling* (SCHED1-IS e SCHED2-IS) desenvolvido por Hart (2002) também foram incorporadas no SIA desenvolvido deste projeto.

4.6.1 Processo de Geração de Escalonamentos

O processo de geração escalonamentos se dá a partir de um antígeno qualquer de tamanho j fornecido ao sistema. Para este, o sistema gera um conjunto de anticorpos dado por *schedules* parciais.

Neste trabalho especificamente, usou-se os seguintes procedimentos:

- Como entrada para o sistema utilizou-se instâncias *job shop*. Estas entradas, inicialmente tiveram seus tempos modificados;
- Utilizou-se AG's para gerar uma população inicial e as próximas populações, garantindo assim, o processo evolutivo das populações;

- As características imunológicas foram aplicadas através dos procedimentos do PRAIS para fornecer uma saída, e do CLARISA na construção do *germline* e na recombinação dos genes para produzir os escalonamentos completos, da seguinte maneira:

- A produção de anticorpos se dá através de escalonamentos parciais a partir de um antígeno.
- A aptidão dos anticorpos é obtida calculando-se o *fitness* pelo algoritmo de *matching*.

A partir dos genes do *germline* (escalonamentos parciais) foi possível reconstruir os escalonamentos completos para tratar o problema JSS.

5 TESTES E RESULTADOS

Neste Capítulo serão apresentados os testes numéricos realizados e seus respectivos resultados. Como observado na seção 4.4.1, nosso interesse básico com o desenvolvimento do SIA para um ambiente de escalonamento, é analisar sua capacidade de reação, face a pequenas mudanças que possam ocorrer no ambiente produtivo. O capítulo irá apresentar ainda a descrição dos critérios adotados para a aplicação dos testes, além dos recursos utilizados para desenvolver e testar o SIA. Os problemas e propostas encontrados para desenvolver o sistema imune, serão apresentados ainda neste capítulo.

5.1 Geração de Testes

Com o propósito de simular situações reais de produção, utilizou-se como entradas para o sistema, algumas instâncias encontradas na literatura. Estas instâncias então disponíveis por WWW em ORLIB (*Operational Research Library*), mantida por Beasley (1990). As instâncias utilizadas pertencem aos seguintes *benchmarks*: TA02 de Taillard (1993), FT06 e FT10 de Fisher e Thompson (1963), ABZ05 de Adams *et al.* (1998) e LA02 de Lawrence (1984).

As instâncias testadas serão disponibilizadas no anexo desta dissertação, juntamente com o resultado dos testes realizados para cada uma.

5.1.1 Parâmetros Analisados

Para todos os testes realizados foram fixados parâmetros com valores considerados razoáveis. Os mesmos parâmetros foram aplicados a todas as instâncias testadas, e que podem ser observados através da tabela 5.1.1.10.

Tabela 5.1.1.10: Parâmetros Analisados

Nomenclatura	Parâmetro	Valores
Tamanho da População	P_{Max}	100
Probabilidade de <i>Crossover</i>	Cx (%)	$(0 \geq Cx \leq 1)$
Probabilidade de Mutação	Mt (%)	$(0 \geq Mt \leq 1)$
Número de Iterações	Ni_{Max}	250
Tolerância Máxima	To_{Max}	20

A descrição dos parâmetros utilizados são apresentados a seguir:

- Tamanho da População: define o tamanho da população a ser gerada;
- Probabilidade de *Crossover*: usada para gerar os indivíduos de cada população a partir da população inicial;
- Probabilidade de Mutação: usada para evoluir e diversificar indivíduos de uma população;
- Número de Iterações: condição de parada, este parâmetro estabelece o número máximo de gerações para cada execução;
- Tolerância Máxima: estabelece um limite máximo de gerações para o AG.

5.1.2 Cenários de Teste

Os cenários de teste estão condicionados ao tamanho das instâncias *job shop* a serem utilizadas. Os testes foram realizados para um conjunto de instâncias de diferentes *benchmarks*. A tabela 5.1.2.11 apresenta as instâncias utilizadas para os testes e seus respectivos *benchmarks*.

Tabela 5.1.2.11: *Benchmarks* e Instâncias Utilizadas

<i>Benchmarks</i>	Instâncias	Tamanho (JXM)	<i>Makespan</i> Original
Adams, Ballas e Zawack	ABZ05	10X10	1234
Fisher e Thompson	FT06	06X06	55
Fisher e Thompson	FT10	10X10	930
Lawrence	LA02	10X05	655
Taillard	TA02	15X15	1244

Cada *benchmarks* possui uma variedade considerável de instâncias, as quais também poderiam servir como entrada para o sistema. No entanto, utilizou-se somente as especificadas na tabela acima.

5.2 Recursos Computacionais

A construção e a execução do sistema imunológico impõem exigências de recursos computacionais de desempenho considerável. Assim, para desenvolver e executar o SIA consumiu-se alguns recursos computacionais, talvez não o ideal, mas suficientes para atender os requisitos/exigências mínimas para este tipo de trabalho. Os itens a seguir mostram os recursos de *hardware* e *software* utilizados.

5.2.1 Recursos de *Software*

Os recursos de softwares utilizados para desenvolvimento e execução do sistema foram os seguintes.

- Recursos de Desenvolvimento: utilizou-se a linguagem de programação JAVA com o ambiente de desenvolvimento *NETBeans*⁷ 5.0 rodando na máquina virtual JAVA: *J2SE Development Kit 5.0*⁸ *Update 7.0* e *J2SE Runtime Environment 5.0*⁹ *Update 7.0*.
- Recursos de Execução: O sistema foi executado na plataforma operacional *Windows XP Professional*¹⁰ e *LINUX Fedora Core 1.0*¹¹.

5.2.2 Recursos de *Hardware*

Os recursos de *hardware* utilizados para executar o sistema são especificados a seguir:

- Processador: foi usado o processador Intel *Pentium 4*¹² com *Clock* de 3.0 GHz;
- Memória: a máquina utilizada para executar o sistema possui memória principal DDR SDRAM de 512 MB.

⁷ *NETBeans* 5.0 é marca registrada da Sun Microsystems, Inc.

⁸ *J2SE Developemnt Kit 5.0* é marca registrada da Sun Microsystems, Inc.

⁹ *J2SE Runtime Environment 5.0* é marca registrada da Sun Microsystems, Inc.

¹⁰ *Windows XP Professional* é marca registrada da Microsoft Corporation.

¹¹ *LINUX Fedora Core 1.0* é marca registrada da Red Hat, inc.

¹² *Pentium 4* é marca registrada da Intel Corporation.

5.3 Testes Aplicados

Para cada instância *job shop* usada como entrada para o sistema aplicou-se os testes através do escalonamento gerado pelo algoritmo genético (AG1 e AG2). Os parâmetros foram fixados e os mesmos foram aplicados tanto para o algoritmo genético com os operadores do AG1 quanto para o AG2. Todas as instâncias foram executadas nos dois AG e com os mesmos parâmetros. A idéia de se utilizar os parâmetros fixos é de verificar o comportamento do sistema diante de diferentes classes do problema. Estabeleceu-se como objetivo minimizar o *makespan*, com relação ao valor original, ou o melhor valor conhecido de cada classe testada.

Na tentativa de se obter melhores resultados, foram realizados testes utilizando somente um operador de cruzamento com o AG1 (*Crossover Uniforme*) e com o AG2 (*Order Based crossover*). As sessões a seguir apresentaram todos os resultados obtidos com os testes realizados.

Neste trabalho não serão apresentados os testes realizados e os resultados obtidos somente com o Algoritmo Genético (AG1 e AG2), embora o sistema esteja preparado para realizá-los. O motivo principal é que o interesse desta pesquisa é verificar o tratamento imunológico do SIA frente às mudanças ocorridas no ambiente produtivo.

5.4 Resultados Obtidos

Para facilitar o entendimento e a análise dos resultados obtidos, serão apresentados primeiramente os resultados de cada classe/instância de acordo com os testes aplicados. As sessões a seguir trazem os resultados alcançados para cada uma dessas entradas. O ponto comum é que serão apresentados em conjunto os resultados obtidos com AG1 e com AG2 a partir dos testes realizados. Ao fim, apresenta-se o resultado de todas as instâncias, para realizar uma análise dos resultados obtidos e compará-los.

