UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE TELEMÁTICA

UMA ABORDAGEM IMUNO-INSPIRADA PARA A RESOLUÇÃO DO PROBLEMA DE PLANEJAMENTO DE PROCESSOS

Leonardo José Ramos Freitas Filho

Orientador : Prof. Dr. Akebo Yamakami
DT/FEEC/Unicamp

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica.

Campinas – São Paulo – Brasil
Fevereiro de 2008

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Freitas Filho, Leonardo José Ramos
F884a Uma abordagem imuno-inspirada para a resolução do problema de planejamento de processos / Leonardo José Ramos Freitas Filho. --Campinas, SP: [s.n.], 2008.

Orientador: Akebo Yamakami
Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Pesquisa operacional. 2. Agenda de execução (Administração). 3. Job Shops. 4. Escalonamento de produção. 5. Sistema imune - Simulação (Computadores). I. Yamakami, Akebo. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: A immune approach for job shop scheduling
Palavras-chave em Inglês: Operational research, Tasks scheduling, Job Shops, Production Planing, Immunological system
Área de concentração: Telecomunicações e Telemática
Titulação: Mestre em Engenharia Elétrica
Banca examinadora: Takaaki Ohishi, Tatiane Regina Bonfim
Data da defesa: 21/02/2008
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Leonardo José Ramos Freitas Filho

Data da Defesa: 21 de fevereiro de 2008

Título da Tese: "Uma Abordagem Imuno Inspirada para a Resolução do Problema de Planejamento de Processos"

Prof. Dr. Akebo Yamakami (Presidente): _____

Profa. Dra. Berenice Camargo Damasceno: _____

Profa. Dra. Tatiane Regina Bonfim: _____

Prof. Dr. Takaaki Ohishi: _____

Resumo

Um dos problemas mais comuns em planejamento de processos é o escalonamento. Este se resume na alocação de tarefas a serem executadas utilizando determinados recursos, sujeitas a uma ordem de prioridades, restrição de tempo e disponibilidade. O Job Shop é um dos sistemas mais conhecidos e estudados, pois reúne características de outras classes de sistemas de planejamento, além de ser necessária a elaboração de métodos de resolução mais sofisticados devido sua alta complexidade (NP-difícil).

A computação natural é constituída por novas abordagens de computação caracterizadas por uma maior proximidade com a natureza. A aplicação de métodos inspirados em sistemas imunológicos demonstra, nos últimos anos, ser promissor em diversas áreas de aplicação e pesquisa, como por exemplo, otimização.

Tendo em vista este cenário, é feita aqui uma abordagem imuno-inspirada para elaboração de um sistema de otimização, através de resolução do escalonamento de um Job Shop, provendo soluções diversificas e de alta qualidade.

Palavras-chave: Pesquisa operacional, Agenda de execução (Administração), Job Shops, Escalonamento de produção, Sistema imune - Simulação (Computadores).

Abstract

Scheduling problems are jobs allocation using distinct sources ordered by sequence constraint and time constraint. *Job Shop* is one the most studied problem because it is possible to find different scheduling features. Also this problem needs sophisticated solving methods by complexity characteristics (NP-Hard).

Natural Computing uses a new computation approach representing natural systems. Immune system methods approaches shows, in the last years, good results in many applications and several researches, for example, optimization.

In this scenario we developed an artificial immune system to solve *Job Shop* system. We will show hight diversity and quality solutions

Keywords: Operational research, Tasks scheduling, Job Shops, Production Planing, Immunological system.

Agradecimentos

Ao meu orientador, Profs. Akebo Yamakami, sou grato pela oportunidade e pela orientação cujos conselhos e sugestões foram muito valiosos neste trabalho.

Aos meus pais, Leonardo e Silvia, e irmãos João Paulo, Rafael e Gabriel por me ensinarem sobre a vida.

A minha companheira Laura, seus pais e irmão, pela dedicação, paciência e carinho.

Aos meus grandes amigos André e Igor que aturavam meus "devaneios heurísticos".

A H2B

Ao Rodrigo pelo apoio e sugestões na etapa final deste trabalho e até mesmo pela referência de Redes de Petri que ficou de mandar...

Ao Carlos que até hoje eu espero a função geradora de números aleatórios!

Aos meus grandes companheiros de laboratório, Fernando e José Leite, pela ajuda "bits & bytes" nos momentos de desespero.

Ao Tiago, pelas correções deste trabalho.

Aos demais colegas de pós-graduação, pelas críticas e sugestões.

A Unicamp, pela infra-estrutura.

A CAPES, pelo apoio financeiro.

Finalmente, agradeço a Deus por todas essas oportunidades.

Sumário

Resumo iii
Abstract iv
Agradecimentos v
Lista de figuras viii
Lista de Diagramas ix
Lista de Tabelas x
Glossário xi
Capítulo 1 1
Estado da Arte – Planejamento de Processos 1
1.1 Introdução 1
1.2 Formulação Matemática do *Job Shop* 4
1.3 Estado da arte 5
1.3.1 Métodos Exatos 5
1.3.2 Heurísticas 6
1.3.3 Meta-Heurísticas 6
Capítulo 2 13
Sistema Imunológico Biológico 13
2.1 Introdução 13
2.2 Sistema imune adaptativo 16
2.2.1 Diversidade imunológica e reconhecimento de padrões 17
2.2.2 Memória e aprendizagem 18
2.2.3 Princípio da seleção clonal 18
2.2.4 Mutação e afinidade 20
2.2.5 Mecanismos de seleção 21
Capítulo 3 22
Sistema Imunológico Artificial e Planejamentos 22
3.1 Introdução 22
3.2 Exemplos de Trabalhos 23
Capítulo 4 34
Representação da solução do *Job Shop* 34
4.1 Introdução 34
4.2 Dois exemplos de representação 34
4.3 Grafo Disjuntivo 39
4.3.1 Movimentação 41
4.3.2 Vizinhança 42
Capítulo 5 46
Abordagem Imuno-Inspirada 46
5.1 Introdução 46
5.2 Representação da solução 47
5.3 Inicialização do repertório 47
5.4 Medida de afinidade 48

5.5 Clonagem ou Seleção Clonal 48
5.6 Maturação de Afinidade (mutação) 50
5.6.1 Definição do movimento 50
5.6.2 Definição da Vizinhança 51
5.7 Mecanismo de Seleção 52
5.8 Memória 52
5.9 Diversidade 52
5.9.1 Supressão 53
5.9.2 Tratamento a ótimos-locais 55
5.10 Pseudocódigo do Sistema Imunológico Artificial 56
Capítulo 6 58
Experimentos e Resultados 58
6.1 Introdução 58
6.2 Avaliação de grupos de soluções 59
6.3 Plataforma 60
6.4 Instâncias testes 60
6.4.1 Fisher e Thompson (1963) 62
6.4.2 Adams et al. (1988) 63
6.4.3 Lawrence (1984) 64
6.4.4 Applegate e Cook (1991) 65
6.5 Análise dos resultados 67
Capítulo 7 75
Conclusão 75
7.1 Considerações Finais 75
7.2 Perspectivas futuras 76
Referências 77

Lista de figuras

Figura 1.1 - Exemplo de linha de montagem 2
Figura 1.2 - Gráfico de Gantt da solução..... 3
Figura 2.1 - Sistema imunológico biológico estruturado 15
Figura 2.2 - Ilustração de uma célula B..... 16
Figura 2.3 - Reconhecimento de antígenos pelos anticorpos 16
Figura 2.4 - Etapas simplificadas do processo imunológico 19
Figura 3.1 - Exemplo de cenário de aplicação..... 24
Figura 3.2 - *Flow Shop*: exemplos de soluções representadas por grafo de Gantt 28
Figura 3.3 - Codificação da solução 30
Figura 3.4 - Moléculas codificadas por inteiros 32
Figura 4.1 - Passos da construção da solução a partir do vetor de fila de prioridade..... 36
Figura 4.2 - Exemplos de soluções por Filas de Prioridade 37
Figura 4.3 - Exemplo de codificação por Permutação 38
Figura 4.4 - Exemplo de recombinação de soluções 38
Figura 4.5 - Grafo com os arcos conjuntivos 39
Figura 4.6 - Grafo com os arcos conjuntivos e disjuntivos. 40
Figura 4.7 - Grafo direcionado e gráfico de Gantt 41
Figura 4.8 - Grafo direcionado e Gantt de uma solução vizinha da Figura 4.7..... 43
Figura 4.9 - Vizinhança de Nowicki e Smutnicki (1996)..... 44
Figura 4.10 - Exemplo onde a vizinhança de Nowicki e Smutnicki limitou a busca..... 45
Figura 4.11 - Grafo Disjuntivo e Gráfico de Gantt..... 45
Figura 5.1 - Variação do número de clones por anticorpos..... 49
Figura 5.2 - Tipo de mutação sofrida pelo clone..... 50
Figura 5.3 - Estrutura da métrica..... 53
Figura 5.4 - Ilustração do mecanismo de seleção..... 54
Figura 5.5 - Ilustração do mecanismo de Tratamento a ótimos-locais. 56
Figura 6.1 - Gráfico com melhor e pior solução de *K* por instância..... 68
Figura 6.2 - Gráfico com a média do conjunto-solução por iteração 69
Figura 6.3 - Média das distâncias relativas do conjunto-solução, exemplo *la06*. 71
Figura 6.4 - Média das distâncias do conjunto-solução de *la06* (outra rodada)..... 71
Figura 6.5 - Gráfico com a avaliação do conjunto solução por iteração 72
Figura 6.6 - Gráfico de diversidade das soluções. 73
Figura 6.7 - Avaliação do conjunto solução por iteração..... 73

Lista de Diagramas

Diagrama 3.1 - Processo do SIA por Alisantoso et al (2003) 29

Lista de Tabelas

Tabela 1.1 - Processamento das tarefas 3
Tabela 3.1 - Ilustração da relação entre o SIA e o Planejamento 31
Tabela 4.1 - Instância teste com os processamentos das tarefas 35
Tabela 6.1 - Tabela de comparação dos parâmetros 61
Tabela 6.2 - Tabela de apresentação das k-soluções para instâncias do tipo *ft.* 62
Tabela 6.3 - Matrizes de distâncias relativas para instâncias do tipo *ft* 62
Tabela 6.4 - Tabela de apresentação das k-soluções para instâncias do tipo *abz.* 63
Tabela 6.5 - Matrizes de distâncias relativas para instâncias do tipo *abz* 63
Tabela 6.6 - Tabela de apresentação das k-soluções para instâncias do tipo *la*. 64
Tabela 6.7 - Matrizes de distâncias relativas das instâncias do tipo *la* 64
Tabela 6.8 - Matrizes de distâncias relativas para instâncias do tipo *la* 65
Tabela 6.9 - Tabela de apresentação das k soluções para instâncias do tipo *orb*. 66
Tabela 6.10 - Matrizes de distâncias relativas para instâncias do tipo *orb* 66
Tabela 6.11 - Tabela sintetizada das k soluções por instância. 67
Tabela 6.12 - Tabela com os desvios em relação a melhor solução da literatura. 68
Tabela 6.13 - Tabela de apresentação dos avaliadores de conjunto-solução *K=10*. 70

Glossário

Conceitos retirados de De Castro (2001)

Aminoterminal: *região do anticorpo que sofre alterações (região-V).*

Anticorpo: *molécula solúvel de proteína plasmática produzida e secretada pelas células Bem resposta a um antígeno, ligando-se especificamente a estes agentes infectantes, ou antígenos, os anticorpos podem neutralizá-los ou prepará-los para o processamento (ingestão e digestão) pelos fagócitos.*

Carboxiterminal: *região do anticorpo que não sofre alterações.*

Efetoras: *biológica.*

Epítopos: *um único determinante antigênico, funcionalmente é a porção do antígeno que liga-se ao paratopo do antígeno.*

Fagócitos: *células que fagocitam, contribuem para a defesa imune ingerindo e digerindo micróbios ou outras células e partículas invasoras.*

Fenotípica: *características físicas e mentais de um organismo conseqüentes do genótipo.*

Histocompatibilidade: *constitui um conjunto de genes que codificam as moléculas e proteínas envolvidas no processamento de antígenos e na defesa do hospedeiro.*

Hipermutação: *durante as respostas das células B aos antígenos, os genes daregião-V sofrem mutação somática com altas taxas, visando a geração de vários anticorpos, alguns dos quais se ligam com maior afinidade as estímulo antigênico.*

Idiotopos: *um único antígeno na região-V do anticorpo.*

Imunoglobulina: *famílias de moléculas de grandes proteínas, também conhecidas por anticorpos.*

Linfócitos: *pequenas células brancas produzidas pelos órgãos linfóides e responsáveis pela resposta imune.*

Macrófagos: *grandes células fagocitárias que atuam devorando micróbios, apresentando antígenos a outras células e expelindo secreções imunes.*

Microorganismos: *organismos microscópicos, unicelulares com exceção de alguns fungos, que incluem bactérias, leveduras e protozoários, podendo causar doenças humanas.*

Neutrófilos: *também conhecidos como leucócitos polimorfonucleares, constituem aprincipal categoria de células brancas no sangue periférico humano.*

Nucleotídio: *material genético.*

Patógenos: *microorganismos infecciosos externos.*

Peptídeos: *marcador antigênico reconhecido pela região-V de um anticorpo.*

Plasmócitos: *células produtoras de anticorpos diferenciadas a partir de células B.*

Capítulo 1

Estado da Arte – Planejamento de Processos

Resumo – Por reunir a característica de difícil resolução e grande aplicabilidade, o *Job Shop* é um sistema muito estudado na literatura; isso explica o constante desenvolvimento de novas técnicas que propõem abordagens mais eficazes e escopo de soluções mais robustas. Este capítulo introduz e conceitualiza os problemas de planejamento, apresenta o *Job Shop* e suas peculiaridades; em seguida, é levantado um breve histórico dos estudos e propostas de soluções desenvolvidas na literatura que tiveram relevância como Métodos Exatos, Heurísticas e Meta-heurísticas.

1.1 Introdução

Um dos problemas mais comuns em planejamento de processos é o escalonamento a curto prazo que planifica uma tarefa *(job)* em um sistema de produção. O escalonamento consiste na alocação de recursos, levando em consideração a execução de um conjunto de operações, com o intuito de atingirem objetivos pré-determinados, sujeito a um conjunto de restrições (Bonfim, 2006). Estes problemas possuem extensa aplicação em indústrias, linha de produção, logística, agendamento de pessoas e outros. São considerados de difícil resolução, *NP-completos,* não existindo algoritmo de solução geral (Lenstra,1977). Devido a tal característica, são geralmente tratados com heurísticas e meta-heurísticas. Na literatura, são relacionados aos problemas com um conjunto ***J*** de tarefas *(jobs)* que precisam ser executadas em um número ***M*** de máquinas; de maneira estendida, um conjunto de operações que precisam ser realizadas utilizando um conjunto de recursos. Tais problemas podem ser segmentados da seguinte maneira:

- *Open Shop*: nos problemas de *Open Shop* não há restrições na ordem das operações, ou seja, qualquer ordem das tarefas processando em qualquer ordem de máquinas, neste caso, o sequenciamento é factível.

- *Flow Shop*: muito comum em linha de montagem de processo de produção, este problema tem a característica de manter fixa a seqüência das máquinas, ocorrendo variação apenas na ordem das tarefas. Essa característica diminui o espaço de busca e qualquer sequenciamento das tarefas é factível.

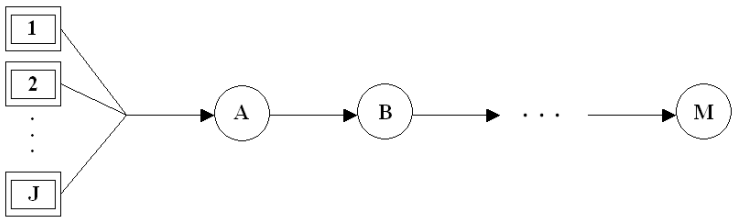


Figura 1.1 - Exemplo de linha de montagem. Cada tarefa de 1 a J deve ser processada em A , B , ... , M; obedecendo esta seqüência fixa de máquinas.

- *Jop Shop*: reúne todas as dificuldades de um problema de planejamento e não apresenta as simplificações características do *Flow Shop* e do *Open Shop*. O *Job Shop* é o mais estudado na literatura. Neste, nem todas as **J** tarefas *(Job)* passam pelas **M** máquinas, algumas tarefas possuem múltiplas operações numa mesma máquina, e cada uma delas pode ter uma seqüência específica de operação. Por fim, existem sequenciamentos infactíveis neste espaço deixando o problema com uma abordagem mais difícil do que os anteriores.

Todos os tipos de problemas podem ter caráter multi-objetivo: respeitar prazos, minimização de estoque em processo, minimização de tempo de execução, reduzir tempo de configuração *(setup)*, minimizar utilização de pessoal e equipamento, atingindo assim uma redução no custo total de produção.

Neste trabalho será estudado o problema de *Job Shop*, utilizando as seguintes suposições para torná-lo mais tratável:

- Os tempos de processamento das tarefas são quantidades determinísticas *(conhecidos, sem variações)*;
- As máquinas só podem processar uma tarefa de cada vez e sem interrupções;
- O tempo de transporte e de configuração de máquina são ignorados ou são incluídos nos tempos de processamento e não dependem da ordem;
- No inicio, todas as tarefas estão prontas para o processamento e todas as máquinas estão prontas para executá-las;
- Todas as tarefas têm igual importância e são independentes;
- Há somente uma máquina de cada tipo;
- As tarefas aceitas para o processamento nas máquinas não podem ser canceladas.

Segue um exemplo ilustrativo de escalonamento de um *Job Shop*:

Na tabela abaixo se encontra o sequenciamento de cada tarefa e o tempo de processamento em cada máquina.

Tarefa	Ordem de operações por máquina (Tempo de processamento)		
1	1 (3)	2 (3)	3 (3)
2	1 (2)	3 (3)	2 (4)
3	2 (3)	1 (2)	3 (1)

Tabela 1.1 - Processamento das tarefas; extraído de Yamada e Nakano (1997)

Uma solução pode ser representada pelo gráfico de Gantt:

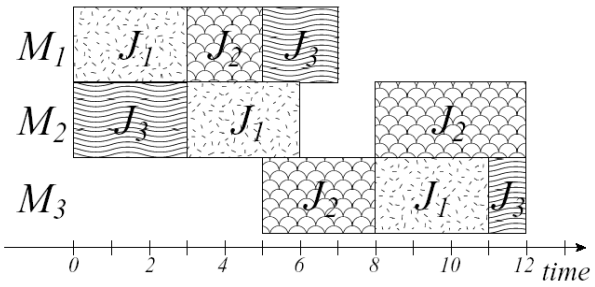


Figura 1.2 - Gráfico de Gantt da solução; fonte (Yamada 1997)

Esta solução significa que na máquina 1 serão processadas seqüencialmente as tarefas 1, 2 e 3; na máquina 2 as tarefas 3, 1 e 2; e finalmente as tarefas 2, 1 e 3 na máquina 3. É importante observar que uma tarefa só pode começar o processamento em uma máquina seguinte se seu processamento na máquina anterior já tiver terminado. O objetivo da otimização é a redução do tempo total de processamento *(makespan)*, que para essa solução corrente foi de *12* unidades de tempo *(times).*

1.2 Formulação Matemática do *Job Shop*

O modelo a seguir foi extraído de Bonfim (2006)

Seja *CM* um conjunto de máquinas e *CJ* um conjunto de tarefas. Seja $O = 0,\ldots,m-1$ o conjunto de operações, onde 0 e $m-1$ representam respectivamente as operações inicial e final de todo o processamento. Neste problema todas as tarefas possuem a mesma quantidade de operações, que é igual ao número de máquinas.