5.4.1 Resultados Obtidos para a Instância FT06

A tabela 5.4.1.12 mostra os resultados obtidos para esta instância.

Tabela 5.4.1.12: Resultados dos Testes Aplicados à Instância FT06

Instância FT06		Tamanho (06X06)		Melhor <i>Makespan</i> Conhecido = 55	
Resultados Utilizando Todos Operadores do AG1 e AG2					
Gerações	Nr. de Iterações	Melhor <i>Makespan</i>	Processamento (T/ms)	Soluções Viáveis	
SIA+ AG1	250	59	6429	10	
SIA + AG2	250	59	6930	9	
Resultados Utilizando Somente 1 Operador do AG1 e AG2					
Gerações	Nr. de Iterações	Melhor <i>Makespan</i>	Processamento (T/ms)	Soluções Viáveis	
SIA+ AG1	250	55	24450	12	
SIA + AG2	250	55	37993	8	

5.4.2 Resultados Obtidos para a Instância FT10

A instância FT10 (10X10) é classificada como um problema de difícil solução. Esta instância representa o cenário principal de teste que se deseja aplicar para o SIA. O resultado dos testes feitos são apresentados na tabela 5.4.2.13.

Tabela 5.4.2.13: Resultados dos Testes Aplicados à Instância FT10

Instância FT10		Tamanho (10X10)	Makespan Ótimo Conhecido = 930	
Resultados Utilizando Todos Operadores do AG1 e AG2				
Gerações	Nr. de Iterações	Melhor Makespan	Processamento (T/ms)	Soluções Viáveis
SIA+ AG1	250	1332	29079	7
SIA + AG2	250	1332	30288	11
Resultados Utilizando Somente 1 Operador do AG1 e AG2				
Gerações	Nr. de Iterações	Melhor Makespan	Processamento (T/ms)	Soluções Viáveis
SIA+ AG1	250	1163	67176	9
SIA + AG2	250	1175	103893	11

5.4.3 Resultados Obtidos para a Instância ABZ05

Assim como a instância FT10, a instância ABZ05 representa o cenário que se julga ideal para testar o sistema imune, por conter a estrutura de 10 *jobs* para 10 máquinas. A tabela 5.4.3.14 apresenta os resultados obtidos para esta instância.

Tabela 5.4.3.14: Resultados dos Testes Aplicados à Instância ABZ05

Instância ABZ05		Tamanho (10X10)	Makespan Ótimo Conhecido = 1234	
Resultados Utilizando Todos Operadores do AG1 e AG2				
Gerações	Nr. de Iterações	Melhor Makespan	Processamento (T/ms)	Soluções Viáveis
SIA+ AG1	250	1558	25186	6
SIA + AG2	250	1558	25847	8
Resultados Utilizando Somente 1 Operador do AG1 e AG2				
Gerações	Nr. de Iterações	Melhor Makespan	Processamento (T/ms)	Soluções Viáveis
SIA+ AG1	250	1313	66079	6
SIA + AG2	250	1338	100707	7

5.4.4 Resultados Obtidos para a Instância TA02

Durante todo o desenvolvimento utilizou-se a instância TA02 para testar o funcionamento do sistema. Trata-se de uma instância com um parâmetro maior (15X15) e conseqüentemente mais complexa. O objetivo de usar esta instância além de efetuar os testes padrões, permite testar o sistema com um problema maior. Os testes da instância são apresentados na tabela 5.1.4.15

Tabela 5.4.4.15: Resultados dos Testes Aplicados à Instância TA02

Instância TA02		Tamanho (15X15)	Makespan Ótimo Conhecido = 1244	
Resultados Utilizando Todos Operadores do AG1 e AG2				
Gerações	Nr. de Iterações	Melhor Makespan	Processamento (T/ms)	Soluções Viáveis
SIA+ AG1	250	1716	70402	8
SIA + AG2	250	1716	70696	4
Resultados Utilizando Somente 1 Operador do AG1 e AG2				
Gerações	Nr. de Iterações	Melhor Makespan	Processamento (T/ms)	Soluções Viáveis
SIA+ AG1	250	1579	139573	5
SIA + AG2	250	1579	209558	4

5.4.5 Resultados Obtidos para a Instância LA02

Os problemas da classe LA são os mais conhecidos na literatura. Trata-se de problemas complexos e com grandes variações de tamanho. Escolheu-se uma instância considerada pequena desta classe (10X5). A tabela 5.4.5.16 apresenta os resultados obtidos para a instância LA02.

Tabela 5.4.5.16: Resultados dos Testes Aplicados à Instância LA02

Instância LA02	Tamanho (10X05)	Makespan Ótimo Conhecido = 655		
Resultados Utilizando Todos Operadores do AG1 e AG2				
Gerações	Nr. de Iterações	Melhor Makespan	Processamento (T/ms)	Soluções Viáveis
SIA+ AG1	250	924	16672	82
SIA + AG2	250	886	18809	81
Resultados Utilizando Somente 1 Operador do AG1 e AG2				
Gerações	Nr. de Iterações	Melhor Makespan	Processamento (T/ms)	Soluções Viáveis
SIA+ AG1	250	762	34681	79
SIA + AG2	250	770	52148	77

5.4.6 Análise Comparativa dos Resultados

Nesta seção será apresentada a análise comparativa dos resultados obtidos para todas as instâncias, que poderão ser observados a partir da tabela 5.4.6.17, para os resultados utilizando todos os operadores do AG1 e do AG2 e a partir da tabela 5.4.6.18, para os resultados utilizando somente um operador do AG1 (Uniforme) e do AG2 (*Order Based*).

Tabela 5.4.6.17: Análise dos Resultados – Testes com Todos Operadores

Análise dos Resultados Obtidos					
Resultados					
Instâncias	Ótimo Conhecido	Melhor <i>Makespan</i>		Desvio (%)	
		SIA+ AG1	SIA + AG2	SIA+ AG1	SIA + AG2
FT06	55	59	59	7,01	7,01
FT10	930	1332	1332	43,00	43,00
ABZ05	1234	1558	1558	26,05	26,05
TA02	1244	1716	1716	37,32	37,32
LA02	655	924	886	41,10	34,37

Tabela 5.4.6.18: Análise dos Resultados – Testes com um Operador

Análise dos Resultados Obtidos					
Resultados					
Instâncias	Ótimo Conhecido	Melhor <i>Makespan</i>		Desvio (%)	
		SIA+ AG1	SIA + AG2	SIA+ AG1	SIA + AG2
FT06	55	55	55	0,0	0,0
FT10	930	1163	1175	25,05	26,34
ABZ05	1234	1313	1338	6,40	8,43
TA02	1244	1579	1579	26,93	26,93
LA02	655	762	770	16,34	17,56

5.4.7 Considerações Sobre os Testes Realizados e os Resultados Obtidos

Os testes revelaram resultados satisfatórios (dentro do esperado) que serviram como base para analisar o desempenho do SIA para as classes de problemas testadas no ambiente produtivo. Diante dos resultados obtidos, como apresentado nas seções anteriores foi possível avaliar o desempenho do sistema imune, onde se destacam os seguintes pontos para os testes realizados utilizando todos os operadores do AG1 (Simples e Uniforme) e todos operadores do AG2 (2 Pontos, *Overlap* e *Order Based*):

- O SIA produziu os mesmos *makespan* para os dois AG's, exceto para a instância LA02;
- Obteve-se um melhor tempo de processamento com o AG1 em todos os casos;
- O número de soluções viáveis se equivale, não apresentando uma diferença relevante entre o AG1 e o AG2. A instância FT10 foi a que apresentou a maior diferença;
- O SIA mostrou-se eficiente, reagindo às pequenas mudanças induzidas no ambiente produtivo, tanto para o AG1, quanto para o AG2.

A seguir destacam-se os pontos analisados com os resultados utilizando somente um operador do AG1(uniforme) e 1 operador do AG2 (*Order Based*).