Seja *A* o conjunto de pares ordenados de operações restringidos por relações de precedência para cada tarefa. Para cada máquina *k*, o conjunto E_k descreve o conjunto de todos os pares de operações a serem executadas na máquina *k*, ou seja, operações que não poderão ser sobrepostas. Para cada operação *i*, seu tempo de processamento p_i é fixado, e o tempo possível para início do processamento da operação *i* na máquina *l* é ti_{il}, que é uma variável determinada durante a otimização do problema.

O objetivo é minimizar o tempo total gasto entre o início da primeira operação e o término da última operação, que representa o tempo de processamento total *M*. A variável *M* é representada pelo tempo final tf_{il} de execução da tarefa mais longa. O modelo matemático para este problema é caracterizado:

$$Minimizar \quad M = \underbrace{MAX}_{0 \le l \le m-1} \{ tf_{il} \}, \quad \forall i; 0 \le i \le n-1. \tag{1}$$

Sujeito à :

$$ti_{jk} - ti_{ik} \geq p_i ; \forall (i,j) \in A, \quad \forall i; 0 \leq i \leq n-1, \quad \forall j; 0 \leq j \leq n-1. \tag{2}$$

$$ti_{jk} - ti_{ik} \geq p_i \;\; ou \;\; ti_{ik} - ti_{jk} \geq p_i ; \forall \{i,j\} \in E_k, \quad \forall k; 0 \leq k \leq m-1. \tag{3}$$

$$ti_{ik} \geq 0; \forall i \in O, \quad \forall i; 0 \leq i \leq n-1, \quad \forall k; o \leq k \leq m-1. \tag{4}$$

No modelo matemático do problema, a primeira restrição assegura que a seqüência de processamento das operações em cada tarefa corresponde a uma ordem predeterminada. A segunda restrição denota que existe somente uma tarefa sendo atendida por uma máquina em um determinado momento e, a terceira restrição assegura a execução de todas as operações. Qualquer solução que atenda essas três restrições é chamada de escalonamento.

1.3 Estado da arte

Por ser um problema de difícil tratamento, porém com muita aplicação no cotidiano, escalonamento de tarefas como o *Job Shop* sempre foram muito estudado e grande alvo de aplicação de técnicas emergentes da Pesquisa Operacional e Inteligência Artificial. Inúmeros são os métodos desenvolvidos para tentar resolver este problema: Métodos Exatos, Heurísticas e Meta-Heurísticas, por exemplo.

1.3.1 Métodos Exatos

O espaço de busca para o problema de *Job Shop* cresce fatorialmente de acordo com as dimensões das instâncias de aplicação Cheng et al. (1999). Logo o desenvolvimento de algoritmos para esse problema deve levar em consideração critérios que atenuem tal característica. Assim, resolver problemas demasiadamente grandes comumente encontrados na vida real pode ser inviável em uma abordagem computacional.

Tais algoritmos têm como escolha da solução simples regras aplicadas sobre o escopo geral do problema. O conhecimento prévio das características de um problema específico pode facilitar o processo de busca, diminuindo assim o esforço computacional em busca de boas soluções.

Existem algoritmos que garantem uma solução ótima. Evidentemente esses algoritmos são aplicáveis em problemas pequenos, nos quais o tratamento computacional seja viável. São exemplos na literatura: Giffler e Thompson (1960), Carlier e Pinson (1989), Applegate e Cook (1991), Brucker et al. (1994), Williamson et al. (1997). Tais autores obtiveram bons resultados em instâncias pequenas inclusive a importante 10x10 publicada no trabalho de Fisher e Thompson (1963) e só resolvida vinte anos mais tarde. Problemas de dimensão 15x15 já são considerados fora do alcance dos métodos exatos.

1.3.2 Heurísticas

Para problemas de grandes dimensões se faz necessário à aplicação de Heurísticas. Heurística é um procedimento para resolver problemas através de um enfoque "intuitivo", em geral racional, no qual a estrutura do problema possa ser interpretada e explorada inteligentemente para obter uma solução razoável (Nicholson et al. 1971).

Análises de métodos heurísticos para *Jop Shop* foram feitas por Pinson (1995), Vaessens et al. (1996) e Cheng et al. (1999). Gray e Hoesada (1991) e Gonçalves e Mendes (1994) desenvolveram trabalhos que incluem regras de despachos de tarefas. Heurísticas baseadas em "the shifting bottleneck approach" foram propostas por Adams et al. (1988) e Applegate e Cook (1991). E, algoritmos de busca local por Vaessens et al. (1996), Lourenço (1995) e Lourenço e Zwijnenburg (1996).

1.3.3 Meta-Heurísticas

Uma meta-heurística é um conjunto de conceitos que podem ser utilizados para definir métodos heurísticos aplicáveis a um extenso conjunto de problemas. Atualmente, são as técnicas que encontraram as melhores soluções para o *Job Shop.*

Segundo Melian et al (2003), as Meta-Heurísticas podem ser classificadas da seguinte maneira:

- Meta-heurísticas de busca por entorno*s*: percorrem o espaço de busca levando em conta, fundamentalmente, a "vizinhança" da solução em mãos, definidas

como o conjunto de soluções que podem ser obtidas a partir da aplicação de algum operador à solução atual. São exemplos, *Simulated Annealing* e Busca Tabu;

- Meta-heurísticas de relaxação: simplificam o problema (criando um *problema relaxado*) e utilizam a solução encontrada como guia para o problema original. Exemplo: Relaxação Lagrangeana;
- Meta-heurísticas construtivas: definem de forma meticulosa o valor de cada componente da solução. Exemplo: GRASP;
- Meta-heurísticas evolutivas: lidam com uma população de soluções que evolui, principalmente através da interação entre seus elementos. Exemplo: Algoritmos Genéticos e Sistemas Imunológicos Artificias.

A seguir uma breve abordagem das principais meta-heurísticas usadas em *Job Shop*:

I Simulated annealing

Simulated Annealing é um método de obtenção de boas soluções para problemas de otimização de difíceis resoluções. Desde a sua introdução como uma estratégia de otimização combinatorial, esta vem sendo intensamente utilizada em diversas áreas, tais como projeto de circuitos integrados auxiliado por computador, processamento de imagem, etc.

Simulated Annealing procura evitar a convergência para um mínimo local, aceitando, às vezes, uma nova solução gerada, mesmo que essa incremente o valor da função objetivo *f,* quando o objetivo seria sua redução. A aceitação ou a rejeição de uma nova solução, que causará um incremento de δ em *f* em uma temperatura *T,* é determinada por um critério probabilístico, através de uma função *g* conhecida por função de aceite. Geralmente, essa função é expressa por:

$$g(\delta, T) = e^{-(\frac{\delta}{T})} \tag{5}$$

implicando em:

- a probabilidade de aceitação de uma nova solução é inversamente proporcional ao incremento de δ e
- quando T é grande, a maioria dos movimentos (de um estado para outro ou de uma solução para outra) é aceita, porém, a medida que T tende a zero, a grande maioria das soluções são rejeitadas, ou seja, a probabilidade de se aceitar um vizinho de maior energia decresce com o tempo.

Procurando evitar uma convergência precoce para um ótimo local, o algoritmo inicia com um valor de T relativamente grande. Esse parâmetro é gradualmente reduzido e, para cada um dos seus valores, são realizadas muitas tentativas de se alcançar uma melhor solução, nas vizinhanças da solução corrente.

Matsuo et al. (1988) já definia importantes vizinhanças para o método. Laarhoven et al. (1992) elaborou um algoritmo genérico que tinha o controle do tamanho da vizinhança, tal método foi considerado robusto e de implementação simplificada. Kolonko (1998) evidenciou que a aplicação em *Job Shop* não tinha uma boa convergência e também não havia simetria na definição da vizinhança padrão. Com isso, foi proposto pelo próprio Kolonko um método híbrido de *simulated annealing* e algoritmo genético.

II Busca Tabu

A Busca Tabu na forma clássica define restrições tabus com o intuito de inibir certos movimentos. São criados procedimentos chamados de critério de aspiração para decidir o quanto um movimento considerado como tabu pode ser executado. Desta maneira, este método induz a busca para regiões ainda não analisadas tentando evitar a convergência da solução para ótimos locais.

As restrições tabus, na maioria das vezes, são controladas por uma memória que guarda os últimos movimentos concebidos. O tempo que um movimento pode permanecer na memória está relacionado com o número de iterações do algoritmo e também com o número de movimentos possíveis, dada uma solução candidata.

A definição de como os movimentos são realizados, critério de aspiração e como é feito o gerenciamento da memória dos movimentos estão diretamente relacionados com o

tipo de aplicação desta meta-heurística, resultam assim, na melhoria da solução, considerando estruturas que permitam uma exploração eficiente do histórico de todo o processo de busca.

Uma pesquisa muito importante foi feita por Nowicki e Smutnicki (1996). O método utilizava a idéia de blocos construtivos pertencentes ao caminho crítico, sugerida por Grabowski et AL. (1983), para realizar movimentos regidos por uma lista tabu. Testes computacionais relatados pelos autores mostraram que a estratégia proposta possui um desempenho melhor, em termos de qualidade e tempo, que outros métodos conhecidos na literatura.

Laguna e Glover (1997) a partir da influência de trabalhos anteriores indicam a inclusão da transferência de tarefa, incrementando sua troca que resultou na melhoria da qualidade das soluções. Tal decisão serviu de inspiração para outros trabalhos como: Hara (1995), Barnes e Chambers (1995).

III GRASP

O GRASP, *Greedy Randomized Adaptive Search Procedure*, é um procedimento iterativo, no qual cada iteração consiste na construção de uma solução inicial de forma gulosa, aleatória e adaptativa, e em seguida é aplicada uma heurística de melhoramento, usualmente um procedimento de busca local. A melhor solução dentre todas as iterações será então o resultado final.

A Construção é a fase responsável pela denominação do método, já que é nela que aparecem as características do GRASP: o método guloso (*greedy*), a aleatoriedade (*randomized*) e a adaptação da função gulosa (*adaptive*). Esta fase é iterativa, já que a cada passo é adicionada à solução um novo elemento, até que se tenha uma solução completa e válida. Cada iteração é composta por 3 subfases:

- construção da lista restrita de candidatos, que contém um reduzido conjunto de elementos candidatos a pertencer à solução, definidos por um função gulosa e mecanismos de restrição;
- escolha aleatória do elemento desta lista e inclusão de elemento na solução; e

- adaptação ou reavaliação da função gulosa para os elementos ainda não pertencentes à solução.

GRASP combina as características de heurísticas de construção e de melhoramento. Sua estratégia é acelerar o procedimento de busca local através da construção de soluções próximas, ou ainda, vizinhas a ótimos locais razoáveis e eventualmente ao ótimo global. A fase de construção aperfeiçoa o desempenho médio da fase de melhoria, sendo o GRASP por este motivo mais rápido e eficiente que os métodos de multi-partida (busca local com soluções inicias construídas de forma totalmente aleatória).

Este método trabalha por amostragem em um espaço de soluções reduzido, sendo cada uma das iterações e suas respectivas soluções totalmente independentes entre si. Tal característica o diferencia de outras meta-heurísticas como o *Simulated Annealing* e Algoritmos Genéticos, onde a cada iteração utiliza informação dos passos anteriores para melhorar a solução corrente. Esta independência entre as iterações caracteriza o GRASP como uma heurística naturalmente paralela.

Resende (1997) apresentou uma aplicação para o GRASP em *Job Shop* que consistia nas operações que quando seqüenciadas próximas devem gerar um resultado menor. Aiex et al. (2003) abordou um GRASP paralelo com ligamento de trajetos obtendo melhores resultados que o anterior, Binato et al. (2002) descreveu "a greedy randomized adaptive search procedure for *Job Shop*".

IV Algoritmo Genético (GA)

Geralmente, problemas de escalonamento são resolvidos por heurística ou meta-heurística, o intuito é encontrar a solução ótima ou pelo menos algo perto dela. Tratando-se de um problema NP-difícil, seria impraticável a resolução por uma busca exaustiva, pois as dimensões de problemas práticos são demasiadamente grandes. Desde antes de 1980, o interesse em desenvolver novas Meta-heurísticas já era bastante intenso. Uma delas, Algoritmo Genético, inspirado no processo evolutivo de Darwin, vem sendo reconhecida como uma boa estratégia de método de otimização. Em contraste com outras, *Simulated Annealing* e Busca Tabu, que são baseadas na manipulação de uma única solução factível,

os Algoritmos Genéticos usam uma população de soluções na sua busca, dando-a mais resistência a uma convergência prematura para mínimos locais. Desde a primeira proposta de Algoritmo Genético para resolução de problemas de escalonamento feito por Davis (1985), o uso desta técnica tem aumentado.

Algoritmos Genéticos Clássicos usam uma representação binária para a solução dos problemas. Mas esta representação não é natural em *Job Shop*, pois a representação clássica usa *crossover* e/ou mutação simples, e dessa forma geraria soluções infactíveis. Assim para tratar tal problema Bierwirth (1995) propôs uma nova representação cromossômica. Bierwirth definiu um algoritmo genético de permutação generalizada com o intuito de melhorar os métodos anteriores. Dorndorf e Pesch (1993) e Yamada e Nakano (1992) usaram algumas variações dos operadores genéticos padrões e também hibridizações de algoritmos existentes. Em 1995, Dorndorf e Pesch, utilizaram uma representação baseada em grafo disjuntivo (disjunctive-graph-based representation). Yamada and Nakano (1995) desenvolveram aperfeiçoamentos baseados na representação de grafos disjuntivos e em 1996 eles introduziram estratégias de blocos críticos de vizinhança e usaram a técnica repasse de processos baseados em esquema estocástico que tratava distâncias entre pais e filhos.

Bagchi et al. (1991), Davis (1985) adotaram construtores que executavam a transação da representação cromossômica para um escalonamento factível, já que codificação não representava diretamente o escalonamento da solução. Nakano e Yamada (1991) e Tamaki e Nishikawa (1992) usaram procedimentos de factibilização para encontrar escalonamentos factíveis a partir de infactíveis. Este tratamento foi chamado de "forcing". Nestas pesquisas, apareceram evidências de que codificações que não mantinham a factibilidade necessitavam de tempo adicional de execução. Se a representação cromossômica sempre mantivesse a factibilidade depois de uma operação genética, o algoritmo era simplificado, pois não era necessária a aplicação de procedimentos como o "forcing".

Na maioria das pesquisas, a geração da população inicial era um procedimento randômico. Entretanto, uma busca local era aplicada, pois com uma boa inicialização da população, aumentava a possibilidade de encontrar boas soluções finais. Assim, é possível

obter bons resultados se o algoritmo produzir uma população inicial factível, diversificada e de qualidade, junto a operadores genéticos proporcionando uma nova população que mantenha as características desta anterior.

Um problema comum em um Algoritmo Genético é a convergência prematura. Embora este tenha mais resistência à convergência prematura do que métodos de busca local, ele não está imune. Uma aproximação para amenizar esse problema é a paralelização do Algoritmo Genético com sub-populações disjuntas, *Fitness Sharing* ou outras opções que caracterizam uma população estruturada (Mahfoud, 1995).

Devido à dificuldade de se encontrar boas soluções no problema de *Job Shop*, novas técnicas híbridas de algoritmo genético com outras heurísticas são desenvolvidas, como: Wang, L. e Zheng, D. (2001), Gonçalves et al. (2002) e Caldeira et al. (2004) Todos eles obtiveram resultados expressivos basicamente fundindo algoritmo genético e procedimentos que permitem busca local. Gonçalves et al. (2005) propuseram uma meta-heurística de Computação Evolutiva que obteve resultados expressivos para este problema; o algoritmo se baseia em uma representação com prioridade de operações e atraso de alocação de cada tarefa, também foi utilizada uma busca local aplicada nas fronteiras dos blocos construtivos pertencentes ao caminho critico.

Capítulo 2

Sistema Imunológico Biológico

Resumo – Este capítulo traz uma introdução aos conceitos básicos referentes ao Sistema Imunológico Biológico (SIB); as abordagens feitas aqui foram baseadas em De Castro (2001), autor que reuniu as características necessárias desse sistema para formalizar em seu Doutorado a Engenharia Imunológica. Para a aplicação neste trabalho são evidenciadas as funcionalidades do SIB consideradas importantes na resolução do problema de *Job Shop*.

2.1 Introdução

Em 1882, Elie Metchnikoff realizou um experimento que consistiu em espetar um acúleo de uma roseira em uma larva transparente de estrela-do-mar; foi verificado 24 horas mais tarde um aumento de células circundando a extremidade do espinho que se projetava na larva. Baseado nas observações de *Fagócitos*, termo criado por Metchnikoff, ele inferiu que esta atividade seria importante na manutenção da integridade do organismo submetido a um agente estranho. Tais observações o levaram a criar um conceito básico de *imunologia*, ramo da biologia desconhecido até então (Wikipédia, 2006).

Atualmente existem conceitos aprimorados e formalizados de *Imunologia*, por exemplo: *Imunologia* é o ramo da biologia que estuda as reações de defesa que proporcionam resistência às doenças (Klein, 1990); ou o sistema que atua na defesa de um animal contra o ataque constante de *microorganismos* é chamado de *Sistema Imunológico* (Tizard, 1995).

Hoje em dia, os Sistemas Imunológicos Biológicos são muito estudados. Especialistas, como biólogos e médicos, despendem esforços para melhor compreendê-los, modelá-los a fim de reproduzir fenômenos visualizados em laboratórios e também tentar prever novos comportamentos ainda não observados. O principal foco de todo esse

interesse está no combate de doenças infecciosas como AIDS, doenças auto-imunes como artrite reumatóide e esclerose múltipla, e outras.

Na área de pesquisa operacional, bem como aplicações em computação, o estudo de sistemas imunológicos vem se difundido devido, principalmente, à sua capacidade de processamento eficiente de informações. Outras características, inerentes ao (SIB), de extrema utilidade para este novo viés de pesquisa são destacadas abaixo (De Castro, 1999):

- Unicidade: cada indivíduo possui seu próprio sistema imunológico, com particulares capacidades e vulnerabilidades;
- Detecção de anomalias: o SIB pode reconhecer e reagir a agentes causadores de anomalia nunca expostos anteriormente;
- Reconhecimento de padrões externos e internos ao sistema: moléculas e células que não são nativas do corpo são reconhecidas e eliminadas pelo SIB;
- Detecção distribuída: as células do sistema são distribuídas por todo o corpo e não são sujeitas a um mecanismo de controle central;
- Detecção com tolerância a ruídos: não é necessário um reconhecimento absoluto para que o sistema reaja contra elementos causadores de patologias;
- Diversidade: em um SIB, o número de moléculas e de células são limitados e responsáveis por reconhecer e combater uma quantidade praticamente infinita de agentes que serão expostos no decorrer da vida do indivíduo; e
- Aprendizagem por reforço e memória: o sistema pode aprender as estruturas dos patógenos melhorando a qualidade das suas respostas; e quando novamente expostos a agentes já conhecidos a atuação do sistema é mais forte e eficiente.

O sistema imunológico é uma coleção de moléculas, células e organismos cuja complexa interação em um sistema eficiente é usualmente capaz de proteger um indivíduo. Este sistema é susceptível a outras influências como vacinas (maneira artificial de elevar a imunidade do indivíduo). Está estruturado em múltiplas camadas, com funções e mecanismos de defesa espalhados em diferentes níveis (Figura 2.1). As camadas constituintes do SIB podem ser divididas na ordem de ativação a seguir (Janeway et al., 2000):

- Barreiras físicas: a pele e o sistema respiratório atuam como uma primeira barreira a ser superada por um agente;
- Barreiras bioquímicas: fluidos corporais como saliva, lágrima, suor, ácidos estomacais, etc, impedem a entrada e ou eliminam boa parte dos microorganismos quando em contato;
- Sistema imune inato: é formado por células fagocitárias, como os *macrófago*s e os *neutrófilos*, além de fatores solúveis e algumas enzimas. Esta camada desempenha uma importantíssima função de primeiro combate a agentes estranhos, e também é responsável pelo posterior direcionamento das respostas imunes adaptativas. Este sistema atua controlando as infecções enquanto o sistema imune adaptativo é ativado por completo, já que esta resposta demora períodos de tempo (ordem de dias); e
- Sistema imune adaptativo: Os *linfócitos* são as principais células que compõem o sistema imune adaptativo. Eles são responsáveis em identificar agentes patogênicos, produzir e secretar anticorpos como resposta. A teoria de Sistemas Imunológicos Artificiais (SIA) é fundamentada em conceitos e funcionalidades do sistema imune adaptativo que será descrito detalhadamente a seguir.

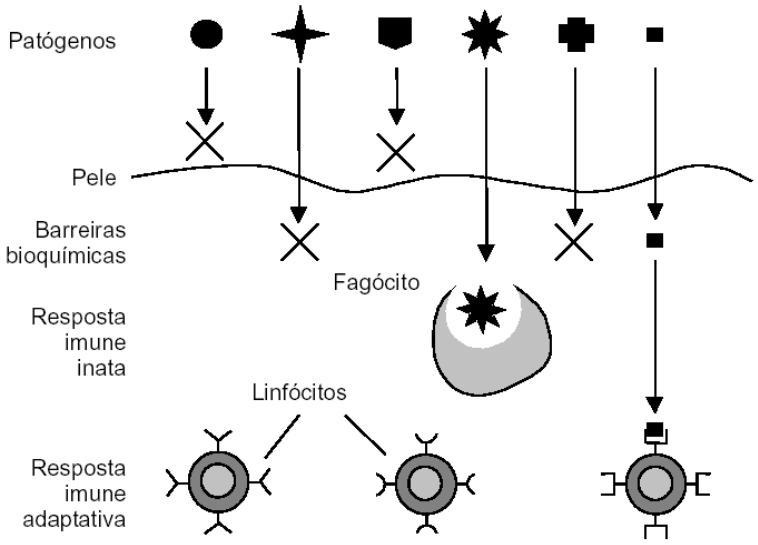


Figura 2.1 - Sistema imunológico biológico estruturado em multicamadas funcionais descritas acima. Fonte: De Castro (1999).

2.2 Sistema imune adaptativo

As células principais que compõem o sistema imune adaptativo são os linfócitos. Eles podem ser classificados em células B ou células T, segundo a sua origem. Células do tipo B são produzidas na medula óssea *(Bone marrow)* e a sua estrutura é formada pelo receptor de antígenos BCR (Figura 2.2), esse receptor corresponde ao *anticorpo* ligado a sua membrana que será secretado quando a célula for ativada.

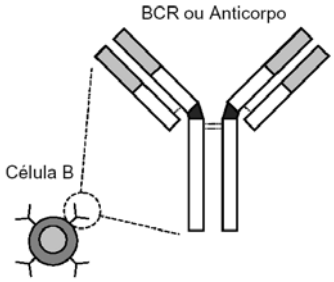


Figura 2.2 - Ilustração de uma célula B com a molécula de anticorpo em sua superfície. Os anticorpos são responsáveis pelo reconhecimento de agentes estranhos no organismo. A região sombreada é a porção, aminoterminal, que sofre variações no processo de mutação. Fonte: De Castro (2001)

Os *epítopos* são as porções de um antígeno que podem ser reconhecidas por um anticorpo. Cada anticorpo possui um único tipo de receptor, porém os antígenos possuem diversos epítopos (Figura 2.3). As células B são capazes de reconhecer antígenos livres e a sua principal função é a produção e secreção de anticorpos como respostas a agentes patogênicos. Assim, um único antígeno pode ser reconhecido por vários anticorpos.

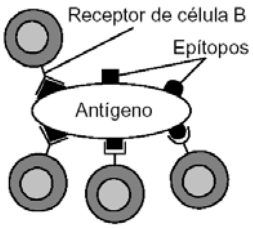


Figura 2.3 - Reconhecimento de antígenos pelos anticorpos. Um antígeno possui diversos epítopos, podendo ser reconhecido por diversos anticorpos. Fonte: De Castro (2001)

As células B são capazes de reconhecer antígenos livres e a sua principal função é a produção e secreção de anticorpos como respostas a agentes patogênicos. As células do tipo T, maturadas no *timo* (Dreher,1995), também possuem receptores de antígenos, os TCR. Diferentemente dos BCR, eles reconhecem apenas antígenos processados, fragmentos sob a forma de *peptídeos* e ligados a uma molécula de *histocompatibilidade* principal. Funções importantes dessas células são: o regulamento das ações de outras células, e o combate a células infectadas do organismo hospedeiro.

2.2.1 Diversidade imunológica e reconhecimento de padrões

Estudos sobre cadeias *polipeptídicas* da molécula do anticorpo mostraram que ela é composta por duas regiões distintas, Figura 2.2. A primeira é a região *aminoterminal* que possui características de ser extremamente variável e é responsável pelo reconhecimento antigênico. A segunda, *carboxiterminal*, é uma região constante de poucos tipos distintos com funções *efetoras* (biológica) de complemento e ligação a outros receptores celulares. A formação da molécula do anticorpo e também do TCR é feita a partir de uma composição de múltiplos segmentos distintos que codificam os seus receptores a partir de uma biblioteca de fragmentos gênicos. Devido às altas taxas de mutações somáticas *(hipermutação)* e recombinações gênicas a diversidade de informação genética codificada no genoma é elevada.

Quando um antígeno ativa uma célula B pela primeira vez, ocorre uma geração de uma população celular com ampla diversidade em sua especificidade, possibilitando assim o surgimento de células compatíveis com o estímulo antigênico. Na proliferação (clonagem) das células emergentes compatíveis, a mutação somática é acionada. Por trocas de bases nucleotídicas, esta hipermutação refina a resposta imunológica ao antígeno reconhecido. Isso permite a criação de moléculas de *imunoglobulina* (anticorpo) capazes de reconhecer aquele antígeno com maior afinidade. Estes mecanismos de geração e mutação proporcionam um reconhecimento poderoso de antígenos, uma vez que o sistema imunológico pode produzir um número praticamente infinito de receptores celulares a partir de um genoma finito.

2.2.2 Memória e aprendizagem

A célula B pode se diferenciar de maneiras diferentes quando ativadas; uma possibilidade é a diferenciação em *plasmócitos* secretando anticorpos com altas taxas. Outra maneira é a diferenciação em *células B de memória* (Figura 2.4). Estas são caracterizadas por um longo período de vida e circulam pelo sangue, tecidos e vasos linfáticos. Essa característica é muito importante, pois quando o organismo é infectado outra vez por um mesmo antígeno, *as células B de memória* se diferenciam em plasmócitos que produzirão anticorpos pré-selecionados e robustos em relação ao antígeno específico estimulante (Ada e Nossal, 1987).

A exposição repetida a agentes antigênicos, junto à produção de células de memória, estimula a aprendizagem do sistema imunológico. Esta aprendizagem envolve o aumento de clones específicos e também da afinidade do receptor (anticorpo-antígeno), podendo resultar na diminuição da produção de outros clones não específicos, uma vez que a quantidade de linfócitos no organismo é controlada. Tal peculiaridade do sistema imunológico é característica de uma *estratégia de aprendizagem por reforço*: um sistema está continuamente evoluindo, promovendo melhoras na execução de tarefas requeridas (Sutton e Barto, 1998).

2.2.3 Princípio da seleção clonal

O principio da seleção clonal está associado às características básicas de uma resposta imune adaptativa a um estimulo antigênico. Apenas células que forem capazes de reconhecer estímulos antigênicos são selecionadas para se proliferar. Células que não foram capazes são eliminadas, vide Figura 2.4. A seleção clonal ocorre nas células T e nas células B, sendo que as células T não secretam anticorpos, mas são importantes, pois regulam as respostas das células B. Um resumo das principais características da seleção clonal pode ser feito da seguinte maneira (Burnet, 1978):

- Os novos linfócitos diferenciados, capazes de reagir com padrões antigênicos expressos por elementos do próprio organismo (denominados antígenos

próprios) são inativados ou eliminados, assegurando assim uma tolerância ao próprio;

- A interação de uma molécula estranha com um receptor de linfócito capaz de ligar-se a essa molécula leva a ativação linfocitária;
- Geração de variações genéticas aleatórias, através de um mecanismo de hipermutação somática, expressa sob a forma de diversos tipos de anticorpos;
- Restrição *fenotípica* de um padrão para cada célula diferenciada e retenção deste padrão pelos descendentes clonais.

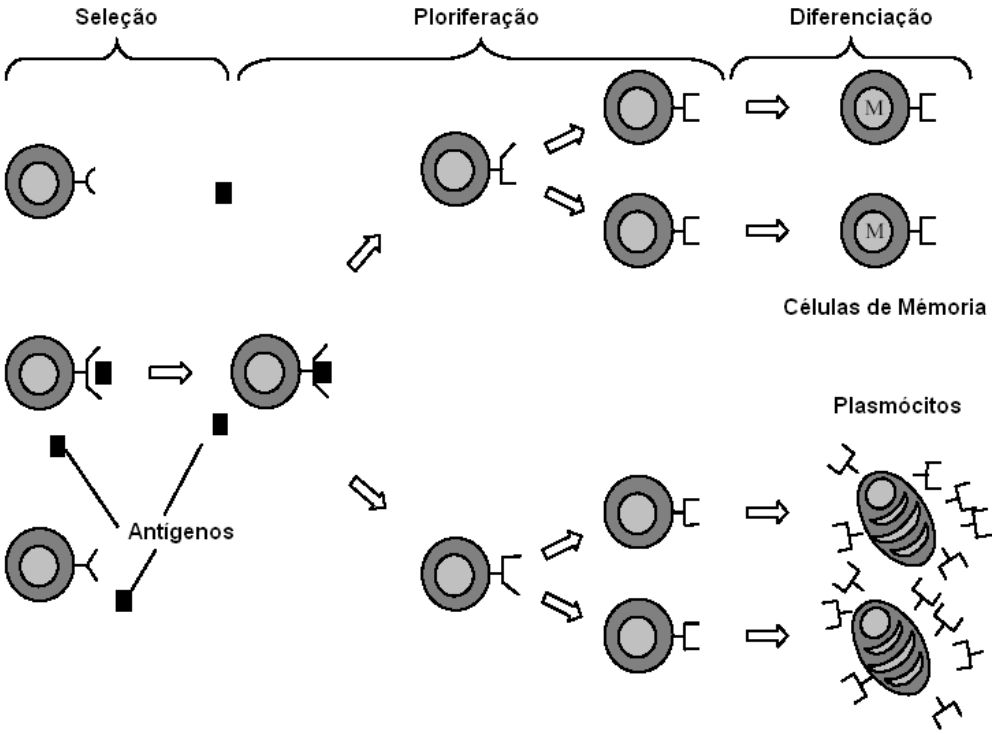


Figura 2.4 - Etapas simplificadas do processo imunológico. Reconhecimento de antígenos pelos anticorpos, seleção seguida da proliferação, especialização do anticorpo para o reconhecimento do antígeno e por fim a diferenciação em plasmócitos e em células de memória. Fonte: De Castro (2001)

2.2.4 Mutação e afinidade

Os mecanismos utilizados na variação genética dos receptores das células B são basicamente a hipermutação somática e a edição de receptores (George e Gray, 1999). Somente os anticorpos que obtiveram reconhecimento antigênico, anticorpos afins, são clonados e mutados promovendo assim variações genéticas (Figura 2.4); e os descendentes de maior afinidade são selecionados para serem células de memória. Em média, os anticorpos das respostas secundárias (memória) possuem maior afinidade do que os das respostas primárias. Este fenômeno, que é específico às respostas dependentes das células T, é chamado de *maturação de afinidade*.

São conhecidos alguns tipos de mecanismos de mutação na região *aminoterminal* do anticorpo. Seguem três situações que podem ser observadas (Allen et al., 1987):

- Mutações pontuais;
- Pequenas deleções; e
- Troca não recíproca de seqüências, seguindo uma conversão genética.

Além disso, George e Gray (1999) defenderam a existência de um mecanismo de edição de receptores durante a maturação de afinidade. Tal processo permitiria uma maior flexibilidade e mobilidade na variação gênica dos anticorpos. É natural que com a aleatoriedade na natureza dos processos de variações genéticas, como por exemplo, na hipermutação, grande parte dos anticorpos alterados torne-se não funcionais, desenvolvendo receptores auto-reativos ou anticorpos de baixa afinidade (Storb, 1998). O controle de inativação e eliminação dessas células é feito pelo *Mecanismo de seleção* que será descrito posteriormente.

Também existe o mecanismo de controle na hipermutação que age de forma a aumentar o desempenho na obtenção de novas células de alta afinidade. Para uma rápida maturação na resposta imunológica é requisitado um ligeiro acúmulo de mutação. Como descrito acima, a maioria das variações nos anticorpos os deixarão não-funcionais, entretanto, quando uma célula que sofreu mutação é capaz de aumentar sua afinidade antigênica, a alta taxa de mutação pode provocar mudanças indesejadas, causando a perda

das variações que levaram ao aumento da afinidade. Sendo assim, um controle na taxa da mutação se faz necessário, um curto pico de hipermutação somática, seguido de um intervalo para seleção e expansão clonal são inerentes no processo de maturação.

É importante que o mecanismo de seleção forneça um controle dependente da afinidade do anticorpo no processo de hipermutação. Assim, anticorpos com baixa afinidade permanecem sendo mutados, enquanto os que atingiram alta afinidade devem ter sua mutação regulada ou até inativada.

2.2.5 Mecanismos de seleção

O sistema imunológico em sua capacidade de reconhecimento é completa. A sua funcionalidade também abrange distinções de células e moléculas do próprio organismo (distinção própria). Esta distinção é de extrema importância, já que uma resposta imunológica a antígenos próprios desencadeará disfunções orgânicas como doenças auto-imunes (De Castro, 2001). A *seleção negativa* desempenha o controle dessa funcionalidade. Quando um linfócito reconhece um antígeno próprio, estas células B e T são eliminadas do repertório imunológico ou inativadas.

A seleção positiva nas células B e T tem a função de selecionar quais os linfócitos que são capazes de atuar como células *imunocompetentes* numa resposta imune adaptativa (De Castro, 1999). Os linfócitos B quando ligados a um antígeno sofrem proliferação e hipermutações. As células filhas mutantes, que reconhecem o antígeno estimulante mais eficientemente, são selecionadas para expansão, sendo resgatada da morte celular.

Dentre toda a complexidade do sistema imunológico com alta capacidade de reconhecimento de agentes próprios e não-próprios do organismo, também é observado a *completude imunológica*, reconhecimento anticorpo-anticorpo (De Castro, 2001). Os anticorpos possuem *idiotopos imunogênicos*, ou seja, eles possuem *epítopos* que podem ser reconhecidos por outros anticorpos, princípio básico da teoria de *Redes Imunológicas*, não focada para a aplicação neste trabalho.

Capítulo 3

Sistema Imunológico Artificial e Planejamentos

Resumo – Neste capítulo foi feita uma revisão e análise de trabalhos onde foram abordados Planejamentos por Sistemas Imunológicos Artificiais para a geração de escalonamentos de tarefas. Em um sistema computacional imuno-inspirado é muito importante que a codificação dos anticorpos, os operadores de seleção, maturação de afinidade e controle de supressão sejam robustos e apropriados para as peculiaridades do cenário de aplicação. Os trabalhos a seguir mostram diversas maneiras de definições desses atributos em escalonamento de tarefas.

3.1 Introdução

Um sistema biológico é muito complexo e possui grande variedade de componentes e funcionalidades. Para o desenvolvimento de sistemas computacionais inspirados nos naturais, é necessária uma simplificação da realidade de maneira a enquadrar os conceitos, julgados mais relevantes, para aplicação no problema em questão.

O Sistema Imunológico Biológico é uma coleção de moléculas, células e organismos cuja complexa interação num sistema eficiente é usualmente capaz de proteger um indivíduo de invasores ou de suas próprias células alteradas devido à doenças. Ele é subdividido em Sistema Imune Inato, que é formado por células fagocitárias responsáveis por um primeiro combate aos *patógenos,* e Sistema Imune Adaptativo, composto por células (*linfócitos*) que evoluem para proporcionar meios de defesas mais versáteis e um maior nível de proteção face às novas infecções pelo mesmo agente. Os linfócitos sofrem um processo semelhante ao da seleção natural (Lederberg, 1988), no qual somente aqueles que encontram um antígeno com o qual seu receptor pode interagir serão ativados para proliferar por clonagem. Estes clones irão se diferenciar em *plasmócitos,* secretando os anticorpos com a mesma especificidade do receptor; ou células de memória, quando o

sistema imunológico estiver novamente exposto a um determinado antígeno, estas células são ativadas a fim de obter uma resposta futura mais eficiente. Na clonagem das células com alta afinidade, a *hipermutação* somática é acionada para refinar, por trocas de bases *nucleotídicas*, a resposta imunológica ao antígeno reconhecido. Isso permite a criação de moléculas de *imunoglobulina* (anticorpo) capazes de reconhecer aquele antígeno com maior eficiência. Estes mecanismos de geração e mutação proporcionam um reconhecimento poderoso de antígenos, uma vez que o sistema imunológico pode produzir um número praticamente infinito de receptores celulares a partir de um genoma finito. Os anticorpos desempenham o principal papel no Sistema Imunológico, pois eles são capazes de se aderir ao antígeno com o objetivo de neutralizá-los e marcá-los para que outras células do sistema imunológico os eliminem.