- O SIA produziu resultados melhores utilizando o AG1 para as instâncias FT10, ABZ05 e LA02;
- O SIA produziu os mesmos resultados tanto com o AG1, quanto com o AG2, para as Instâncias FT06 e TA02;
- O SIA atingiu o ótimo para a instância FT06 com os dois AG's;
- Obteve-se um melhor tempo de processamento com o AG1 em todos os casos;
- O número de soluções viáveis se equivale, não apresentando uma diferença relevante entre o AG1 e o AG2;
- O SIA mostrou-se eficiente, reagindo às pequenas mudanças induzidas no ambiente produtivo com os dois AG's;

Um ponto positivo no sistema desenvolvido é que ele permite o tratamento de problemas de escalonamentos, usando não somente o SIA, mas também o algoritmo genético, com o AG1, e com o AG2, sem tratamento imunológico. Isto faz do sistema desenvolvido

uma ferramenta para geração de escalonamentos factíveis, para os mais diversos problemas encontrados na literatura.

E uma próxima etapa, pretende-se construir algumas rotinas para tentar produzir resultados melhores. Deseja-se também realizar novos testes, utilizando instâncias maiores com diferentes composições de máquinas e *jobs*.

6 CONSIDERAÇÕES FINAIS

Várias áreas da computação científica suportam a aplicação de sistemas imunes, como por exemplo: otimização combinatória, inteligência artificial, reconhecimento de padrões, desenvolvimento orientado a componentes, agentes inteligentes e modelagem matemática e computacional.

Os sistemas imunes podem oferecer soluções para os problemas presentes nestas áreas por apresentarem características particulares, como: paralelismo, aprendizagem e processamento distribuído. Estas características dotam os sistemas imunes de portabilidade e robustez, possibilitando sua aplicação em vários seguimentos de problemas computacionais. As características imunológicas dos SIA, as quais são baseadas nas representações dos sistemas imunes biológicos e em seus componentes, possibilitam uma forte interação para o tratamento destes problemas.

A partir dos testes aplicados e dos resultados obtidos, constatou-se que os SIA juntamente com os Algoritmos Genéticos podem ser usados para o tratamento de problemas do tipo *Job Shop Scheduling*. Embora os resultados alcançados pelo sistema não tenham atingido a solução ótima conhecida (exceto ft06), este mostrou-se eficiente para o tratamento de pequenas mudanças feitas no ambiente produtivo.

Os sistemas imunes artificiais são estudados há pouco mais de uma década. Durante este tempo, vários trabalhos foram propostos e desenvolvidos para tratar os mais diversos problemas. Os avanços tecnológicos e o surgimento de novas metodologias e paradigmas envolvendo os Sistemas Imunes Artificiais e seus componentes trás grandes perspectivas para a solução de problemas computacionais e de otimização, colocando os SIA no limiar das pesquisas científicas.

Alguns esforços estão sendo gastos para desenvolver ferramentas e metodologias que represente e modele os componentes do sistema imunológico natural. Estes estudos são fundamentais para o avanço dos sistemas imunológicos artificiais possibilitando a sua aplicação em ambientes reais.

Durante a realização desta pesquisa deparou-se com alguns problemas eminentes com relação ao desenvolvimento do SIA e sua aplicação ao problema de *scheduling*. Um dos pontos observados é de que outras heurísticas podem ser associadas ao sistema imune para melhorar um resultado encontrado.

A construção de uma memória imunológica é essencial para que o sistema desenvolva a resposta imune secundária, disparando assim, uma resposta direcionada e muito mais intensa a um determinado antígeno. Assim, em fases posteriores pretende desenvolver o mecanismo de resposta imunológica, visando uma melhoria nos resultados do SIA.

A falta de uma ferramenta para modelar o ambiente imunológico e os seus componentes também foi um fator a ser considerado. Diante dos pontos observados, propõe-se construir uma ferramenta ou linguagem para modelagem dos SIA.

Uma outra linha a ser seguida, e que pretende ser estudada em uma nova oportunidade é o desenvolvimento de um Sistema Imune Artificial para Problemas de Escalonamento Baseado em Agentes Inteligentes. Isto se justifica, considerando o poder dos agentes inteligentes, que aliados aos SIA, podem proporcionar um ambiente dinâmico, customizado e com certa autonomia.

7 REFERÊNCIAS BIBLIOGRÁFICAS

As Referências Bibliográficas utilizadas no desenvolvimento desta dissertação constituem um rico acervo de consulta e uma importante contribuição ao aprofundamento do conhecimento que buscou-se produzir nesta área do conhecimento.

- [01] **ADAMS, J., BALAS, R., ZAWACK, D.;** The Shifting Bottleneck Procedure for Job Shop Scheduling; Management Science, 1998.
- [02] **ALONSO, A. M. G., RUBIO, L. M.;** Introducción al Sistema Inmune. Inmunidad Natural Y Adquirida; Cap. 03; In: Rinitis Alérgica: Mecanismo y Tratamiento. MRA Ediciones; Barcelona; 2005.
- [03] **BARRAZA, M .D. Z.;** Una Introducción a los Sistemas Inmunes Artificiales; Relatório Técnico; Congreso de Optimización; UTFSM; Valparaíso; 2004.
- [04] **BEAN, J. C;** Genetic Algorithms and Random Keys for Sequencing and Optimization; 1994;
- [05] **BEASLEY, J. E.;** Operation Research Library: Distributing Test Problem by Electronic Mail; Journal of the Operational Research Society; vol.41, no.11, 1990.
- [06] **BLAZEWICZ, J.; ECKER, K.; SCHIMIDT G.; WEGLARZ, J.;** Scheduling in Computer and Manufacturing Systems; Ed. Springer – Verlag; Berlin; 1996
- [07] **CELADA, F., SEIDEN, P.;** A Computer Model of Cellular Interactions in the Immune System. Immunology Today; Ed. Elsevier, 1992.
- [08] **COHEN, I. R.;** Tending Adam's Garden: Evolving the Cognitive Immune System Self. Ed. Academic Press; San Diego-CA; 2002.
- [09] **DASGUPTA, D.;** An Overview of Artificial Immune Systems and Their Applications, In: News Ideas in Optimization; Ed. McGraw Hill; London; 1999a.
- [10] **DASGUPTA, D.** Artificial Immune Systems and their Applications. Ed. Springer Verlag, 1998.
- [11] **DASGUPTA. D.;** Immunity-Based Intrusion Detection Systems: A General Framework. Proceedings of the 22nd National Information Systems Security Conference (NISSC'99); 1999b.
- [12] **DAVIS, L.;** Job-Shop Scheduling with Genetic Algorithm; Ed. Lawrence Erlbaum; Pittsburgh; PA, (1985).
- [13] **D'HAESELEER, P., FORREST, S., HELMAN, P.;** An Immunological Approach to Change Detection: Algorithms, Analysis and Implications; Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy. IEEE Computer Society Press; 1996.