Dado o escopo de características biológicas, cada aplicação do Sistema Imunológico Artificial busca, através de adequações das funcionalidades citadas, abordagens mais robustas e eficientes onde outras estratégias de solução se mostrariam inócuas (De Castro, 2001). Segundo Dasgupta (1998), os SIA's são mecanismos computacionais compostos por metodologias inteligentes, inspirados no Sistema Imunológico Biológico, para a solução de problemas do mundo real. Muitas das suas propriedades são vantajosas em diversas aplicações como *Segurança Computacional*, *Detecção de Anomalia em Base de Dados, Reconhecimento de Padrões* e outros. Este capítulo traz exemplos de resolução de problemas de escalonamento por Sistema Imunológico Artificial.

3.2 Exemplos de Trabalhos

Abaixo são apresentados alguns trabalhos relevantes que utilizaram Sistemas Imunológicos Artificiais como ferramenta para a resolução de escalonamento de tarefas.

I Sistema de Planejamento Adaptativo Inspirado em Sistema Imunológico

Adaptive Scheduling System Inspired by Immune System (Mori et al. 1998)

Os autores propuseram um algoritmo imuno-inspirado para encontrar escalonamentos no qual o pacote de operações em cada recurso poderia ser variável. A Figura 3.1 exemplifica o modelo deste tipo de escalonamento.

Este problema está subdividido em outros dois: (1) determinar o tamanho do lote apropriado para cada ordenação e (2) determinar a maneira de como é feita (planejada) cada uma destas ordenações, buscando:

- Minimização da soma dos tempos de *setup*;
- Minimização dos tempos de espera; e
- Minimização do tempo máximo para completar todas as operações.

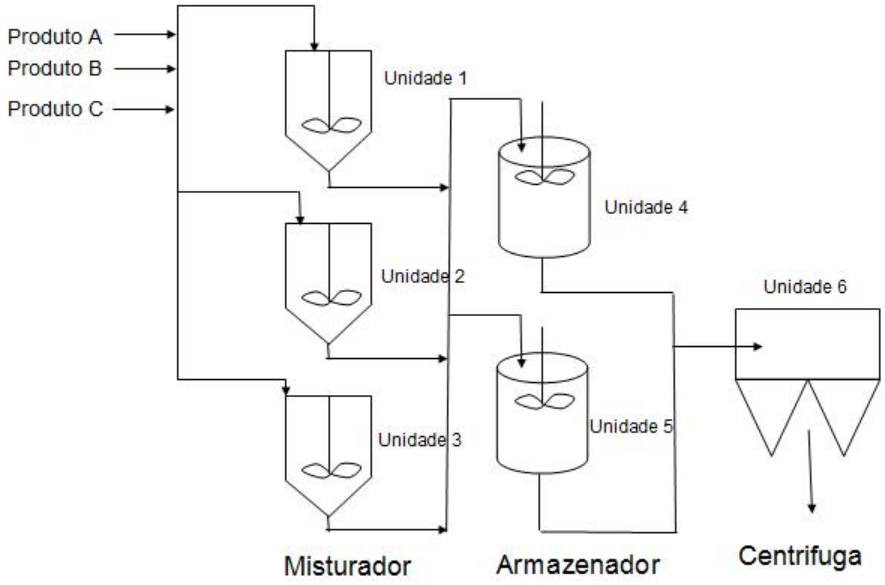


Figura 3.1 - Exemplo de cenário de aplicação de escalonamento com lotes de operações variáveis; fonte (Mori et al, 1998).

Além disso, com restrições:

- Entre processos e máquinas;
- Tempo de processo depende do tipo do produto;
- Tempo de processo depende do tipo da máquina;
- Tempo de processo depende do tamanho do lote;

- Tempo de *setup*; e
- Capacidade de processamento.

No algoritmo 3.1 proposto e descrito em 6 passos, existem dois tipos de anticorpos representados de forma vetorial: anticorpo do Tipo I que codifica o tamanho dos lotes; e Tipo II que são as prioridades das operações.

Algoritmo 3.1 – Algoritmo de alto nível de Mori et al. (1998)

[1] Reconhecimento de antígeno: o sistema reconhece a invasão de antígenos, que significa a entrada de dados ou um distúrbio no sistema.

[2] Produção de anticorpos a partir de células de memória: as células que foram eficientes em "matar" antígenos no passado são ativadas para produzir anticorpos. Isso corresponde a boas soluções passadas.

[3] Cálculo da afinidade: avaliação das soluções.

[4] Diferenciação dos linfócitos: o anticorpo que reconheceu o antígeno é direcionado para a memória com o intuito de responder estímulos futuros mais rapidamente. Esse direcionamento corresponde ao armazenamento de uma solução na base de dados.

[5] Supressão dos anticorpos: quando um anticorpo tem alta afinidade sua proliferação é excessiva. Para ter o controle desse fenômeno determina-se a concentração de anticorpos na região de busca. Este cálculo é feito pela afinidade baseando-se no fenótipo e não no genótipo, ajudando a diminuir o esforço computacional do sistema.

[6] Proliferação dos anticorpos: para responder aos estímulos antigênicos, novos anticorpos são produzidos e repassados para substituírem os que foram eliminados em [5]**.** Este procedimento pode gerar diversidade através de operador de reprodução genética, mutação e outros.

Os autores mostraram que a recombinação somática e a mutação ajudam no aumento da diversidade dos anticorpos, passo [6]. Também foi visto que o reconhecimento mútuo entre os anticorpos contribui para o controle da proliferação [5]. Sendo assim, é possível obter boas soluções candidatas ao problema com uma busca paralela por vizinhança no espaço.

II Minimização de tempo de processo em processadores paralelos: Uma abordagem Imuno-Inspirada

Makespan Minimization on Parallel Processors: An Immune-Based Approach (Costa et al. 2002)

Máquinas paralelas é outro tipo de escalonamento que se constitui em um conjunto de tarefas que precisam ser realizadas. A diferença em relação aos demais é que todas as máquinas (recursos) são iguais. Por exemplo, se for preciso fazer um trabalho de datilografia, o corpo do texto pode ser dividido em partes direcionando-as para um grupo de datilógrafos que executam a mesma tarefa. Porém cada datilógrafo tem sua habilidade, tempo de duração para realizar um trabalho. O escalonamento consiste em distribuir estas tarefas para os datilógrafos (recurso) de maneira a minimizar o tempo de realização de todo o processo.

Como os demais, Máquinas Paralelas é transcrito de maneira simples para um conjunto de tarefas que podem ser processados em um conjunto de máquinas idênticas, podendo também ter as características de tempo de configuração, tempo de espera, capacidade de processamento e etc.

Costa et al. (2002) propuseram um Sistema Imunológico Artificial baseado em Seleção Clonal e maturação de afinidade para a minimização do *makespan* de processos paralelos (Máquinas Paralelas). Além disso, os autores implementaram duas heurísticas construtivas (LPT e *Multifit*), uma *Simulated Annealing* e uma Busca Local (heurística de melhoria para as demais) com o intuito de confrontar os resultados com o SIA.

No algoritmo proposto, uma solução candidata foi codificada por um vetor com o número de posições igual ao número de tarefas, cada posição desse vetor continha a máquina que a respectiva tarefa fora alocada. Logo após, operadores de seleção e mutação interagem promovendo a evolução do repertório. O algoritmo é descrito de forma simplificada a seguir:

Algoritmo 3.2 – Pseudo código de Costa et al. (2002). Seleção Clonal e Maturação

Cria um repertório de k anticorpos (soluções factíveis do problema)

Para cada geração **do**
    Para cada anticorpo**, do**
        decodifica o anticorpo e determina sua afinidade;
    determina o número de clones para cada anticorpo;
    determina o número de mutações;
    do clonagem e mutação;
    Para cada clone, **do:**
        decodifica o clone;
        determina sua afinidade;
        if afin(clone) > afin(anticorpo) → anticorpo = clone;
While critério de parada = **false.**

O repertório inicial é gerado aleatoriamente além da inserção de um anticorpo vindo da LPT + Busca Local. A seleção clonal se dá pela avaliação da afinidade, ou seja, a qualidade de uma solução. Dada a seleção de um anticorpo, o número de clones gerados é diretamente proporcional a sua afinidade e o número de mutações sofridas inversamente. O uso dessa estimativa para definir o número de clones e a taxa de mutação é inerente a uma abordagem imuno-inspirada.

Foram definidos cinco tipos de mutações,

- Mutação1: uma tarefa é escolhida randomicamente e associada a uma máquina randomicamente;
- Mutação2: troca-se aleatoriamente duas tarefas escolhendo duas máquinas de forma randômica;
- Mutação3: troca-se uma tarefa da máquina mais carregada para a menos carregada;
- Mutação4: troca-se uma tarefa da máquina mais carregada para uma outra aleatória;
- Mutação5: (por exemplo, com dois processos), troca a máquina de cada tarefa de uma parte do anticorpo k (tarefas pertencentes a máquina 0 agora pertencerão a máquina 1 e vice e versa);

Todas as mutações assumem o compromisso de reforçar a explotação e a exploração do espaço de busca.

Os resultados obtidos pelo SIA mostraram-se superiores aos das outras estratégias, principalmente quando avaliada a diversidade do grupo de soluções encontradas para cada cenário.

III Um Algoritmo Imunológico Para o Planejamento do Flow Shop Flexivel

An immune algorithm approach to the scheduling of a flexible PCB flow shop (Alisantoso et al. 2003)

Alisantoso et al. (2003) publicaram um Sistema imunológico Artificial para Planejamento de *Flow Shop* Flexível com operadores de mutação e *crossover* baseados em Seleção por Roleta *(Roullete Wheel Selection)*.

O problema de *Flow Shop*, visto no Capítulo 1, pode ter uma abordagem específica ou a Flexível, vide Figura 3.2. O grafo de Gantt (A) da Figura 3.2, ilustra um Flow Shop Simples com 3 processos (P1-P3) e 2 Tarefas (J1 e J2). As duas tarefas passaram nas três máquinas na ordem P1, P2 e P3. Primeiro inicia com o J1 na máquina 1, quando este é terminado, vai para a máquina 2 e em paralelo o J2 é iniciado na máquina1. A solução é formada quando todas as tarefas forem executadas na mesma ordem de máquinas.

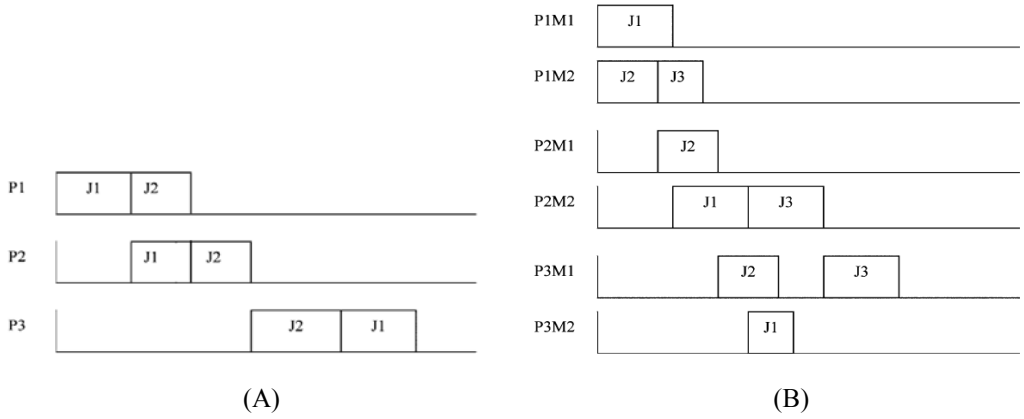


Figura 3.2 - *Flow Shop*: exemplos de soluções representadas por grafo de Gantt para os dois tipos de Folow Sohp, Flow Shop Simples (A) e Flow Shop Flexível (B); fonte (Alisantoso et al, 2003).

O *Flow Shop* Flexível é uma extensão deste anterior. Aqui, cada processo pode ter mais de uma máquina para executar as tarefas na cadeia de ordenação fixa. O grafo de Gantt (B) da Figura 3.2, mostra um sistema com 3 processos (P1-P3), 2 máquinas (M1 e M2) por processo e 3 tarefas (J1-J3). Todas as tarefas serão processadas na ordem P1, P2 e P3, uma tarefa pode ser designada na máquina 1 ou 2 em cada processo.

Neste cenário, a minimização do tempo de processamento das tarefas, tempo de espera, reduções dos custos de configuração e inventário podem ser focados na função objetivo do problema de otimização. O diagrama abaixo representa de forma simplificada o SIA desenvolvido na aplicação deste problema de *Flow Shop* Flexível:

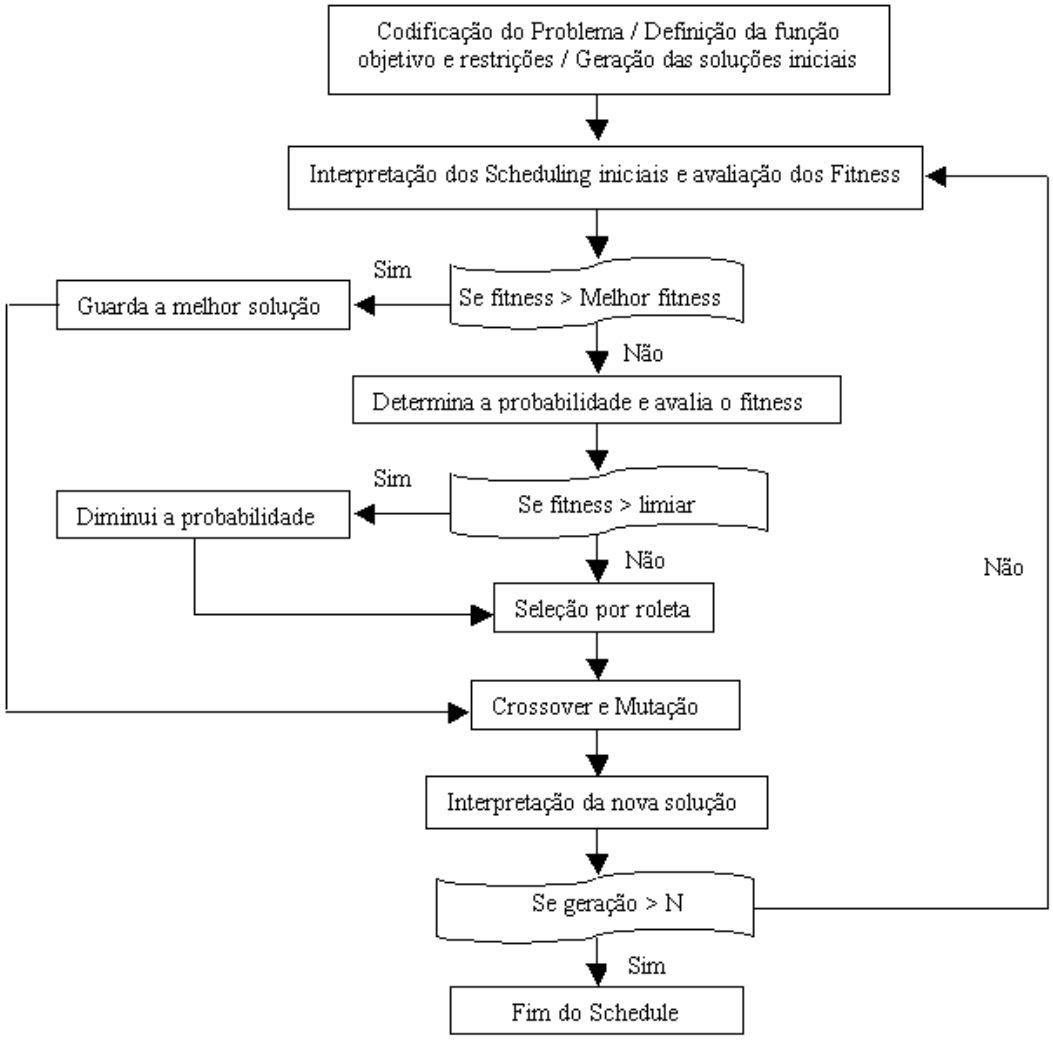


Diagrama 3.1 – Diagrama de processo do SIA aplicado por Alisantoso et al (2003).

No algoritmo imunológico proposto aqui, os antígenos referem-se à função objetivo que precisar ser otimizada. Um candidato a solução é representado por uma seqüência das tarefas e designações sobre as máquinas em cada processo (Figura 3.3); isso implica que a ordem que as tarefas aparecem na seqüência, determinam a ordem que eles devem ser executados.

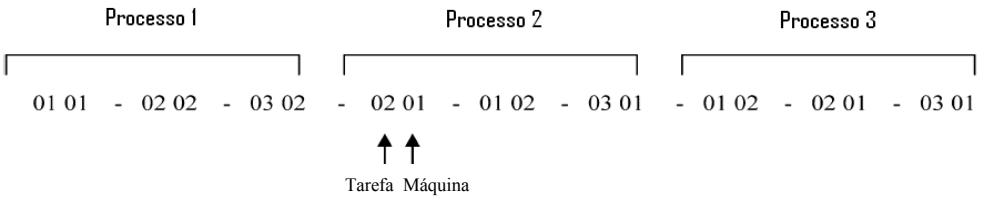


Figura 3.3 - Codificação da solução

Neste contexto, para o Processo 1 a ordem das tarefas a serem executadas é a tarefa1 seguida da tarefa2 e por fim a tarefa3. É importante notar que um tarefa pode ser executada antes de uma outra, porém em uma máquina diferente. Essa codificação pode apresentar ambigüidade (soluções de diferentes codificações que representam a mesma solução), como por exemplo, 0202-0101-0302-0102-0201-0301-0201-0301-0102, que representa o mesmo escalonamento (solução) que foi apresentado na Figura 3.3.

Os operadores de variações genéticas usados foram o *crossover* e a *mutação*. A partir do *Crossover OX* modificado e mutações em pares tarefa/máquina alternando aleatoriamente a máquina dessa referente tarefa, o conjunto de indivíduos do sistema obteve variações genéticas com boa exploração do espaço de busca.

IV Produzindo Escalonamentos Robustos via Sistema Imunológico Artificial

Producing Robust Schedules Via an Artificial Immune System (Hart et al. 1998)

Um sistema imunológico artificial capaz de desenvolver arranjos e seqüências de maneira rápida foi proposto por Hart et al. (1998). De forma eficiente, esse sistema trata circunstâncias peculiares comumente encontradas, por exemplo, em linhas de montagem de manufatura.

Hart et al. (1998) fazem as seguintes analogias entre o sistema imunológico biológico e o ambiente de planejamento de uma fábrica:

Planejamento	**Sistema Imunológico**
O escalonamento que emerge através da evolução de um algoritmo genético	Resposta Imunológica
Conjunto antigênico que representa as situações possíveis nos cenários de aplicação (ambiente de produção)	Universo antigênico
Sequenciamento de tarefas em uma máquina dado um determinado cenário	Antígeno
Partes de uma seqüência de tarefas que são comuns a mais de uma máquina	Anticorpo
Número limite máximo de tempo que uma tarefa pode estar atrasada em um escalonamento produzido pela interação antígeno-anticorpo. Uma situação ideal de escalonamento é quando o tempo de espera é *0*, ou seja, valor de ligação igual a zero. Quanto maior for este valor menor será a qualidade do escalonamento.	Valor de ligação
A ligação anticorpo-antígeno ocorre se o escalonamento produzido utilizando as instruções dadas por um anticorpo e a informação fornecida por um antígeno resultarem na realização da tarefa antes do prazo final especificado pelo antígeno	Ligação *(matching)*

Tabela 3.1 - Ilustração da relação entre o SIA e o Planejamento

Como enfatizado anteriormente, para a simulação de um cenário bio-inspirado, é necessária a simplificação das características importantes no desenvolvimento em questão:

- O ambiente de operações de uma fábrica está sempre sujeito a variações e modificações no cenário de aplicações que podem requerer alterações nos escalonamentos já pré-definidos. Esses tipos de variações são demasiadamente grandes, algumas dessas são de natureza imprevisível e outras já costumam acontecer freqüentemente (Hart et al. 1998).