-
- [14] **De CASTRO, L. N. C.**; Engenharia Imunológica: Desenvolvimento e Aplicações de Ferramentas Computacionais Inspiradas em Sistema Imunológicos Artificiais; Doctorate Thesis; UNICAMP; Campinas – SP; 2001.
 - [15] **De CASTRO, L. N. C.; TIMMIS J.**; Artificial Immune System: A New Computational Intelligence Approach; Ed. Springer, Canterbury 2002.
 - [16] **De CASTRO, L. N., TIMMIS, J.**; Artificial Immune System: A Novel Paradigm to Pattern Recognition; Technical Report; University of Paisley; 2002.
 - [17] **De CASTRO, L. N. C., VON ZUBEN, F.J.**; An Evolutionary Immune Network for Data Clustering; Proceedings IEEE SBRN; 2000a.
 - [18] **De CASTRO, L. N. C., VON ZUBEN, F.J.**; The Clonal Selection Algorithm with Engineering Applications; GECCO'00 – Workshop Proceedings; 2000b.
 - [19] **DORLAND**; Dorland's Illustrated Medical Dictionary, Ed. Dorland; 2003.
 - [20] **FANG, H. L.**; Algorithms in Timetabling and Scheduling; PhD Thesis, Department of Artificial Intelligence. University of Edinburgh, 1994.
 - [21] **FANG, H. L., ROSS, P., CORNE, D.**; A Promising Genetic Algorithm Approach to Job Shop Scheduling, Rescheduling and Open Shop Scheduling Problems; In Proceedings of the 5th International Conference on Genetic Algorithms; Urbana Champaign, IL; 1993.
 - [22] **FISHER, H.; THOMPSON, J. L.**; Probabilistic Learning Combination of local Job Shop Scheduling Rules; Industrial Scheduling; Ed. Prentice Hall; Englewood Cliffs, New Jersey; 1963.
 - [23] **FORREST, S., PERELSON, A., ALLEN, L., CHERUKURI, R.**; Self-Non self Discrimination in a Computer; In Proceedings IEEE Symposium on Research in Security and Privacy; 1994.
 - [24] **GREFENSTETTE, J. J.**; Genesis; A System for Using Genetic Search Procedures; Proceedings of a Conference on Intelligent Systems and Machines; 1984.
 - [25] **GOLDBERG, D. E.**; Genetics Algorithms in Search, Optimization and Machine Learning; Ed. Addison-Wesley, Massachusetts; 1989.
 - [26] **GONÇALVES, J. F., MENDES, J.J. M., RESENDE, M.G.C.**; A Hybrid Genetic Algorithm for Job Shop Scheduling Problem; Technical Report; EJOR - European Journal of Operational Research; 2002.
 - [27] **HART, E.**; Immunology as a Metaphor for Computational Information Processing: Fact or Fiction? PhD Thesis; Department of Artificial Intelligence; University of Edinburgh; 2002.
 - [28] **HART, E.**; The State of the Art in Evolutionary Approaches to Timetabling and Scheduling; Technical Report; EVO Workshop Proceedings; Working Group; 1997.
 - [29] **HART, E., ROSS, P.**; A Heuristic Combination Method for Solving Job Shop Scheduling Problems; Ed. Springer Berlin; Heidelberg; 1998.
 - [30] **HART, E., ROSS, P.**; A Systematic Investigation of GA Performance on Job Shop Scheduling Problems; EVO Workshop Proceedings DBLP; 2000a.
 - [31] **HART, E., ROSS, P.**; An Immune System Approach to Scheduling in Changing Environments; Proceedings GECCO'99; 1999a.
 - [32] **HART, E., ROSS, P.**; Enhancing the Performance of a GA Through Visualization; In Proceedings GECCO'00; 2000b.
 - [33] **HART, E., ROSS, P.**; The Evolution and Analysis of an Potential Antibody Library for Use in Job Shop; In: New Ideas in Optimization, Ed. McGraw Hill; London; 1999b.
 - [34] **HART, E., ROSS, P., NELSON, J.**; Producing Robust Schedules via a Artificial Immune System; In Proceedings ICEC'98; 1998.

-
- [35] **HIGHTOWER, R. R., FORREST, S. A., PERELSON, A. S.;** The Baldwin Effect in the Immune System: Learning by Somatic Hypermutation, Addison Wesley, San Francisco; 1996.
 - [36] **HIGHTOWER, R. R., FORREST, S. A., PERELSON, A. S.;** The Evolution of Emergent Organization in Immune System Libraries; In Proceedings of the 6th International Conference on Genetic Algorithms; 1995.
 - [37] **HOLLAND, J. H.;** Adaptation in Natural and Artificial Systems; Ed. Ann Arbor: Michigan Press; University of Michigan; 1975.
 - [38] **JAIN, A. S., MEERAN, S.;** A State-of-the-Art Review of Job-Shop. Scheduling Techniques; Technical Report, Department of Applied Physics, Electronic and Mechanical Engineering, University of Dundee, Dundee, Scotland, 1998.
 - [39] **KLEIN, R.;** Scheduling of Resource Constrained Projects; Ed. Kluwer Academic Publisher; Norwell – Massachusetts, 2000.
 - [40] **KIM, J.; BENTLEY, P.;** The Artificial Immune System Model for Network Intrusion Detection; In Proceedings of the 7th European Conference on Intelligent Techniques and Soft Computing; EUFIT'99; 1999.
 - [41] **LAWRENCE, S.;** Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques (Supplements); Graduate School Of Industrial Administration; Carnegie-Mellon University; Pittsburgh, Pennsylvania; 1984.
 - [42] **PINEDO, M.;** Scheduling – Theory, Algorithms and Systems; Ed. Prentice Hall; Englewood Cliffs – NJ; 1995.
 - [43] **ROSS, P., HART, E., CORNE, D.;** Genetic Algorithms and Timetabling in Evolutionary Computing; Springer-Verlag; 2000.
 - [44] **STEWART, J. L.;** A Consideration of the Evolution of Microbes and the Immune System; Technical Report, BioOne; 2003.
 - [45] **SOMAYAJI, A., HOFMEYR, S, FORREST, S.;** Principles of a Computer Immune System; In Proceedings of the 2nd New Security Paradigms Workshop, 1997
 - [46] **TAILLARD, E.;** Benchmarks for Basic Scheduling Problem; EJOR – European Journal of Operational Research; 1993.
 - [47] **VARGAS, P. A., De CASTRO, L. N., VON ZUBEN;** Artificial Immune System as Complex Adaptive System; In Proceedings ICARIS'02; 2002.

ANEXOS

Os anexos deste trabalho incluem a interface do sistema implementado, a arquitetura completa do AG, as instâncias utilizadas nos testes e os LOG's dos testes realizados. A divulgação deste conteúdo tem como propósito enriquecer o material e estimular a continuidade do trabalho em pesquisas futuras por parte dos leitores.