- Biologicamente o SIB possui mecanismos e características que são robustos quando tratadas variações de cenários e situações de possíveis anomalias. O corpo humano possui uma quantidade limitada de material genético que são capazes de produzirem um número praticamente infinito de receptores celulares, proporcionando um reconhecimento poderoso de antígenos (Hart et al. 1998).

O SIA desenvolve o repertório baseado em Hightower *et al.* (1995) e Hightower *et al.* (1996), onde o material genético usado para formar um anticorpo está armazenado em uma biblioteca, e a partir de um algoritmo genético os componentes da biblioteca são evoluídos com sucesso:

Algoritmo 3.3 – Passos para a geração do escalonamento de Hart et Al. (1998)

- Seleciona um segmento de cada biblioteca randomicamente para formar um anticorpo
- Aplica mutações aleatórias ao anticorpo; e
- Avaliação do escalonamento a partir das instruções presentes no anticorpo aos antígenos dados.

O espaço das soluções é do tipo inteiro onde as moléculas são representadas por permutações dos L elementos que compõem o vetor. O comprimento L_1 dos antígenos é maior ou igual ao comprimento L_2 dos anticorpos, uma vez que estes representam partes de uma seqüência de tarefas (componentes de biblioteca). Os elementos do tipo " * " (*don't care*), no anticorpo, podem ser ligados a qualquer elemento tarefa.

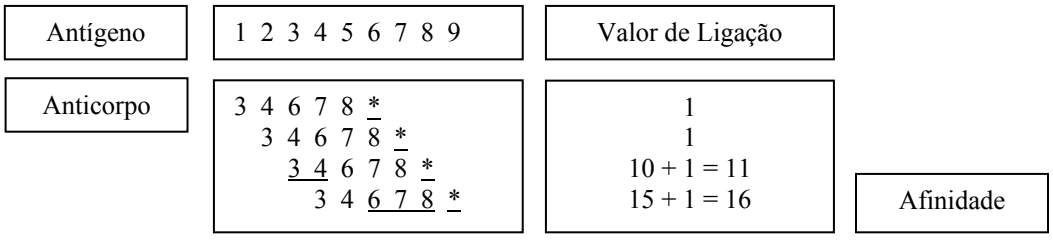


Figura 3.4 - Moléculas codificadas por inteiros, junto com a função de ligação *(matching)*. Correspondências entre os números equivalem a um valor de ligação 5, enquanto o espaço não definido *('don't care')* adiciona 1 a afinidade.

Pode-se observar na Figura 3.4, que a afinidade é calculada a partir de uma função de ligação *(matching)*. Para isso, as duas cadeias são alinhadas verificando a correspondência elemento a elemento. Se as cadeias possuírem comprimentos distintos é calculado um valor de ligação para cada alinhamento possível.

Com o controle da evolução da população, a partir da grande variação de diversidade, os anticorpos produzidos (evoluídos), podem construir o escalonamento original apresentado ao SIA. E por fim, devido a esta alta diversidade, obtiveram-se possibilidades de anticorpos capazes de construírem escalonamentos não expostos previamente a este cenário.

Capítulo 4

Representação da solução do *Job Shop*

Resumo – A representação da solução de um problema de otimização é muito importante e está diretamente relacionada com a definição de operadores de movimento, vizinhança, factibilidade e outros agentes fundamentais para a eficiência da busca. Este capítulo traz um exemplo didático, *Representação por Permutação*, junto à análises de outras duas representações, *Fila de Prioridade e Grafo Disjuntivo*, que serão utilizadas na implementação do Sistema Imunológico Artificial em questão.

4.1 Introdução

Existem na literatura vários tipos de representação (codificação) de solução para o problema de *Job Shop* (Yamada e Nakano, 1997). A escolha de uma representação abrange também a escolha de uma vizinhança apropriada, definição de operadores de movimentos, factibilidade e por fim, avaliação do comportamento de todos estes agentes perante o cenário de aplicação. Tratando-se de algoritmos populacionais, como o SIA, é muito importante que essa codificação permita uma boa exploração do espaço de busca e, principalmente, permita um bom controle de diversidade suportado por métricas bem definidas, eliminação de soluções ambíguas e redundantes.

4.2 Dois exemplos de representação

Os dois exemplos dados a seguir são representações *antagônicas* (o que é vantajoso em uma, é desvantajoso na outra), e foram escolhidos para ilustrar tais características e impactos quando aplicados ao problema. A representação por *Fila de Prioridade* (Gonçalves et al. 2005) tem uma estrutura elaborada não existindo relação direta entre a codificação e o escalonamento. Já a segunda, representação por *Permutação* (Yamada e

Nakano, 1997), é simples e associa-se ao escalonamento facilmente, porém admite soluções infactíveis.

Todos os exemplos discutidos neste capítulo utilizam os dados da instância teste da tabela 4.1.

Tarefa	Ordem de operações por máquina (Tempo de processamento)	
0	0 (3)	1 (2)
1	1 (1)	0 (3)
2	0 (2)	1 (4)

Tabela 4.1 - Instância teste com os processamentos das tarefas

I Representação por Fila de Prioridade (Gonçalves et al. 2005)

A representação por fila de prioridade utiliza um vetor V de inteiros com $J\,x\,M$ posições e valores $v_i \in [\,0\,,J-1\,]$, onde J é o número de tarefas e M é o número de máquinas do problema. A partir deste vetor, um *construtor de solução* lê o sequenciamento dos inteiros e para cada valor *i*, insere a próxima operação da n-ésima tarefa incompleta. A lista de tarefas incompletas mantida pelo construtor é circular, ou seja, se não houver *n-ésima* tarefa incompleta, é escolhido a tarefa seguinte $n+1$, caso $n+1=J-1$ (última tarefa), a próxima tarefa incompleta será a zero.

Dado um exemplo de solução representada pelo vetor gerado aleatoriamente,

[0 1 0 1 1 2],

segue abaixo os passos do *construtor* do escalonamento e em seguida, a montagem construtiva do gráfico de Gantt, Figura 4.1:

Passo 1: $0 \rightarrow$ 1ª tarefa incompleta - tarefa 0 na máquina 0.

Passo 2: $1 \rightarrow$ 2ª tarefa incompleta - tarefa 1 na máquina 1.

Passo 3: $0 \rightarrow$ 1ª tarefa incompleta - tarefa 0 na máquina 1, *fim do tarefa 0*. Agora a tarefa1 é a 1ª e a 2ª tarefa incompleta.

Passo 4: $1 \rightarrow 2^{a}$ tarefa incompleta - tarefa 1 na máquina 0, *fim do tarefa 1*. Agora a tarefa 2 é a 1^{a}, 2^{a} e 3^{a} tarefa incompleta.

Passo 5: $1 \rightarrow 2^{a}$ tarefa incompleta - tarefa 2 na máquina 0.

Passo 6: $2 \rightarrow 3^{a}$ tarefa incompleta - tarefa 2 na máquina 1, *última tarefa incompleta,* fim do escalonamento.

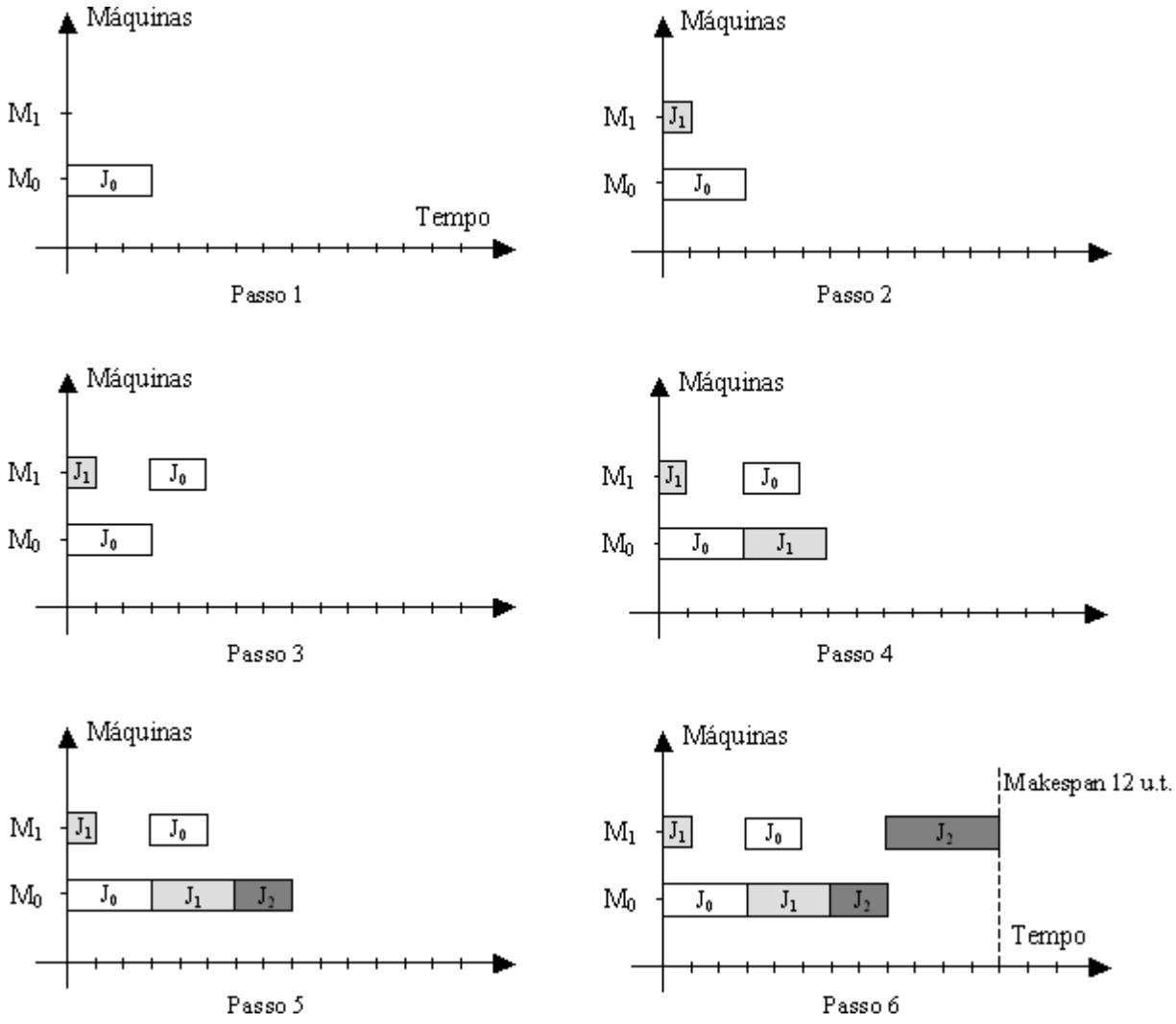


Figura 4.1 - Passos da construção da solução a partir do vetor de fila de prioridade

Esse tipo de representação tem um alto grau de redundância por não haver uma relação direta entre o valor de v_i e uma tarefa no escalonamento, ou seja, dois vetores diferentes podem representar uma mesmo solução, Figura 4.2 (A). Além disso, variações nas seções finais do vetor causam impacto menor na solução que variações no início do vetor. Isso ocorre porque o significado dos valores dos inteiros no final do vetor depende muito dos valores que estão no início do mesmo, Figura 4.2 (B). Por outro lado essa representação não gera soluções infactíveis e qualquer troca dos elementos do vetor também gerará outra solução factível.

No entanto, os operadores de movimento, que a partir de alterações na codificação geram outra solução, devem contornar essas características citadas acima e exemplificadas na Figura 4.2 (A) e (B).

[0 1 0 1 1 2] e [1 0 0 0 2 2]

representam o mesmo escalonamento

(A)

[0 1 0 1 1 2] ⟶ [**2** 1 0 1 1 2] ⟶ causa grande impacto na solução da figura 4.1

[0 1 0 1 1 2] ⟶ [0 1 0 1 1 **0**] ⟶ não causa nenhum impacto na solução da figura 4.1

(B)

Figura 4.2 - Exemplos de soluções por Filas de Prioridade

A representação por *Fila de Prioridade* junto a representação por *Grafo Disjuntivo* descrita na secção 4.3, serão utilizadas para gerar o repertório trabalhado nesta abordagem imuno-inspirada do *Job Shop.*

II Representação por Permutação

O *Job Shop* pode ser relacionado a um problema de ordenação como o problema do Caixeiro Viajante *(Traveling Salesman Problem)*. O escalonamento pode ser representado por um conjunto de permutações das tarefas em cada máquina, em outras palavras, *m*-permutações particionadas (Yamada e Nakano, 1997). A Figura 4.3 mostra essa

representação com a mesma solução da Figura 4.1; na primeira partição do vetor está representada a ordem de processamento das tarefas na máquina M_0 (tarefa0, tarefa1 e tarefa2), na segunda partição os da máquina seguinte e assim sucessivamente.

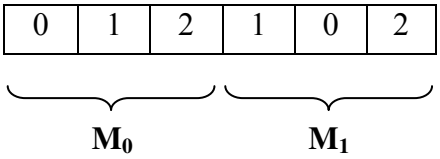


Figura 4.3 - Exemplo de codificação por Permutação

Uma vantagem dessa representação é a transcrição imediata da codificação para a estrutura do escalonamento, ou seja, não é necessária a aplicação de um construtor de solução como a representação por Fila de Prioridade necessita. E, por ser relacionado ao TSP, podem-se usar operadores de movimentos já conhecidos, por exemplo, o *"Subsequence Exchange Crossover (SXX)"* proposto por (Kobayashi e Yamamura, 1995).

Considere as duas soluções *pais* (p1 e p2) representadas pela Figura 4.4. Para as partições equivalentes das mesmas máquinas nas duas soluções é determinado um conjunto de inteiros que serão invertidos gerando dois novos filhos. Em *p1*, são selecionados, por exemplo, 0 e 2, na partição equivalente em *p2,* deve-se identificar os mesmos inteiros, Figura 4.4. Na segunda partição dos dois pais, a seleção de novos inteiros para o intercâmbio é semelhante. Assim serão criados dois novos filhos com trechos de codificação dos dois pais.

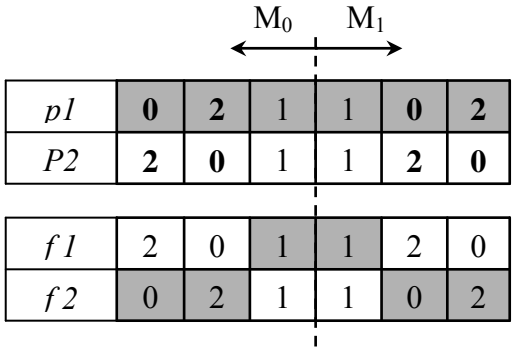


Figura 4.4 - Exemplo de recombinação de soluções

Essa recombinação SXX mantém a estrutura de que cada máquina processe todas as tarefas, ou seja, impede que, após a recombinação uma máquina que antes processava a tarefa0, tarefa1 e tarefa2 passe a processar a tarefa1, tarefa2 e tarefa2 erroneamente. Essa recombinação pode gerar soluções infactíveis, impossíveis de se montar um escalonamento. Para contornar tal situação pode-se aplicar mecanismos de reparo de factibilidade.

4.3 Grafo Disjuntivo

A representação do *Job Shop* por grafo disjuntivo foi proposta por Balas em 1969 no trabalho *"Machine scheduling via disjunctive graph"*. Outras abordagens pós Balas que aplicaram essa representação junto a robustos algoritmos de busca obtiveram resultado eficientes para o problema como Nowicki e Smutnicki (1996) e Scrich (1997).

Considere o exemplo de *Job Shop* apresentado da tabela 4.1, no qual para cada operação tarefa/máquina é criado um nó com peso igual ao seu tempo de processamento, e dois nós artificiais com pesos nulos são criados para representar a operação *inicial* e *final*. Um arco é criado do nó *inicial* ao nó correspondente à primeira operação de cada tarefa; os últimos nós representando as últimas operações da cada tarefa são ligados ao nó artificial *final*; para representar as restrições de precedência entre as operações de uma mesma tarefa, são inseridos os arcos centrais do grafo. Todos estes arcos inseridos até agora são fixos e chamados de arcos Conjuntivos, (Figura 4.5).

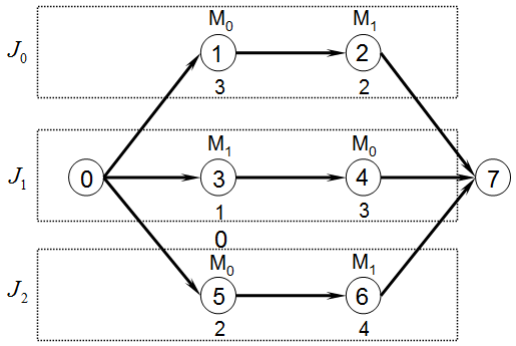


Figura 4.5 - Grafo com os arcos conjuntivos

Para completar, no grafo são inseridos os arcos *Disjuntivos* ligando os nós correspondentes às operações a serem executadas na mesma máquina, estes arcos não têm direções definidas, ou seja, o grafo ainda não representa uma solução do *Job Shop*, Figura 4.6.

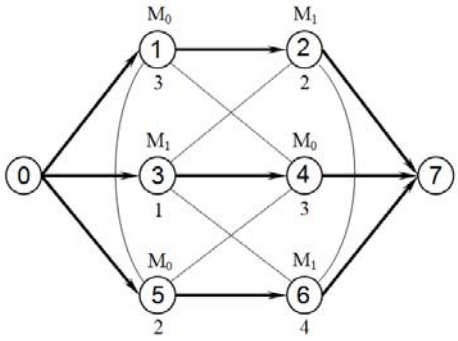
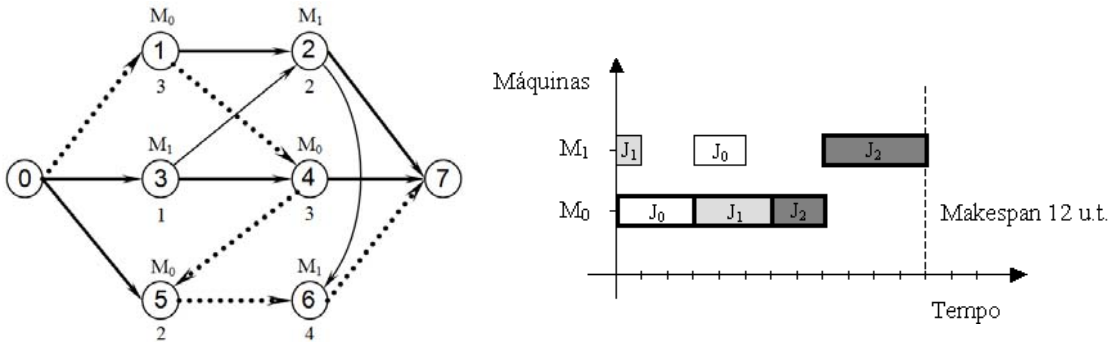


Figura 4.6 - Grafo com os arcos conjuntivos e disjuntivos.