I Interface do Sistema Imune Artificial

Sistema Imune Artificial

Arquivo de entrada: E:\Projeto Sildenir\Sildenir\ta02(1).txt

Arquivo de log: E:\Projeto Sildenir\Sildenir\log.txt

Tamanho da população: 100

Probabilidade de cossover: 70

Probabilidade de mutação: 5

Número de iterações: 100

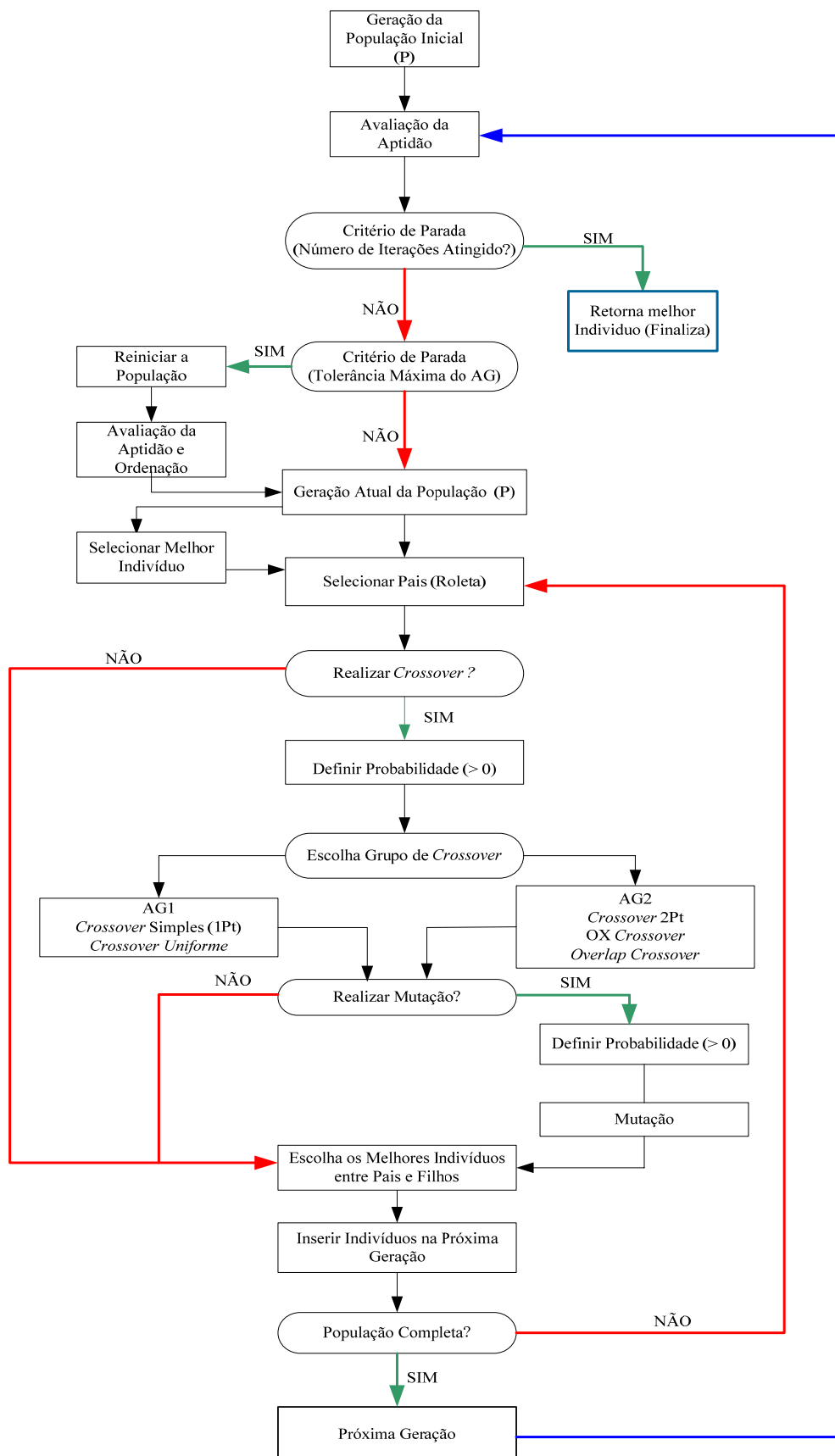
Tolerância máxima: 20

☒ S.I.A.

☐ A.G. 1 ☒ A.G. 2

Executar

II Arquitetura do Algoritmo Genético



III Instâncias Utilizadas

Instância FT06
6 6
(2,1) (0,3) (1, 6) (3, 7) (5, 3) (4,6)
(1,8) (2,5) (4,10) (5,10) (0,10) (3,4)
(2,5) (3,4) (5, 8) (0, 9) (1, 1) (4, 7)
(1,5) (0,5) (2, 5) (3, 3) (4, 8) (5, 9)
(2,9) (1,3) (4, 5) (5, 4) (0, 3) (3, 1)
(1,3) (3,3) (5, 9) (0, 10) (4, 4) (2,1)

Instância FT10
10 10
(0,29) (1,78) (2, 9) (3,36) (4,49) (5,11) (6,62) (7,56) (8,44) (9,21)
(0,43) (2,90) (4,75) (9,11) (3,69) (1,28) (6,46) (5,46) (7,72) (8,30)
(1,91) (0,85) (3,39) (2,74) (8,90) (5,10) (7,12) (6,89) (9,45) (4,33)
(1,81) (2,95) (0,71) (4,99) (6, 9) (8,52) (7,85) (3,98) (9, 22) (5,43)
(2,14) (0, 6) (1,22) (5,61) (3,26) (4,69) (8,21) (7,49) (9,72) (6,53)
(2,84) (1, 2) (5,52) (3,95) (8,48) (9,72) (0,47) (6,65) (4, 6) (7,25)
(1,46) (0,37) (3,61) (2,13) (6,32) (5,21) (9,32) (8,89) (7,30) (4,55)
(2,31) (0,86) (1,46) (5,74) (4,32) (6,88) (8,19) (9,48) (7,36) (3,79)
(0,76) (1,69) (3,76) (5,51) (2,85) (9,11) (6,40) (7,89) (4,26) (8,74)
(1,85) (0,13) (2,61) (6, 7) (8,64) (9,76) (5,47) (3,52) (4,90) (7,45)

Instância ABZ05
10 10
(4,88) (8,68) (6,94) (5,99) (1,67) (2,89) (9,77) (7,99) (0,86) (3,92)
(5,72) (3,50) (6,69) (4,75) (2,94) (8,66) (0,92) (1,82) (7,94) (9,63)
(9,83) (8,61) (0,83) (1,65) (6,64) (5,85) (7,78) (4,85) (2,55) (3, 77)
(7,94) (2,68) (1,61) (4,99) (3,54) (6,75) (5,66) (0,76) (9,63) (8,67)
(3,69) (4,88) (9,82) (8,95) (0,99) (2,67) (6,95) (5,68) (7,67) (1,86)
(1,99) (4,81) (5,64) (6,66) (8,80) (2,80) (7,69) (9,62) (3,79) (0,88)
(7,50) (1,86) (4,97) (3,96) (0,95) (8,97) (2,66) (5,99) (6,52) (9,71)
(4,98) (6,73) (3,82) (2,51) (1,71) (5,94) (7,85) (0,62) (8,95) (9,79)
(0,94) (6,71) (3,81) (7,85) (1,66) (2,90) (4,76) (5,58) (8,93) (9,97)
(3,50) (0,59) (1,82) (8,67) (7,56) (9,96) (6,58) (4,81) (5,59) (2,96)

Instância TA02 - 15X15
<p>15 15</p> <p>(9, 86) (14,60) (4, 10) (13,59) (10, 65) (3, 94) (7, 71) (8, 25) (0, 98) (5, 49) (1, 43) (2, 8) (12,90) (6, 21) (11, 73) (10,68) (8, 28) (11,38) (14,36) (3, 93) (13,35) (9, 37) (7, 28) (4, 62) (2, 86) (6, 65) (1, 11) (5, 20) (12,82) (0, 23) (7, 33) (0, 67) (6, 96) (5, 91) (14, 83) (13,81) (2, 60) (11,88) (4, 20) (12,62) (1, 22) (9, 79) (3, 38) (10,40) (8, 82) (9, 13) (11,14) (14,73) (0, 88) (1, 24) (8, 16) (5, 78) (10,70) (12, 53) (4, 68) (13,73) (3, 90) (6, 58) (7, 7) (2, 4) (11,93) (4, 52) (13,63) (3, 13) (8, 19) (1, 41) (10,71) (12,59) (2, 19) (14,60) (6, 85) (7, 99) (0, 73) (9, 95) (5, 19) (5, 62) (2, 60) (1, 93) (10, 16) (0, 10) (4, 72) (8, 88) (14,69) (6, 58) (3, 41) (9, 46) (7, 63) (11,76) (12,83) (13, 62) (5, 50) (10,68) (13,90) (0, 34) (9, 44) (8, 5) (1, 8) (11,25) (14, 70) (7, 53) (12, 78) (2, 92) (6, 62) (4, 85) (3, 70) (12,60) (0, 64) (9, 92) (3, 44) (13,63) (6, 91) (5, 21) (7, 1) (2, 96) (14,19) (11,59) (8, 12) (10,41) (1, 11) (4, 94) (11,93) (10,46) (5, 51) (13, 37) (1, 91) (9, 90) (8, 63) (7, 40) (3, 68) (6, 13) (0, 16) (2, 83) (14,49) (12, 24) (4, 23) (2, 5) (14,35) (3, 21) (10, 14) (6, 66) (1, 3) (0, 6) (13,98) (11, 63) (4, 64) (5, 76) (8, 94) (7, 17) (12, 62) (9, 37) (11,35) (14,42) (13,62) (5, 68) (4, 73) (9, 27) (1, 52) (6, 39) (12, 41) (0, 25) (2, 9) (8, 34) (10,50) (3, 41) (7, 98) (12,23) (3, 32) (10,35) (8, 10) (4, 29) (7, 68) (13,20) (11, 8) (14, 58) (1, 62) (2, 39) (0, 32) (5, 8) (6, 33) (9, 91) (8, 28) (13,31) (5, 3) (0, 28) (11,66) (9, 59) (4, 24) (12,45) (1, 81) (10, 8) (6, 44) (2, 42) (7, 2) (14, 23) (3, 53) (2, 11) (5, 93) (4, 27) (3, 59) (9, 62) (1, 23) (11,23) (13, 7) (7, 77) (6, 64) (10,60) (14,97) (0, 36) (8, 53) (12,72) (1, 36) (10,98) (4, 38) (2, 24) (0, 84) (7, 47) (6, 72) (9, 1) (11, 91) (12, 85) (5, 68) (14,42) (3, 20) (13, 30) (8, 30)</p>

Instância LA02 - 10X05
<p>10 5</p> <p>(0,20) (3,87) (1,31) (4,76) (2,17) (4,25) (2,32) (0,24) (1,18) (3,81) (1,72) (2,23) (4,28) (0,58) (3,99) (2,86) (1,76) (4,97) (0,45) (3,90) (4,27) (0,42) (3,48) (2,17) (1,46) (1,67) (0,98) (4,48) (3,27) (2,62) (4,28) (1,12) (3,19) (0,80) (2,50) (1,63) (0,94) (2,98) (3,50) (4,80) (4,14) (0,75) (2,50) (1,41) (3,55) (4,72) (2,18) (1,37) (3,79) (0,61)</p>

IV LOG dos Testes Realizados com todos Operadores

Instância FT06 06X06 com AG1

CREATING PERTURBATIONS
CREATING SCENARIOS
USING GA1

Machine-0: 3 antibodies
Machine-1: 1 antibodies
Machine-2: 2 antibodies
Machine-3: 2 antibodies
Machine-4: 1 antibodies
Machine-5: 1 antibodies

SCHEDULING STARTS

makespan = 59
Final makespan = 59
Number of trials = 751
Number of feasible solutions = 10
Search time = 30.0ms
Total time consumed: 6429.0ms

Instância FT06 06X06 com AG2

CREATING PERTURBATIONS
CREATING SCENARIOS
USING GA2

Machine-0: 1 antibodies
Machine-1: 2 antibodies
Machine-2: 1 antibodies
Machine-3: 2 antibodies
Machine-4: 2 antibodies
Machine-5: 1 antibodies

SCHEDULING STARTS

Makespan = 59.0
Final makespan = 59
Number of trials = 751
Number of feasible solutions = 9
Search time = 40.0ms
Total time consumed: 6930.0ms

Instância FT10 10X10 com AG1

CREATING PERTURBATIONS
CREATING SCENARIOS
USING GA1

Machine-0: 4 antibodies
Machine-1: 3 antibodies
Machine-2: 4 antibodies
Machine-3: 2 antibodies
Machine-4: 3 antibodies
Machine-5: 4 antibodies
Machine-6: 2 antibodies
Machine-7: 4 antibodies
Machine-8: 2 antibodies
Machine-9: 3 antibodies

SCHEDULING STARTS

Makespan = 1332.0
Final makespan = 1332
Number of trials = 80001
Number of feasible solutions = 7
Search time = 7852.0ms
Total time consumed: 29079.0ms

Instância FT10 10X10 com AG2

CREATING PERTURBATIONS
CREATING SCENARIOS
USING GA2

Machine-0: 3 antibodies
Machine-1: 4 antibodies
Machine-2: 2 antibodies
Machine-3: 4 antibodies
Machine-4: 3 antibodies
Machine-5: 3 antibodies
Machine-6: 4 antibodies
Machine-7: 3 antibodies
Machine-8: 3 antibodies
Machine-9: 4 antibodies

SCHEDULING STARTS

Makespan = 1332.0
Final makespan = 1558
Number of trials = 80001
Number of feasible solutions = 11
Search time = 7891.0ms
Total time consumed: 30288.0ms

Instância ABZ05 10X10 com AG1

CREATING PERTURBATIONS
CREATING SCENARIOS
USING GA1

Machine-0: 4 antibodies
Machine-1: 3 antibodies
Machine-2: 4 antibodies
Machine-3: 3 antibodies
Machine-4: 3 antibodies
Machine-5: 4 antibodies
Machine-6: 2 antibodies
Machine-7: 4 antibodies
Machine-8: 3 antibodies
Machine-9: 3 antibodies

SCHEDULING STARTS

Makespan = 1558.0
Final makespan = 1558
Number of trials = 80001
Number of feasible solutions = 6
Search time = 7852.0ms
Total time consumed: 25186.0ms

Instância ABZ05 10X10 com AG2

CREATING PERTURBATIONS
CREATING SCENARIOS
USING GA2

Machine-0: 4 antibodies
Machine-1: 3 antibodies
Machine-2: 2 antibodies
Machine-3: 3 antibodies
Machine-4: 3 antibodies
Machine-5: 3 antibodies
Machine-6: 4 antibodies
Machine-7: 3 antibodies
Machine-8: 3 antibodies
Machine-9: 4 antibodies

SCHEDULING STARTS

Makespan = 1558.0
Final makespan = 1558
Number of trials = 80001
Number of feasible solutions = 8
Search time = 7891.0ms
Total time consumed: 25847.0ms

Instância TA02 15X15 com AG1

CREATING PERTURBATIONS
CREATING SCENARIOS
USING GA1

Machine-0: 5 antibodies
Machine-1: 3 antibodies
Machine-2: 3 antibodies
Machine-3: 3 antibodies
Machine-4: 2 antibodies
Machine-5: 4 antibodies
Machine-6: 4 antibodies
Machine-7: 4 antibodies
Machine-8: 3 antibodies
Machine-9: 3 antibodies
Machine-10: 3 antibodies
Machine-11: 3 antibodies
Machine-12: 4 antibodies
Machine-13: 3 antibodies
Machine-14: 3 antibodies

SCHEDULING STARTS

Makespan = 1716.0
Final makespan = 1716
Number of trials = 80001
Number of feasible solutions = 8
Search time = 20109.0ms
Total time consumed: 70402.0ms

Instância TA02 15X15 com AG2

CREATING PERTURBATIONS
CREATING SCENARIOS
USING GA2

Machine-0: 4 antibodies
Machine-1: 4 antibodies
Machine-2: 5 antibodies
Machine-3: 4 antibodies
Machine-4: 3 antibodies
Machine-5: 3 antibodies
Machine-6: 4 antibodies
Machine-7: 3 antibodies
Machine-8: 3 antibodies
Machine-9: 2 antibodies
Machine-10: 3 antibodies
Machine-11: 4 antibodies
Machine-12: 5 antibodies
Machine-13: 2 antibodies
Machine-14: 4 antibodies

SCHEDULING STARTS

Makespan = 1716.0
Final makespan = 1716
Number of trials = 80001
Number of feasible solutions = 4
Search time = 19889.0ms
Total time consumed: 70692.0ms

Instância LA02 10X05 com AG1

CREATING PERTURBATIONS
CREATING SCENARIOS
USING GA1

Machine-0: 2 antibodies
Machine-1: 4 antibodies
Machine-2: 3 antibodies
Machine-3: 5 antibodies
Machine-4: 5 antibodies

SCHEDULING STARTS

Makespan = 924.0
Final makespan = 924
Number of trials = 80001
Number of feasible solutions = 82
Search time = 4196.0ms
Total time consumed: 16672.0ms

Instância LA02 10X05 com AG2

CREATING PERTURBATIONS
CREATING SCENARIOS
USING GA2

Machine-0: 4 antibodies
Machine-1: 3 antibodies
Machine-2: 2 antibodies
Machine-3: 4 antibodies
Machine-4: 3 antibodies

SCHEDULING STARTS

Makespan = 886.0
Final makespan = 886
Number of trials = 80001
Number of feasible solutions = 81
Search time = 4296.0ms
Total time consumed: 18809.0ms

V LOG dos Testes Realizados Somente com Um Operador

Instância FT06 06X06 com AG1

	GA1	AIS	t_GA(ms)	t_AIS(ms)
1:	59	60	1241	24669
2:	55	56	1268	24663
3:	59	55	1246	24541
4:	59	56	1206	24649
5:	59	55	1218	24496
6:	55	55	1238	24450
7:	59	59	1217	24803
8:	58	55	1205	24552
9:	58	55	1218	24571
10:	58	56	1232	24541
RESULTS:				
	min	max	mean	s.d.
GA1:	55	59	57.90	1.60
AIS:	55	60	56.20	1.81
TIMES (ms)				
	min	max	mean	s.d.
GA1:	1205	1268	1229	20
AIS:	24450	24803	24594	103

Instância FT06 06X06 com AG2

	GA2	AIS	t_GA(ms)	t_AIS(ms)
1:	55	55	2528	38169
2:	55	55	2537	38042
3:	55	55	2543	38206
4:	55	55	2531	37993
5:	57	55	2562	38161
6:	57	55	2544	38069
7:	57	55	2527	38184
8:	57	55	2574	38831
9:	57	55	2579	39405
10:	59	55	2673	39282
RESULTS:				
	min	max	mean	s.d.
GA2:	55	59	56.40	1.35
AIS:	55	55	55.00	0.00
TIMES (ms)				
	min	max	mean	s.d.
GA2:	2527	2673	2560	44
AIS:	37993	39405	38434	533