Na Figura 4.6, a tarefa 0 é composta pelas operações 1 e 2 que são processadas nas máquinas M_0 e M_1, respectivamente. As tarefas 1 e 2 são compostas pelas operações 3 a 4 e 5 a 6, respectivamente. Os arcos disjuntivos mostram que as operações 1, 4 e 5 são processadas na máquina M_0 e na máquina M_1 as operações 2, 3 e 6.

Quando as direções dos arcos disjuntivos são escolhidas, obtém-se um grafo direcionado, se o grafo for acíclico, esta configuração representa uma solução factível para o problema. Para determinar o *makespan*, tempo total do processamento, basta calcular o caminho máximo entre o nó inicial e o final do grafo disjuntivo; Dijkstra e Ford-Moore-Bellman são algoritmos clássicos de caminho crítico que podem ser usados. É importante notar que um grafo cíclico não estabelece precedência e, portanto, o caminho mais longo é infinito.



(A) (B)

Figura 4.7 - Grafo direcionado e gráfico de Gantt. Em (A) as setas pontilhadas representam o caminho crítico; (B) o caminho crítico é representado pelos blocos marcados; (A) e (B) representam uma mesma solução de *makespan* 12 para o problema da tabela 4.1.

A Figura 4.7 (A) mostra o grafo direcionado representando a mesma solução da Figura 4.1. O *makespan* referente é calculado a partir do caminho máximo (crítico) passando pelos nós 0-1-4-5-6-7. Em (B), o gráfico de Gantt ilustra o escalonamento obtido por esta solução, onde a máquina 0 processa a seqüência 0, 1 e 2 de tarefas determinado pelo sequenciamento dos arcos (1,4) e (4,5); e a máquina 1 processa as tarefas 1, 0 e 2 nesta ordem determinada pelo sequenciamento dos arcos (3,2) e (2,6). O mapeamento do grafo disjuntivo para o gráfico de Gantt é auxiliado por uma matriz de escalonamento *ME* que guarda as ordens das tarefas em cada máquina.

4.3.1 Movimentação

Primeiramente, são definidas duas propriedades em relação à inversão de arcos do grafo direcionado (Balas, 1969 e Van Laarhoven et al. 1992).

- Lema 4.1: Suponha que (v,w) é um arco disjuntivo pertencente ao caminho critico da tarefa T do grafo acíclico D_i . Seja D_j o grafo direcionado obtido de D_i pela inversão do arco (v,w). Então D_j também a acíclico.

Prova: Suponha que D_j é cíclico. Como D_i é acíclico, o arco (w,v) é parte do ciclo em D_j . Conseqüentemente, existe o caminho (v, x, y, ... , w) em D_j. Mas este caminho também pode ser encontrado em D_i e, claramente, é um caminho maior de v a w que o do arco (v,w) presente no caminho crítico da tarefa T em D_i. Portanto D_j é acíclico. - Logo a inversão de um arco disjuntivo pertencente ao caminho crítico resulta em outra solução factível -

- Esta segunda propriedade pode ser estendida diretamente da primeira e, portanto, segue-se que a inversão de um arco disjuntivo (x,y) do grafo acíclico D_i que não pertence ao caminho crítico, se resultar em um grafo acíclico D_j , não reduzirá o *makespan*, pois os caminhos críticos anteriores ainda continuam em D_j . - Logo se a inversão de um arco disjuntivo que não pertence ao caminho crítico resultar em uma solução factível, então o comprimento do caminho crítico do grafo desta nova solução é maior ou igual ao comprimento do caminho crítico do grafo da solução factível anterior -

Como *makespan* é a tarefa de maior duração, a *movimentação de uma solução* é definida a partir de inversões de arcos disjuntivos pertencentes ao caminho crítico do grafo.

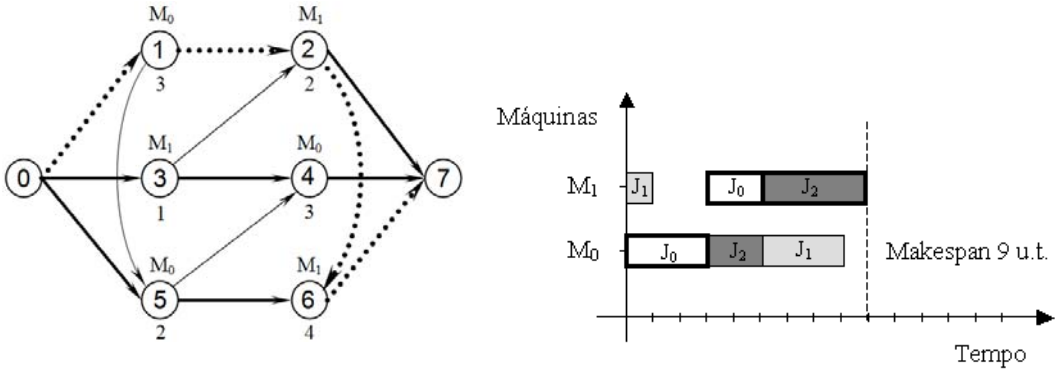
4.3.2 Vizinhança

Para esta abordagem imuno-inspirada do *Job Shop* são apresentados dois tipos de vizinhanças, como descritas a seguir.

I Vizinhança de Balas (1969)

A vizinhança proposta por Balas (1969) consiste em todas as seleções possíveis geradas a partir da inversão do sentido de um dos arcos disjuntivos do caminho crítico. É do tipo 1-opt, ou seja, uma solução D_i é vizinha de D_j se for preciso apenas uma inversão de arco de D_j para chegar a D_i . A partir disso, pode-se determinar qualquer tipo de vizinhança; de maneira abrangente a *n-opt*, se caracteriza a partir de quantos movimentos *n* de trocas de

arcos são definidos. Considerando a Figura 4.7, uma solução vizinha segundo Balas, gerada pela inversão do arco (4,5), pode ser vista na Figura 4.8 abaixo.



(A) (B)

Figura 4.8 - Grafo direcionado e Gantt de uma solução vizinha da Figura 4.7.

Os arcos disjuntivos (1,4) e (4,5) da Figura 4.7 indicam que a ordem das tarefas na máquina 0 é tarefa0- tarefa1- tarefa2. Ao inverter o arco (4,5) para (5,4) implicou em uma nova seqüência na máquina 0, tarefa0- tarefa2- tarefa1, Figura 4.8.

II Vizinhança de Nowicki e Smutnicki (1996)

Nowicki e Smutnicki (1996) apresentaram uma busca tabu baseada em trocas de arcos pertencentes ao caminho crítico do grafo direcionado. Eles utilizaram para gerar a solução inicial um algoritmo baseado em técnicas de inserção desenvolvido por Nawaz et al. (1983) para aplicação em *Flow Shop* e posteriormente refinado para aplicação em *Job Shop* por Warner e Winkler (1992).

A vizinhança utilizada foi baseada na movimentação provocada por trocas de operações das bordas dos blocos construtivos (blocos pertencentes ao caminho crítico). Para um *Job Shop* de 4 máquinas e 7 tarefas, segue um exemplo de gráfico de Gantt ilustrando o caminho critico, os blocos construtivos e os possíveis movimentos de trocas para esta situação corrente, Figura 4.9. Se for realizada uma troca por cada movimento, observa-se que esta vizinhança é uma especialização da Vizinhança de Balas.

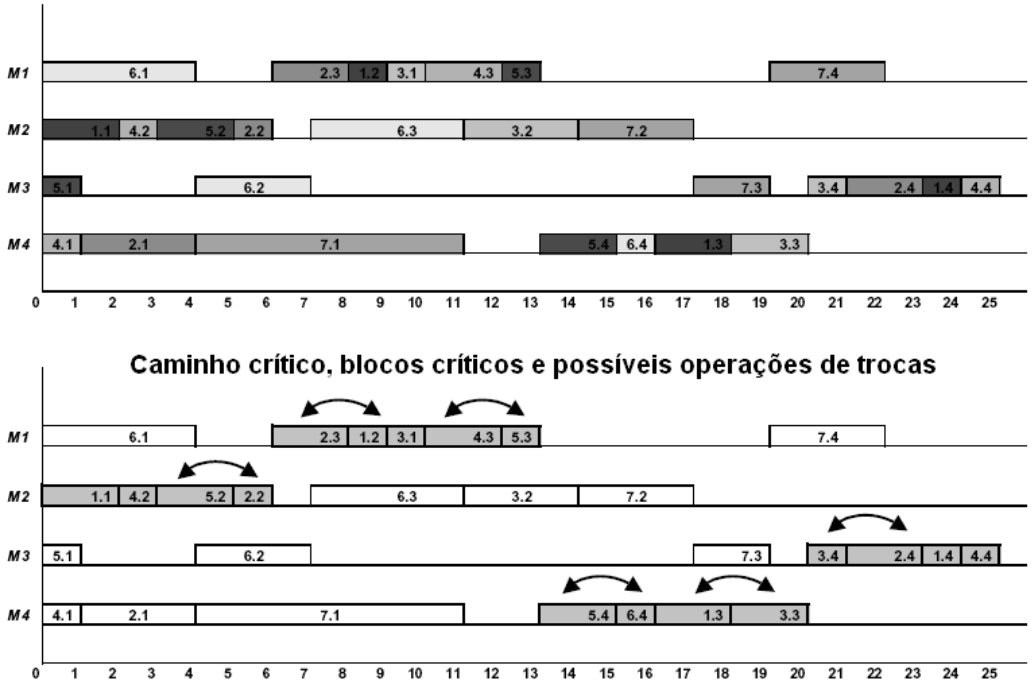
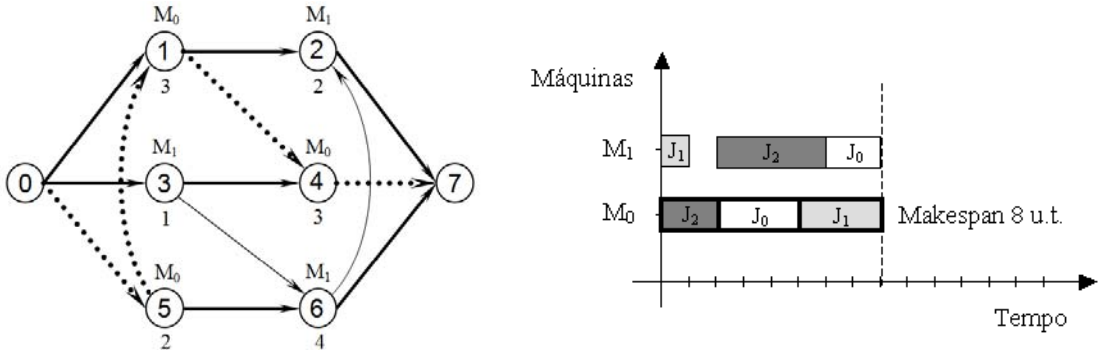


Figura 4.9 - Vizinhança de Nowicki e Smutnicki (1996). Fonte: Gonçalves et al. (2005).

Dada a Figura 4.9, com exceção da primeira (1.1) e da última (4.4) operações do caminho crítico, as demais inversões de bordas (inversão dos arcos que as representam), são permitidas como movimentos.

Esta vizinhança é significativamente reduzida, pois os movimentos que não melhoram imediatamente o *makespan* são removidos (Scrich, 1997), isto é, pode-se mostrar que a troca de determinadas operações (tarefa/máquina) de um bloco construtivo resulta em uma solução com o *makespan* menor ou igual a da anterior. Para exemplificar esta característica, considere o grafo da Figura 4.8. Naquela configuração não existe nenhum movimento possível segundo a vizinhança de Nowicki e Smutnicki (1996), mas se for invertido o arco (2,6) e em seguida o arco (1,5), será obtido uma solução de menor *makespan*, Figura 4.10.

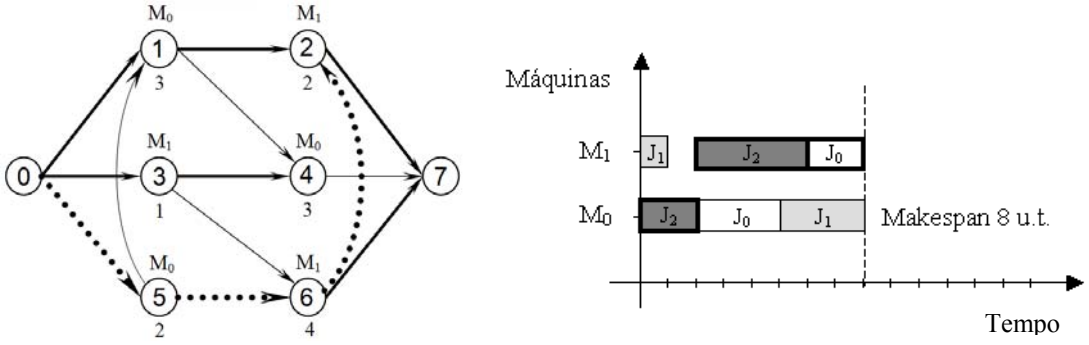
Como a solução da Figura 4.10 não tem atrasos em uma das máquinas (a máquina M_0 funciona ininterruptamente durante todo o processo), esta é uma solução ótima para o problema descrito na tabela 4.1.



(A) (B)

Figura 4.10 - Exemplo onde a vizinhança de Nowicki e Smutnicki limitou a busca

Também é importante observar que um grafo pode ter caminhos críticos diferentes representando a mesma solução e *makespan*. Considere a Figura 4.11 abaixo, o seu gráfico de Gantt representa a mesma solução e *makespan* da Figura 4.10, porém os caminhos críticos são diferentes.



(A) (B)

Figura 4.11 - Grafo Disjuntivo e Gráfico de Gantt representando a mesma solução da Figura 4.10, porém com caminho crítico diferente

Fica claro então que para descobrir se duas soluções são diferentes, não basta apenas analisar seus *caminhos críticos*, mas verificar a configuração de *arcos disjuntivos* que as compõem.

Capítulo 5

Abordagem Imuno-Inspirada

Resumo – Este capítulo traz a descrição detalhada do Sistema Imunológico Artificial implementado. Os tópicos a seguir são embasados nas abordagens imuno-inspiradas vistas até agora e são apresentados seguindo o fluxo de execução do algoritmo: criação do repertório inicial, seleção clonal, maturação, seleção e tratamento de diversidade.

5.1 Introdução

Conforme descrito no Capítulo 2, o sistema imunológico biológico é complexo e possui grande variedade de componentes e funcionalidades, onde a interação de moléculas, células e organismos proporcionam um meio de defesa muito eficiente (De Castro, 2001). Para o desenvolvimento de um Sistema Imunológico Artificial é necessária uma simplificação da realidade biológica de maneira a enquadrar os conceitos, julgados mais relevantes, para a aplicação no *Job Shop*. O algoritmo proposto aqui é inspirado nos conceitos apresentados no Capítulo 2 junto às considerações de *representação, vizinhança e movimentação* vistas no Capítulo 4.

De maneira resumida, o *repertório* do SIA é formado por *anticorpos* (grafos disjuntivos) que geram novos indivíduos herdeiros (*clones*) com as mesmas características dos anticorpos-pai. Uma *medida de afinidade* é utilizada de forma a privilegiar os anticorpos mais adaptados ao ambiente (seleção clonal), ou seja, somente aqueles anticorpos que representam soluções de baixo custo pré-estabelecido serão ativados para se proliferar. Os resultados da proliferação são novos anticorpos refinados que serão mantidos no repertório se houver afinidade superior que seus respectivos pais.

5.2 Representação da solução

Este Sistema Imunológico Artificial utiliza a representação de Grafo Disjuntivo (Balas, 1969). Na criação do repertório aleatório inicial é usada também a representação de Fila de Prioridade, auxiliando a construção das soluções bases para o mapeamento em grafos disjuntivos. O processo de construção de uma solução factível é descrito em seguida.

5.3 Inicialização do repertório

Em um SIA é muito importante que o procedimento de inicialização do repertório permita um bom espalhamento dos anticorpos pela superfície de busca (Freitas Filho et al. 2007). Primeiramente, utiliza-se a Fila de Prioridade, isso torna o processo de inicialização simples e eficiente, pois uma solução factível surge de números aleatórios que compõe o vetor de prioridades. Com uma solução factível em mãos, converte-se facilmente em Grafo Disjuntivo.

Seja uma solução da instância da tabela 4.1 dada pelo vetor [0 1 0 1 1 2]; o *construtor de solução* (vide secção 4.1, Representação por Fila de Prioridade, Figura 4.1) lê este vetor e o transforma em uma *Matriz de Escalonamento*,

$$ME = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 2 \end{pmatrix},$$

onde as linhas representam a ordenação das tarefas em cada máquina tal como é mostrado no gráfico de Gantt.

Dada a *MP*, a formação dos arcos disjuntivos é imediata. Para representar a ordenação das tarefas na máquina 0, insere-se os arcos (1,4) e (4,5) no grafo da Figura 4.5, analogamente os arcos (3,2) e (2,6) para máquina 1. Então está formada a solução do *Job Shop* por Grafo Disjuntivo, que para este exemplo, está representado pela Figura 4.7 do capítulo anterior.

O repertório inicial *(A)* é formado por soluções desse tipo, grafos disjuntivos diversificados e factíveis.

5.4 Medida de afinidade

Seja $\widetilde{A}$ o escalonamento que possui o *menor tempo de processamento* conhecido da instância avaliada, ou seja, o menor *makespan* encontrado na literatura. É definida uma medida que avalia a qualidade de um anticorpo A_i em relação aos outros do repertório:

$$Afinidade\ (A_i) = \begin{cases} \dfrac{makespan\ (\widetilde{A})}{makespan\ (A_i)} & , \quad se \quad makespan\ (A_i) \geq makespan\ (\widetilde{A}) \\ 1 & , \quad se \quad makespan\ (A_i) < makespan\ (\widetilde{A}) \end{cases} \tag{6}$$

A $Afinidade\ (A_i) \in [0,1]$ e quanto mais próximo o *makespan* de A_i estiver do *makespan* de $\widetilde{A}$, maior será a sua afinidade. Dessa forma, fica definido como *antígeno* o *makespan* da melhor solução encontrada na literatura, $\widetilde{A}$.

Uma vez detectando-se os anticorpos de alta afinidade, estes sofrerão um processo de clonagem. O seu material genético (codificação) é replicado em outros novos anticorpos do sistema.

5.5 Clonagem ou Seleção Clonal

O processo de clonagem (De Castro, 2001) trata-se de uma geração de cópias idênticas dos anticorpos selecionados, baseando-se nos critérios imuno-inspirados abaixo:

- anticorpo de alta afinidade terá uma clonagem intensificada, pois é interessante replicar e refinar esta codificação de boa qualidade;
- para anticorpo com afinidade não é tão expressiva, porém ainda com qualidade, o processo de clonagem é menos intenso do que o anterior;
- por fim, anticorpo de baixa afinidade não é selecionado para a clonagem e será eliminado do repertório.

Primeiramente, são selecionados para clonagem anticorpos com $Afinidade \geq \lambda$, para o limiar $\lambda \in (0,1)$. Os demais são substituídos no repertório por outros gerados aleatoriamente.

Para os anticorpos selecionados, considere:

$n_c(A_i)$ = *número de clones idênticos gerados a partir do anticorpo pai* A_i.

$c_{\min}$ = *número mínimo de clones.*

$c_{\max}$ = *número máximo de clones.*

A equação abaixo é baseada na parametrização da reta da Figura 5.1, onde a clonagem de um anticorpo leva em consideração sua afinidade:

$$n_c(A_i) = \left\lfloor c_{\max} + \left(\frac{c_{\max} - c_{\min}}{1-\lambda} \right) (Afinidade(A_i) - 1) \right\rfloor . \quad (7)$$

Dessa forma,

$n_c(A_i) \in [c_{\min}, c_{\max}]$, onde $c_{\min}$, $c_{\max}$ *e* $n_c(A_i) \in \mathbb{N}$.

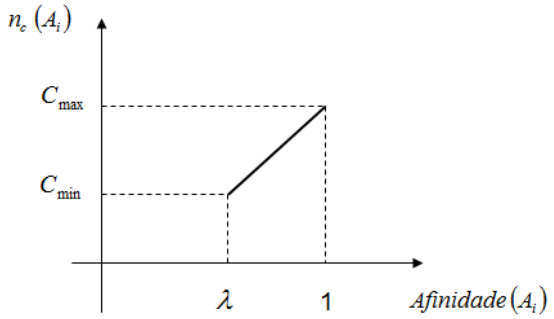


Figura 5.1 - Variação do número de clones por anticorpos em função da afinidade

A criação de clones *(C)* feita dessa maneira, junto à movimentação definida em 5.6.1, intensifica a capacidade de exploração do algoritmo.

5.6 Maturação de Afinidade (mutação)

Os clones são diferenciados e refinados pelo processo de Maturação de Afinidade, gerando a partir de *C* um novo grupo de clones *(C')*, com o intuito de aumentar a qualidade do reconhecimento em relação ao antígeno. Este processo do sistema imunológico artificial deve ser estruturado de maneira a varrer localmente uma solução selecionada para clonagem. Ou seja, os mecanismos de movimentação e de vizinhança devem ser robustos e apropriados para o problema em questão.

5.6.1 Definição do movimento

A maturação a ser realizada é baseada no tipo de movimento que a solução clone sofrerá, estes movimentos são definidos de 1-*opt* até *β*-*opt*. O movimento de 1-*opt* significa uma inversão de arco disjuntivo pertencente ao caminho crítico do grafo, logo o *β*-*opt* são *β* movimentos.

Também é interessante que a mutação seja inversamente proporcional à afinidade do anticorpo pai, *Ai:*

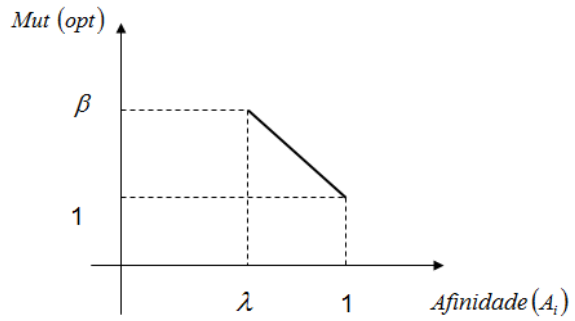


Figura 5.2 - Tipo de mutação sofrida pelo clone de acordo com a afinidade do pai.

Dessa maneira, cada anticorpo terá uma mutação diferenciada de acordo com sua especificidade, caracterizada pela quantidade de movimentos. Este número é calculado baseado no gráfico da Figura 5.2:

$$Mut(opt) = \left\lfloor 1 + \left(\frac{1-\beta}{1-\lambda} \right) \left(Afinidade\left(A_i\right) - 1 \right) \right\rfloor, \tag{8}$$

onde $\beta \in \mathsf{N}$ é número máximo de inversões de arcos, *β-opt.*

A medida de afinidade é utilizada para definir o **número de clones gerados por anticorpo** e o **tipo de movimento**. De acordo com o que foi apresentado em 5.5 e 5.6.1, anticorpos de alta afinidade geram muitos clones com poucas maturações, pois neste caso, o interesse é *intensificar* a busca localmente. Já anticorpos de baixa afinidade geram poucos clones com muitas maturações, pois um número elevado de maturações permite maior *diversificação* das soluções, promovendo assim uma melhor exploração do espaço de busca. Este é um aspecto inerente às abordagens imuno-inspiradas.

5.6.2 Definição da Vizinhança

Balas (1969) e Nowicki & Smutnicki (1996) são utilizados no algoritmo em momentos *apropriados*, onde a característica de cada anticorpo pai (afinidade) induzirá uma escolha adaptada da vizinhança para a clonagem resultando em anticorpos filhos.

Como visto no capítulo 4, a vizinhança de Nowicki & Smutnicki é limitada pois não permite movimentos que piorem a solução. É interessante sua utilização quando há necessidades de refinamento em boas soluções onde o tipo de movimento selecionado pelo gráfico da Figura 5.2 é de 1-*opt*, por exemplo. O critério que seleciona o tipo de vizinhança é baseado no Método da Roleta *(Roullete Wheel)*, quanto maior a afinidade do anticorpo pai *(Ai)*, maior é a probabilidade que seus clones (C*i)* sofram maturação segundo esta vizinhança.

Com a definição da **vizinhança** a partir da afinidade junto ao método da roleta, insere-se um grau adaptativo na busca, pois soluções que sofrem poucos ou 1 único movimento de mutação possuem grande probabilidade de se executar na vizinhança de Nowicki e Smutnicki (1996). Por fim, a roleta tenderá a escolher Balas para anticorpos com afinidades não tão expressivas, onde mutações múltiplas proporcionam maior mobilidade da solução em busca de regiões promissoras.

5.7 Mecanismo de Seleção

O processo de seleção dos clones maturados é inspirado nos princípios da Seleção Clonal (De Castro e Von Zuben, 2001). Portanto, o método utilizado é elitista (Michalewicz, 1996), pois privilegia o clone com maior grau de afinidade *(c*)*. Assim, o clone mais adaptado (solução de menor *makespan*) é selecionado do repertório de clones e comparado com o seu anticorpo-pai e apenas o melhor anticorpo é preservado no repertório.

5.8 Memória

O uso da memória é interessante quando se tem um volume de informações correntes que não é interessante "carregar" durante todas as interações, mas que são importantes e de alguma forma devem ser mantidas e armazenadas. Essa aplicação aumenta o desempenho do algoritmo diminuindo assim o esforço computacional.

No contexto em aplicação, a memória também auxilia os mecanismos de controle de diversidade. Todo anticorpo com boa afinidade é preservado na memória antes de ser eliminado do repertório ou de ter sua codificação alterada.

5.9 Diversidade

Diversidade elevada é uma forte característica do Sistema Imunológico Natural (De Castro e Von Zuben, 2001). A inicialização aleatória do repertório está dentro deste requisito. Além disso, é muito importante a existência de mecanismos que controlem e mantenham essa característica, nesta aplicação foram desenvolvidas *Supressão* e *Tratamento a Ótimos-locais.*

5.9.1 Supressão

O controle da diversidade durante as iterações do algoritmo é feito utilizando uma métrica baseada no número de arcos disjuntivos distintos entre dois grafos. Vale lembrar que os arcos conjuntivos são fixos e iguais para todas as soluções.

Seja J e M os números de tarefas e máquinas respectivamente, o número total de arcos do grafo é

$$(M+1)J + (J-1)M\ , \tag{9}$$

onde o primeiro bloco da soma desta expressão faz referência aos arcos conjuntivos e o segundo aos arcos disjuntivos (Balas, 1969). Para achar a quantidade de arcos disjuntivos distintos entre duas soluções, basta subtrair do total $D_{\max} = (J-1)M$, o número de arcos disjuntivos compartilhados entre elas. A distância avaliada por esta métrica é simétrica e definida como segue a expressão:

$$D(A,B) = D_{\max} - n_{arc}\ ,\ n_{arc}\ \textit{é o nº de arcos disjuntivo compartilhados entre A e B} \tag{10}$$

Dado o grafo (A) e (B) da Figura 5.3, eles possuem em comum os arcos (3,2) e (2,6). Neste exemplo a distância máxima entre dois grafos é $D_{\max} = (J-1)M = 4$ e a distância entre A e B é $D(A,B) = 4-2 = 2$, logo o grau de diversidade entre estas duas soluções é 50%. Dessa maneira é possível identificar quando o repertório está perdendo diversidade, ou seja, quando os grafos estiverem topologicamente próximos uns dos outros.

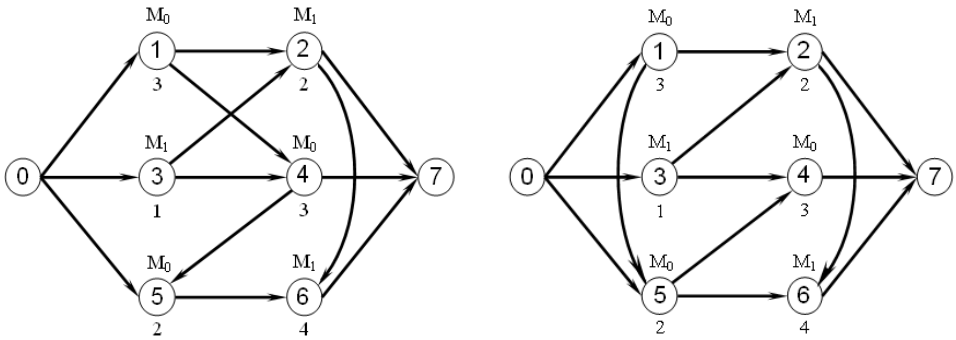
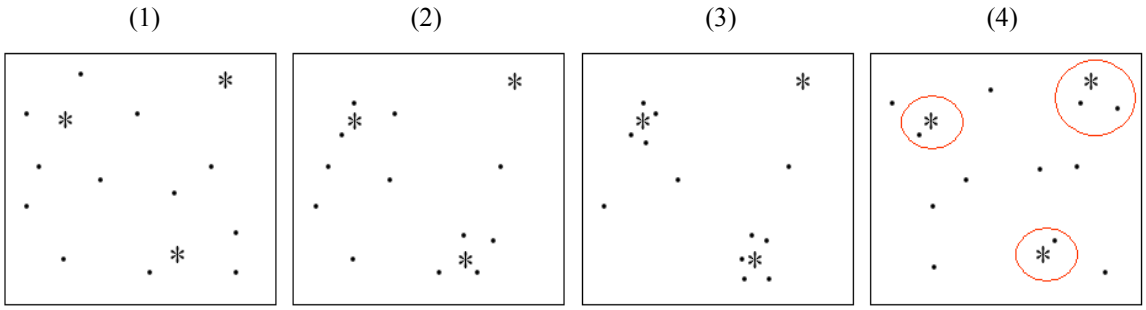


Figura 5.3 - Estrutura da métrica, onde as soluções (A) e (B) estão distantes entre si por 2 und de distância.

Essa métrica deu subsídio à criação de um mecanismo de *Supressão* para evitar que os anticorpos se concentrem em regiões promissoras.

A Figura 5.4 ilustra a evolução de um repertório no tempo. Inicialmente, os anticorpos estão espalhados na superfície de busca, com o passar das iterações, a tendência é que eles se concentrem em regiões promissoras. Porém, não é necessária a presença de muitos deles nestas regiões, pois a capacidade de *explotação* já é alta devido ao processo de clonagem de um único anticorpo.



* regiões promissoras do espaço de busca • Anticorpos

Figura 5.4 - Ilustração do mecanismo de seleção

Para o reconhecimento dos anticorpos a serem suprimidos, pode-se definir uma distância mínima que eles devem ter entre si. Assim, apenas um anticorpo que identifica uma determinada região em potencial é mantido enquanto os demais são armazenados na Memória e substituídos por outros gerados aleatoriamente no repertório principal. No entanto o custo computacional da verificação de diversidade é alto, já que cada anticorpo do repertório deve ser comparado com os outros; sendo assim, a S*upressão* deve ser aplicada periodicamente nos anticorpos resultantes da clonagem. Nos demais anticorpos não é necessário, pois estes são eliminados devido ao critério elitista da *Seleção Clonal*. A supressão ainda permite que novas regiões sejam exploradas a partir da inserção destes novos anticorpos aleatórios, Figura 5.4 quadro (4).

Outro mecanismo proposto é a substituição de anticorpos de baixa afinidade, não selecionados para clonagem, por outros aleatórios. A substituição está de acordo com o conceito de *Repertório Disponível*, anticorpos que não estão ativos no momento, mas

podem vir a ser utilizados (De Castro e Von Zuben, 2001). Este processo pode ser entendido como uma *exploração* no espaço de busca; já o princípio de seleção clonal permite uma busca local em diferentes direções, a *explotação* (Coello e Cortés, 2005).

5.9.2 Tratamento a ótimos-locais

Além de existir a tendência dos anticorpos se concentrarem em regiões promissoras, é muito comum que um anticorpo caia em uma região de ótimo local. Nesta situação a clonagem nesta vizinhança não trará melhorias e este anticorpo permanece inutilizado e ocorrerão operações desnecessárias até a finalização do algoritmo.

Para contornar tal situação, são definidos dois critérios de *Tratamento a ótimos-locais* (fuga de ótimos-locais). Quando um anticorpo com *afinidade* $\geq \lambda$ atinge um número máximo de iterações sem melhorias, os critérios de tratamento são ativados:

- Eliminação: o anticorpo em questão é copiado na Memória, eliminado do repertório e substituído por outro aleatoriamente.
- Perturbação Proporcional: o anticorpo em questão é copiado na Memória, no repertório ele é alterado por uma perturbação do tipo *k-opt* com inversões aleatórias dos arcos disjuntivos pertencentes ao caminho crítico do grafo. A parametrização de *k* deve ser suficiente para permitir o "escape" do anticorpo,

$$k = \beta + \lfloor \alpha . \beta \rfloor, \quad \alpha \in \Re^{+} . \tag{11}$$

Assim é possível que soluções presas em ótimos-locais tenham **mobilidade** e possam escapar em busca de novas regiões promissoras.

Na Figura 5.5 pode ser observado um exemplo ilustrativo desse mecanismo. Em (A) é representada a eliminação do anticorpo *Ai* na posição P_1, seguido da criação de um aleatoriamente *Ai'* em P_2. Em (B) é mostrada a perturbação no anticorpo fazendo-o migrar do ponto P_1 para um não tão distante P_3 com grande tendência a atingir o mínimo global da função nas próximas iterações.

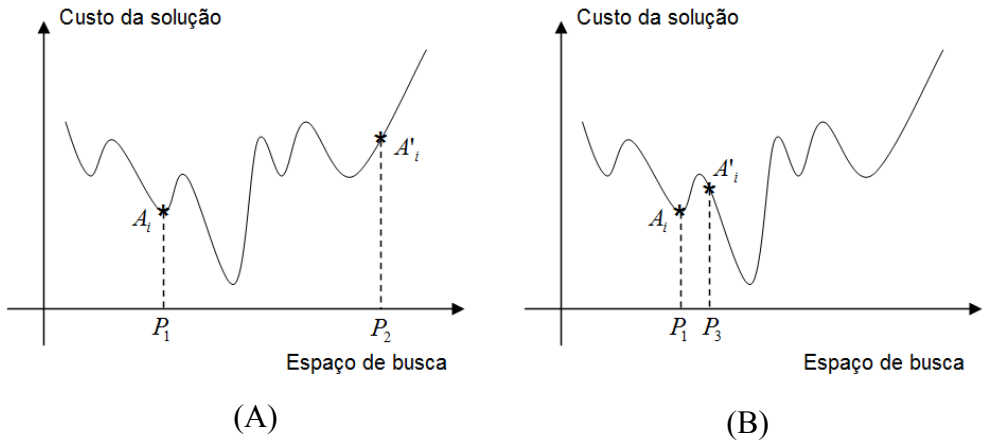


Figura 5.5 - Ilustração do mecanismo de Tratamento a ótimos-locais. (A) representa a eliminação do anticorpo Ai na posição P1, seguido da criação de um aleatoriamente Ai' em P2; (B) mostra a perturbação no anticorpo fazendo-o migrar do ponto P1 para P3.

Quando uma solução tem boa afinidade, é interessante aplicar a *Perturbação Proporcional*, assim não é perdida toda a maturação feita até então. Se esta solução após alguns ciclos de aplicação da *Perturbação Proporcional* persistir em retornar ao mesmo ótimo-local, então torna-se fundamental a aplicação da *Eliminação* ou um aumento do parâmetro α.

5.10 Pseudocódigo do Sistema Imunológico Artificial

O pseudocódigo a seguir mostra como está estruturado o algoritmo para a resolução do problema

Dados de entrada: Instância em arquivo *(.txt)* contendo as tarefas, máquinas, ordem de processamento e tempos de execução.

Algoritmo 5.1 – Sistema Imunológico Artificial Implementado

```
A ← Inicialização o repertório com n anticorpos (grafos disjuntivos)
Afinidade ← Cálculo da afinidade do repertório A
geração ← 1
enquanto critério de parada não atingido faça
    para cada anticorpo A_i faça
        se Afinidade (A_i) ≥ λ então
            Seleção do anticorpo A_i para clonagem
                Cálculo do n_c(A_i)
                C ← n_c(A_i) clones de A_i
                Definição da vizinhança da mutação por roleta
                Definição do tipo do movimento, calculando Mut(opt)
                C' ← Clones C maturados de acordo com a vizinhança e mutação
                f_c ← Cálculo da Afinidade dos clones C'
                c* ← Seleção do melhor clone de C'
                se Afinidade(c*) ≥ Afinidade(A_i) então
                    A_i ← c*
                fim - se
        fim - se
        se Afinidade (A_i) < λ então
            Substituir os anticorpos A_i não-selecionados por outros aleatórios
        fim - se
        Atualização da Afinidade
    fim – para

    Verificação periódica da diversidade, função[Supressão]{
        função[Memória]
    }
    Verificação de mobilidade, função[Tratamento a ótimos-locais]{
        função[Memória]
    }
    geração ← geração+ 1
fim – enquanto
```

Dados de Saída: Grafos Disjuntivos representando soluções diversificadas e de baixo custo.

Capítulo 6

Experimentos e Resultados

Resumo – Este capítulo apresenta os resultados computacionais provenientes da aplicação do Sistema Imunológico Artificial proposto no Capítulo 5 a diversas instâncias clássicas do *Job Shop*. Juntamente com a aplicação, também são feitas análises de custos e diversidade dos conjuntos-soluções obtidos.