Instância FT10 10X10 com AG1

	GA1	AIS	t_GA(ms)	t_AIS(ms)
1:	1094	1287	3087	68028
2:	1119	1361	3284	67537
3:	1101	1471	3234	68800
4:	1006	1339	3157	68473
5:	1050	1360	3189	69094
6:	1088	1256	3208	70087
7:	1149	1223	3206	68724
8:	1137	1379	3224	69164
9:	1114	1163	3247	67176
10:	1088	1384	3129	69829

RESULTS:

	min	max	mean	s.d.
GA1:	1006	1149	1094.60	41.76
AIS:	1163	1471	1322.30	90.19

TIMES (ms)

	min	max	mean	s.d.
GA1:	3087	3284	3196	58
AIS:	67176	70087	68691	928

Instância FT10 10X10 com AG2

	GA2	AIS	t_GA(ms)	t_AIS(ms)
1:	1031	1234	6841	106801
2:	1072	1280	6907	104157
3:	1110	1300	6728	105531
4:	1088	1316	6753	105299
5:	1187	1255	6814	105044
6:	1163	1271	6865	105260
7:	1222	1258	7158	103893
8:	1071	1294	6785	105847
9:	1137	1302	6835	105744
10:	1172	1175	6807	104827

RESULTS:

	min	max	mean	s.d.
GA2:	1031	1222	1125.30	60.74
AIS:	1175	1316	1268.50	41.34

TIMES (ms)

	min	max	mean	s.d.
GA2:	6728	7158	6849	120
AIS:	103893	106801	105240	840

Instância ABZ05 10X10 com AG1

	GA1	AIS	t_GA(ms)	t_AIS(ms)
1:	1330	1380	2991	67751
2:	1340	1406	3033	67508
3:	1295	1313	3073	67715
4:	1432	1390	3062	67326
5:	1350	1396	3077	67711
6:	1397	1333	3078	67713
7:	1339	1329	3046	66079
8:	1446	1403	3112	69970
9:	1302	1380	3078	67558
10:	1376	1338	3043	67172
RESULTS:				
	min	max	mean	s.d.
GA1:	1295	1446	1360.70	51.24
AIS:	1313	1406	1366.80	34.77
TIMES (ms)				
	min	max	mean	s.d.
GA1:	2991	3112	3059	33
AIS:	66079	69970	67650	957

Instância ABZ05 10X10 com AG2

	GA2	AIS	t_GA(ms)	t_AIS(ms)
1:	1468	1393	6457	101573
2:	1353	1405	6444	100707
3:	1593	1397	6588	101530
4:	1347	1426	6490	102010
5:	1514	1395	6522	101346
6:	1416	1405	6611	101585
7:	1332	1354	6484	101528
8:	1356	1338	6559	101987
9:	1357	1348	6568	101370
10:	1422	1410	6503	101696
RESULTS:				
	min	max	mean	s.d.
GA2:	1332	1593	1415.80	85.98
AIS:	1338	1426	1387.10	29.64
TIMES (ms)				
	min	max	mean	s.d.
GA2:	6444	6611	6523	57
AIS:	100707	102010	101533	366

Instância TA02 15X15 com AG1

	GA1	AIS	t_GA(ms)	t_AIS(ms)
1:	1531	1596	6358	146569
2:	1570	1625	6438	144819
3:	1556	1579	5761	148124
4:	1599	1652	5709	139697
5:	1568	1631	5752	147958
6:	1618	1623	6460	144248
7:	1606	1631	5690	147542
8:	1607	1584	5827	139573
9:	1613	1579	5665	148339
10:	1525	1619	6468	148533

RESULTS:

	min	max	mean	s.d.
GA1:	1525	1618	1579.30	34.22
AIS:	1579	1652	1611.90	25.56

TIMES (ms)

	min	max	mean	s.d.
GA1:	5665	6468	6013	364
AIS:	139573	148533	145540	3437

Instância TA02 15X15 com AG2

	GA2	AIS	t_GA(ms)	t_AIS(ms)
1:	1563	1579	14198	242485
2:	1592	1625	14145	218609
3:	1604	1579	14761	216374
4:	1539	1593	13353	215699
5:	1730	1579	13083	213603
6:	1598	1591	12913	209558
7:	1660	1581	13578	217086
8:	1544	1623	12826	215214
9:	1551	1579	14490	223342
10:	1599	1594	17116	211272

RESULTS:

	min	max	mean	s.d.
GA2:	1539	1730	1598.00	58.85
AIS:	1579	1625	1592.30	17.80

TIMES (ms)

	min	max	mean	s.d.
GA2:	12826	17116	14046	1272
AIS:	209558	242485	218324	9308

Instância LA02 15X05 com AG1

	GA1	AIS	t_GA(ms)	t_AIS(ms)
1:	708	826	1673	34770
2:	734	773	1758	35831
3:	720	826	1820	36403
4:	727	826	1712	35392
5:	739	811	1767	35564
6:	695	762	1722	34986
7:	704	841	1750	34984
8:	697	838	1690	34681
9:	709	826	1705	34763
10:	722	826	1732	34836

RESULTS:

	min	max	mean	s.d.
GA1:	695	739	715.50	15.21
AIS:	762	841	815.50	26.66

TIMES (ms)

	min	max	mean	s.d.
GA1:	1673	1820	1733	43
AIS:	34681	36403	35221	566

Instância LA02 15X05 com AG1

	GA2	AIS	t_GA(ms)	t_AIS(ms)
1:	739	780	3451	53819
2:	739	826	3565	54477
3:	742	770	3444	54720
4:	721	795	3589	54030
5:	736	780	3497	54078
6:	714	826	3455	52374
7:	721	826	3449	52148
8:	753	826	3583	52647
9:	754	826	3590	54020
10:	676	792	3717	54733

RESULTS:

	min	max	mean	s.d.
GA2:	676	754	729.50	23.04
AIS:	770	826	804.70	23.45

TIMES (ms)

	min	max	mean	s.d.
GA2:	3444	3717	3534	90
AIS:	52148	54733	53705	964

APÊNDICE

Glossário

Apresenta-se neste glossário uma descrição rápida de termos poucos comuns utilizados nesta dissertação, destacando principalmente terminologias biológicas, computacionais, matemáticas e algorítmicas.

A

Algoritmo: Seqüência de passos bem definidos, estruturados e organizados hierarquicamente, que permite resolver um problema ou gerar um resultado quando executado.

Algoritmos Evolucionários: São extensões dos Algoritmos Genéticos. Utilizam modelos computacionais de processamento evolucionário no desenvolvimento de sistemas de computação para resolver problemas complexos. Baseia-se conceitualmente em simulações da evolução através de três operadores básicos: seleção, cruzamento e mutação.

Algoritmos Genéticos: Algoritmos evolutivos. Possuem características evolutivas baseada na teoria da evolução Darwinista. Foi inicialmente proposto John Holland em 1975.

Anomalias: Defeitos, má formação. De anormalidade ou fora do normal.

Anticorpo: Proteínas presentes no plasma que atacam proteínas estranhas ao corpo, chamadas de antígenos, realizando assim a defesa do organismo. São representadas e secretadas pelas células B. Também são chamados de imunoglobulinas.

Antígeno: Um antígeno é toda partícula ou molécula, que quando introduzida no organismo, é capaz de iniciar a produção de um anticorpo específico. São representados no sistema pelas células T.

B

Baço: Órgão do corpo humano formado por duas regiões conhecidas como polpa branca e polpa vermelha onde encontram-se as células T, e as células B.

Bactérias: São microorganismos unicelulares, procariontes, aclorofilados, microscópicos, que se reproduzem por divisão binária.

Biotecnologia: Engenharia biológica. Processo tecnológico que utiliza material biológico (plantas e animais), onde são aplicados conhecimentos matemáticos, biológicos, químicos e físicos para produzir bens e serviços.

C

Células: Unidades estruturais e funcionais dos organismos vivos. A célula representa a menor porção de matéria viva dotada da capacidade de autoduplicação independente.

Célula B: Tipo de linfócito que produz anticorpos. Compõe o sistema imunológico e são essenciais para a defesa do organismo. Este tipo de célula é capaz de responder rapidamente e de forma eficiente a agentes que já tenha ocorridos no sistema.

Célula NK: Células assassinas, matadoras. Pertencem a uma classe de linfócitos não T e não B.

Célula T: Tipo de linfócito responsável pela imunidade mediada por células (imunidade celular); inclui as células T citotóxicas e as células T helper.

Citosinas: Moléculas peptídicas que participam na comunicação entre as células. Grupo de fatores extracelulares que podem ser produzidos por diversas células, como monócitos, macrófagos e linfócitos, dotadas de funções importantes no controle da resposta inflamatória local e sistêmica.

CLARISA: *Classification via Artificial Immune System Analogy*. Sistema imune artificial para classificação de padrões de *scheduling*, análogo aos sistemas imunes biológicos.

Classe NP: Compreende todos os problemas da Classe NP (Não Polinomial) sem solução em tempo polinomial viável.

CLONALG: *Clonal Selection Algorithm*. Algoritmo que representa o modelo da teoria da seleção clonal. Visa à representação da resposta adaptativa.

Clone: Indivíduo derivado de outro e que possui as mesmas características genéticas do indivíduo original.

Cromossomo: Conjunto ou cadeias de genes que determina as características dos indivíduos.

Crossover: Operadores genéticos de cruzamento ou recombinação dos indivíduos.

D

Data Mining: Conjunto de técnicas que buscam a aquisição de novos conhecimentos através da análise de grandes bases de dados.

E

Efeito Dominó: Expressão utilizada metaforicamente que indica uma reação em cadeia.

Esplenalgia: Dor localizada na região do baço correspondente ao processo de esplenocêntrica que tem por objetivo repor os glóbulos vermelhos circulantes.

Epitopo: Antígeno determinante. Parte da toxina reconhecida pela célula T. Liga ao paratopo do antígeno.

F

Fenótipo: Características de um indivíduo determinada pelo genótipo. O fenótipo apresenta constituição e a aparência física de um indivíduo

G

Gânglios: Agrupamento de células nervosas no sistema nervoso periférico.

Gânglios Linfáticos: Estruturas nodulares que formam parte do sistema linfático, formando agrupamentos em forma de conjuntos localizados nas axilas, colo, abdômen e outros órgãos.

Gene: Unidade fundamental da hereditariedade. Cada gene é formado por uma sequência específica de ácidos nucleicos. O conjunto dos genes de um organismo, população ou espécie constitui o genoma.

Genótipo: Conjunto dos genes de um indivíduo ou conjunto dos fatores hereditários que entram na constituição de um indivíduo ou de uma linhagem.

H

Heurísticas: Método de solução de problemas indutivo baseado em regras derivadas do senso comum ou da experiência de um modelo teórico da matemática.

Humoral: Seroso. Tipo de imunidade específica, chamado de imunidade humoral. Produzidos por secreções, como: nasal e ocular. Não possui capacidade de proteção.

I

Idiotopo: Antígeno único na região – V do anticorpo.

Imunoglobina: Família ou conjunto de molécula com grandes propriedades protéicas, também conhecidas como anticorpos.

Interleucina: Citocinas elaboradas por leucócitos. São envolvidas na sinalização entre as células do sistema imunológico.

Infecções: Processo inflamatório no qual exista um agente infeccioso.

Infecções Metabólicas: Infecções não virais do organismo. Infecções que são causadas por agentes patogênicos.

J

Job Shop: Escalonar recursos ou processos em postos de trabalho para executar uma coleção de tarefas (*jobs*) de acordo com o objetivo de produção.

L

Leucócitos: São células sangüíneas que atuam no reconhecimento e erradicação de agentes infecciosos ou quaisquer células estranhas ao organismo.

Linfócito: Célula branca (glóbulo) do sangue responsável pelas respostas imunológicas. Há dois tipos principais de linfócitos: células B, responsáveis pela produção de anticorpos, e células T, que regem todos os aspectos da resposta imune. Realizam funções especializadas, como destruir células infectadas por microrganismos. As células B e T são produzidas na medula óssea e no timo, respectivamente.

LOG: Registro de testes computacionais realizados para algum aplicativo ou aplicação. Em alguns casos este registro é uma cópia (*backup*) dos dados processados e armazenados em outro local.

M

Makespan: Resultado do processamento ou medida de desempenho obtida pela a duração total de um escalonamento.

Matching: Região apresentadora de resultados ou parâmetros. Relação de equivalência obtida após resultados comparativos.

Medula: Designação genérica que se caracteriza por sua situação central com referência ao órgão ou estrutura em que se encontra.

Medula Óssea: Tecido gelatinoso que preenche a cavidade interna de vários ossos e fabrica os elementos figurados do sangue periférico como: hemácias, leucócitos e plaquetas.

Memória Imunológica: Reação imune intensa dado pela capacidade dos leucócitos de reconhecer o mesmo estímulo antigênico caso ele entre novamente em contato com o organismo.

Micro Organismos: Organismos que se encontram na guaia, nos alimentos, dentro de outros organismos ou no meio ambiente e que podem causar doenças ao homem, animais ou plantas.

Molécula: Menor porção de uma substância que ainda conserva as características e propriedades da mesma.

Mutação: Mudança, transformação. Modificação casual ou induzida da informação genética de um indivíduo.

Mutação Somática: Tipo de mutação que ocorre em células somáticas e, conseqüentemente, afeta somente os seus descendentes. Possui características não herdável.

N

NP Completo: Subconjunto de problemas da classe NP, que possui a seguinte surpreendente propriedade: se qualquer um deles puder ser resolvido em tempo polinomial determinístico, então todos os outros problemas em NP também podem.

NP Hard: Subconjunto da classe NP (Não Polinomial) de difícil solução. Acredita-se não existir um algoritmo de complexidade com tempo polinomial viável para resolver um problema dessa classe.

O

Organismo: Na biologia e na ecologia, um organismo é todo e qualquer ser vivo.

Otimização: Aplicação de métodos e modelos a determinado problema, analisando e avaliando todas as restrições possíveis com o intento de alcançar a melhor solução.

P

Paratopo: Ponto ou marcador existente na molécula de anticorpo e que liga ao antígeno correspondente. Representação utilizada para representar a ligação de um antígeno e um anticorpo.

Patogênicos: Agentes infecciosos. Corpo estranho, não próprio do organismo humano.

Patógenos: Microorganismos indicadores de contaminação ambiental (vírus e bactérias).

PRAIS: Pattern Recognizing Artificial Imune System. Sistema Imune Artificial para reconhecimento de padrões.

Peptídeo: Antigênico reconhecido por uma região-V de um anticorpo. Ligados através das moléculas MHC de forma alélica específica.

Poli-peptídico: De peptídeo: antigênico com vários alelos de ligação.

Processamento Paralelo: Mesmo que computação paralela. Processamentos que simultaneamente são dados por um conjunto de computadores que fazem tarefas em conjunto ou paralelo.

S

Seleção Clonal: Preside o reconhecimento das células T pelos antígenos e permite a maturação apenas das famílias de células que captam HMC através da seleção.

Scheduling: Agendamento, programação, escalonamento. Atividade de sequenciamento ou processamento de tarefas. Trata-se geralmente de um problema da classe Job shop.

Sistema Imune: Conjunto de células e órgãos que defendem o organismo contra germes ou agentes patogênicos.

Sistema Nervoso: Conjunto de órgãos que transmitem os impulsos necessários aos movimentos e às diversas funções de todo o organismo. Recebem do próprio organismo e do mundo externo as sensações e estímulos.

T

Timo: Órgão linfático mais desenvolvido no período pré-natal, involui desde o nascimento até a puberdade.

V

Vacinas: Substâncias que, ao serem introduzidas no organismo, suscitam uma reação do sistema imunológico semelhante ao que ocorreria no caso de uma infecção por um determinado agente patogênico, tornando o organismo imune a esse agente ou doenças por ele provocadas.

Vírus: Agentes patogênicos ou micróbios causadores de doenças infecciosas. Sua reprodução se dá apenas em células vivas.