6.1 Introdução

Este trabalho não tem só como proposta encontrar a solução de menor *makespan* de uma dada instância, mas também encontrar um grupo com soluções diversificadas de baixos *makespans* associados. Visto tal objetivo, é justificável a escolha de uma Meta-Heurística populacional onde características inerentes de manutenção de diversidade tornam-na muito eficiente para o propósito em questão. Sendo assim, foi desenvolvido um Sistema Imunológico Artificial nos moldes propostos por Dasgupta (1998) e De Castro (2001) para encontrar um *conjunto-solução* diversificado tentando contornar tal complexidade.

Para avaliar a eficiência do algoritmo implementado, foram feitas aplicações em instâncias clássicas da literatura. Ele foi executado 30 vezes por instância, sendo apresentados aqui os resultados da melhor rodada por instância, ou seja, melhor conjunto-solução dadas as variações de parâmetros realizadas por rodada.

Foi criada uma medida de comparação que leva em consideração os *makespans* e o grau de diversidade, possibilitando assim mensurar qual o melhor conjunto-solução de todas as rodadas.

6.2 Avaliação de grupos de soluções

A principal meta do sistema proposto é encontrar um grupo com *k* soluções diversificadas com baixo custo associado. Então, é definida uma medida que avalie tais requisitos (Avaliação de Qualidade do Grupo de Soluções *K*):

$$AQGS_K = \delta SL^* + \delta D,$$

onde δSL^* é o desvio da média dos makespans das *k* soluções em relação ao menor já encontrado na literatura e δD é o desvio da média das distância relativas entre cada par de solução de *K* em relação à distância máxima definida pela métrica na secção 5.9.1.

Considere as soluções $S_1 =$ Figura 4.7, $S_2 =$ Figura 4.8 e $S_3 =$ Figura 4.10 com respectivos makespans (12, 9, 8). Para representar as distâncias relativas entra cada solução é utilizada a Matriz de Distâncias Relativa,

$$MDR = \begin{array}{c|ccc} & S_1 & S_2 & S_3 \\ \hline S_1 & 0 & 2 & 3 \\ S_2 & 2 & 0 & 4 \\ S_3 & 3 & 4 & 0 \end{array} .$$

MDR possui simetria devido à definição da métrica utilizada e ainda vale ressaltar que sua diagonal principal é nula, pois a distância ente uma solução e ela mesma é 0. Tendo essa informação em mãos, o cálculo da média das distâncias deve apenas considerar uma porção triangular desta matriz, visto que a outra parte é replicação das distâncias anteriores.

Os desvios para este exemplo *(K=3)* são:

$$\delta\overline{SL}^* = \frac{9{,}66 - 8}{8} = 0{,}208 \text{ , onde } \overline{S} = \frac{12+9+8}{3} = 9{,}66$$

$$\delta D = \frac{4-3}{4} = 0{,}25 \text{ , onde } \overline{D} = \frac{2+3+4}{3} = 3 \text{ ;}$$

e a avaliação do conjunto *K* é dada por $AQGS_K = 0{,}458$

Segundo as definições acima, quanto menor for $AQGS_K$, maior é a qualidade do grupo quando considerados custo e diversidade. Obviamente é possível ponderar a importância destes requisitos atribuindo multiplicadores nos desvios, abordagem não focada nesta aplicação.

6.3 Plataforma

O algoritmo central e suas funções foram implementados em linguagem de programação C no ambiente de desenvolvimento *Bloodsheld Dev-C++* versão 4.9.9.2. Todos os testes foram realizados no sistema operacional *Windows Vista* de um PC com processador *AMD Turion*^{X2}, 1.6 GHz e 1 GB de memória RAM. Por fim, para análises estatísticas e apresentações gráficas foi utilizada a ferramenta *Excel* 2003.

6.4 Instâncias testes

Para aplicação neste trabalho, foram escolhidas 14 instâncias bastante conhecidas de diversos autores:

- *ft,* retiradas de Fisher e Thompson (1963);
- *abz,* retiradas de Adams et al. (*1988*);
- *la*, Lawrence (*1984*); e
- *orb*, Applegate e Cook (*1991*).

assim foi possível avaliar o comportamento do algoritmo em diferentes classes de testes.

Como dito anteriormente, o foco da busca é encontrar um conjunto-solução de baixo custo e alta diversidade. Logo, dada uma instância de um determinado autor, serão apresentadas as soluções do conjunto *K* e sua Matriz de Distância Relativa (*MDR*).

Vale ressaltar que os parâmetros e resultados apresentados aqui são referentes ao conjunto-solução *(K=10)* da melhor rodada de cada instância. Este valor *K* igual a *10* são as

melhores soluções encontrada no algoritmo desde as soluções corrente até as que estão em memória.

A tabela 6.1 traz comparações dos parâmetros utilizados nos testes. Aqui é possível observar os campos *número de anticorpos*, *número de iterações*, λ, β e $(c_{\min}, c_{\max})$, campos de extrema importância para a eficiência do algoritmo. Tais parâmetros devem ser devidamente calibrados para cada instância, levando em consideração sua complexidade de busca. Caso contrário, os resultados obtidos não serão satisfatórios.

Instância	***Nº de anticorpos***	***Nº de iterações***	λ	β	$(c_{\min}, c_{\max})$
ft06	60	15	0,7333	4	(5,15)
ft10	200	450	0,689	13	(10,45)
abz05	200	300	0,7389	10	(20,45)
abz06	200	400	0,7143	15	(20,45)
la01	70	17	0,7483	7	(8,18)
la06	50	17	0,7123	6	(8,18)
la10	50	30	0,7983	6	(8,18)
la13	150	200	0,82	6	(10,40)
la15	120	411	0,8155	9	(15,30)
la19	150	180	0,7165	10	(20,45)
orb02	570	760	0,7161	13	(10,45)
orb05	320	450	0,7062	11	(10,45)
orb07	580	690	0,693	13	(10,45)
orb10	660	750	0,684	13	(10,45)

Tabela 6.1 - Tabela de comparação dos parâmetros utilizados na melhor rodada por instância.

A seguir serão apresentados os resultados das instâncias separadas pelos autores que as criaram. Cada bloco de autor traz:

- uma tabela com as *instâncias testadas, melhor resultado conhecido na literatura (L^*)* e *conjunto-solução* vinculado ao *tempo de execução*; e
- uma tabela com as matrizes de distâncias relativas de cada par de soluções.

Na secção 6.5 será feita a análise dos resultados junto a estudos estatísticos e comparativos aos melhores resultados conhecidos na literatura.

6.4.1 Fisher e Thompson (1963)

O trabalho de Fisher e Thompson (1963) foi um dos pioneiros em Aprendizagem Probabilística na resolução do *Job Shop*. Também serviu de estímulo para outros trabalhos, pois foi constatado na época que combinações randômicas de regras de despacho têm mais efeito positivo que tomá-las separadamente, e, aprendizagem era possível.

A tabela 6.2 mostra os resultados do Sistema Imunológico Artificial quando aplicado em instâncias de Fisher e Thompson (1963).

Instância	L^*	***Conjunto-solução (K=10) encontrado pelo SIA***	***Tempo(s)***
ft06	55	55 55 58 58 59 59 59 60 60 60	0,752
ft10	930	930 948 955 958 959 960 961 962 964 965	732,21

Tabela 6.2 - Tabela de apresentação das k-soluções para instâncias do tipo *ft.*

A diversidade das soluções de *K* pode ser observada na tabela 6.3.

Instância	***Instância***
0 8 20 19 21 24 19 22 23 19 8 0 17 15 24 25 20 20 25 17 20 17 0 21 21 23 20 24 22 22 19 15 21 0 20 23 21 13 18 11 21 24 21 20 0 11 27 18 14 22 24 25 23 23 11 0 29 24 17 22 19 20 20 21 27 29 0 19 23 25 22 20 24 13 18 24 19 0 22 16 23 25 22 18 14 17 23 22 0 22 19 17 22 11 22 22 25 16 22 0 ft06	0 57 56 55 55 73 51 44 55 62 57 0 35 6 64 68 63 49 65 62 56 35 0 36 70 58 64 53 69 65 55 6 36 0 67 66 64 50 62 57 55 64 70 67 0 80 59 54 61 76 73 68 58 66 80 0 81 71 82 58 51 63 64 64 59 81 0 53 24 74 44 49 53 50 54 71 53 0 59 68 55 65 69 62 61 82 24 59 0 72 62 62 65 57 76 58 74 68 72 0 ft10

Tabela 6.3 - Matrizes de distâncias relativas para instâncias do tipo *ft.* A distância máxima entre cada par de solução é 30 para a instância *ft06* e 90 para *ft10*.

6.4.2 Adams et al. (1988)

Adams et al. (1998) propuseram um método de aproximação para a resolução do *Job Shop* baseando-se no sequenciamento de máquinas mais carregadas. A cada sequenciamento de uma máquina é realizada uma reotimização local por procedimentos de gargalos *(Shifting Bottleneeck Procedure)*. Juntamente com este procedimento os autores apresentaram as instâncias *abz* para avaliação.

A tabela 6.4 traz os resultados do SIA quando aplicado em instâncias de Adams et al. (1988).

Instância	L^*	***Conjunto-solução (K=10) encontrado pelo SIA***	***Tempo(s)***
abz05	1234	1234 1251 1255 1256 1258 1261 1264 1264 1265 1267	580,06
abz06	943	945 948 950 952 962 962 962 962 962 966	393,885

Tabela 6.4 - Tabela de apresentação das k-soluções para instâncias do tipo *abz.*

A diversidade das soluções pode ser vista na tabela 6.5.

Instância	***Instância***
0 64 65 62 58 30 66 68 72 62 64 0 36 24 30 70 51 73 67 59 65 36 0 39 40 65 44 65 63 55 62 24 39 0 16 72 45 74 66 62 58 30 40 16 0 67 41 69 65 57 30 70 65 72 67 0 71 68 63 65 66 51 44 45 41 71 0 52 49 47 68 73 65 74 69 68 52 0 41 45 72 67 63 66 65 63 49 41 0 36 62 59 55 62 57 65 47 45 36 0 abz05	0 39 34 62 62 64 59 61 59 71 39 0 41 63 68 51 67 61 60 59 34 41 0 64 64 55 64 62 62 68 62 63 64 0 38 60 38 35 35 71 62 68 64 38 0 64 24 23 24 62 64 51 55 60 64 0 65 59 60 68 59 67 64 38 24 65 0 36 31 69 61 61 62 35 23 59 36 0 10 61 59 60 62 35 24 60 31 10 0 59 71 59 68 71 62 68 69 61 59 0 abz06

Tabela 6.5 - Matrizes de distâncias relativas para instâncias do tipo *abz.* A distância máxima entre cada par de solução é 90 para as duas instâncias.

6.4.3 Lawrence (1984)

O trabalho sobre investigação de heurísticas para escalonamento *(Resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques)* publicado por Lawrence (1984) fez muitas análises de diversos métodos heurísticos além de propor um grupo de instâncias constantemente citadas e usadas até então.

A tabela 6.2 mostra os resultados do Sistema Imunológico Artificial quando aplicado em instâncias de Lawrence (1984).

Instância	L^*	***Conjunto-solução (K=10) encontrado pelo SIA***	***Tempo(s)***
la01	666	666 695 718 718 725 728 731 740 746 763	2,659
la06	926	926 936 944 964 965 968 971 975 985 986	17,451
la10	958	958 958 962 972 973 979 979 981 982 992	10,818
la13	1150	1150 1150 1150 1150 1150 1150 1150 1150 1150 1150	119,07
la15	1207	1207 1220 1223 1231 1239 1239 1251 1251 1251 1251	225,947
la19	842	850 856 867 872 872 873 876 878 887 888	163,721

Tabela 6.6 - Tabela de apresentação das k-soluções para instâncias do tipo *la*.

A diversidade das soluções pode ser vista nas Tabelas 6.7 e 6.8.

Instância	***Instância***
0 38 38 36 38 40 42 39 37 41 38 0 35 38 30 38 42 43 39 40 38 35 0 40 39 33 34 38 41 40 36 38 40 0 38 42 38 40 44 36 38 30 39 38 0 43 37 39 38 42 40 38 33 42 43 0 36 34 36 39 42 42 34 38 37 36 0 36 35 36 39 43 38 40 39 34 36 0 37 39 37 39 41 44 38 36 35 37 0 39 41 40 40 36 42 39 36 39 39 0 la01	0 59 66 63 58 60 59 59 65 59 59 0 63 63 66 65 63 61 60 64 66 63 0 64 61 65 59 63 64 69 63 63 64 0 61 53 64 60 61 67 58 66 61 61 0 65 64 66 64 67 60 65 65 53 65 0 62 59 60 66 59 63 59 64 64 62 0 61 64 64 59 61 63 60 66 59 61 0 65 62 65 60 64 61 64 60 64 65 0 63 59 64 69 67 67 66 64 62 63 0 la06

Tabela 6.7 - Matrizes de distâncias relativas das instâncias do tipo *la*. A distância máxima entre cada par de solução é 45 para a instância *la01, 7*0 para *la06.*

Instância	***Instância***
0 65 64 14 64 67 66 64 66 65 65 0 9 63 65 67 57 61 65 62 64 9 0 62 63 67 57 62 65 62 14 63 62 0 64 65 64 63 64 67 64 65 63 64 0 61 66 62 65 64 67 67 67 65 61 0 62 66 63 64 66 57 57 64 66 62 0 68 65 66 64 61 62 63 62 66 68 0 63 68 66 65 65 64 65 63 65 63 0 65 65 62 62 67 64 64 66 68 65 0 la10	0 33 58 55 61 71 70 91 72 72 33 0 45 39 41 65 68 89 66 69 58 45 0 23 36 50 61 90 59 70 55 39 23 0 21 46 57 91 53 65 61 41 36 21 0 38 44 92 46 55 71 65 50 46 38 0 30 90 41 44 70 68 61 57 44 30 0 91 24 36 91 89 90 91 92 90 91 0 88 91 72 66 59 53 46 41 24 88 0 37 72 69 70 65 55 44 36 91 37 0 la13
0 47 45 34 70 56 80 78 77 76 47 0 25 40 56 24 80 75 77 77 45 25 0 29 62 40 79 74 79 79 34 40 29 0 69 48 78 76 78 78 70 56 62 69 0 46 74 71 50 43 56 24 40 48 46 0 84 79 73 69 80 80 79 78 74 84 0 12 59 56 78 75 74 76 71 79 12 0 55 55 77 77 79 78 50 73 59 55 0 20 76 77 79 78 43 69 56 55 20 0 la15	0 8 15 51 54 15 43 60 38 58 8 0 19 51 55 7 41 60 38 58 15 19 0 57 58 26 33 55 35 58 51 51 57 0 15 51 57 53 66 54 54 55 58 15 0 55 59 51 67 52 15 7 26 51 55 0 38 61 39 60 43 41 33 57 59 38 0 62 26 65 60 60 55 53 51 61 62 0 68 23 38 38 35 66 67 39 26 68 0 68 58 58 58 54 52 60 65 23 68 0 la19

Tabela 6.8 - Matrizes de distâncias relativas para instâncias do tipo *la.* A distância máxima entre cada par de solução é *7*0 para *la10, 95* para *la13*, 95 para a instância *la15 e 90* para *la19*.

6.4.4 Applegate e Cook (1991)

Applegate e Coock (1991) propuseram uma heurística baseada em Método de Plano de Corte e Algoritmo de *Branch and Bound* em buscas de bons escalonamentos. As instâncias de Applegate e Coock (1991) fazem parte da *ORB Library*, site de Pesquisa Operacional bastante difundido.

A tabela 6.2 traz os resultados do SIA quando aplicado em instâncias de Applegate e Coock (1991).

Instância	L^*	*Conjunto-solução (K=10) encontrado pelo SIA*	*Tempo(s)*
orb02	888	888 895 895 899 902 902 902 903 904 908	822,342
orb05	887	889 889 889 889 901 916 918 918 918 918	702,768
orb07	397	397 400 403 404 404 404 404 404 405 405	805,312
orb10	944	944 944 952 955 955 956 958 958 959 959	973,114

Tabela 6.9 - Tabela de apresentação das k soluções para instâncias do tipo *orb*.

A diversidade das soluções pode ser observada na tabela 6.10.

Instância	*Instância*
0 38 37 31 69 63 35 75 31 44 38 0 5 42 71 64 32 62 16 47 37 5 0 39 72 61 32 61 14 51 31 42 39 0 70 55 35 73 35 46 69 71 72 70 0 66 69 20 73 67 63 64 61 55 66 0 59 66 60 54 35 32 32 35 69 59 0 70 24 44 75 62 61 73 20 66 70 0 69 73 31 16 14 35 73 60 24 69 0 45 44 47 51 46 67 54 44 73 45 0 orb02	0 12 8 3 23 47 66 63 66 67 12 0 8 9 11 42 63 64 63 68 8 8 0 5 19 40 65 66 65 70 3 9 5 0 20 44 64 65 64 69 23 11 19 20 0 32 66 67 65 67 47 42 40 44 32 0 71 72 70 72 66 63 65 64 66 71 0 14 14 24 63 64 66 65 67 72 14 0 6 22 66 63 65 64 65 70 14 6 0 23 67 68 70 69 67 72 24 22 23 0 orb05
0 17 60 63 59 61 63 63 50 48 17 0 63 65 62 64 66 62 54 52 60 63 0 55 59 57 54 55 65 68 63 65 55 0 19 17 20 25 35 35 59 62 59 19 0 2 11 12 40 40 61 64 57 17 2 0 9 14 42 42 63 66 54 20 11 9 0 5 40 40 63 62 55 25 12 14 5 0 38 38 50 54 65 35 40 42 40 38 0 3 48 52 68 35 40 42 40 38 3 0 orb07	0 12 50 67 64 22 25 54 32 30 12 0 50 64 64 24 27 50 23 28 50 50 0 53 51 48 48 34 49 57 67 64 53 0 4 66 64 64 62 64 64 64 51 4 0 65 63 64 62 65 22 24 48 66 65 0 15 51 27 36 25 27 48 64 63 15 0 47 17 33 54 50 34 64 64 51 47 0 47 56 32 23 49 62 62 27 17 47 0 23 30 28 57 64 65 36 33 56 23 0 orb10

Tabela 6.10 - Matrizes de distâncias relativas para instâncias do tipo *orb*. A distância máxima entre cada par de solução é 90 para as quatro instâncias.

6.5 Análise dos resultados

Nesta seção serão apresentados os resultados encontrados pelo Sistema Imunológico Artificial. Nas instâncias que tiveram maiores tempos de execução, foi exigido um esforço considerável do algoritmo. Isto é devido ao número de anticorpos e iterações necessárias para atingir o objetivo de tentar manter a diversidade da inicialização do repertório e prover melhorias a um conjunto de soluções.

Na tabela 6.11, podem ser observados (L^*), *conjunto-solução* (K) e *o tempo de execução* para cada instância. É visível a qualidade das soluções encontradas pelo algoritmo, pois em praticamente todos os testes L^* faz parte de K e as demais que o compõe possuem *makespans* muito baixos.

Instância	L^*	***(K=10)***	***Tempo(s)***
ft06	55	55 55 58 58 59 59 59 60 60 60	0,752
ft10	930	930 948 955 958 959 960 961 962 964 965	732,21
abz05	1234	1234 1251 1255 1256 1258 1261 1264 1264 1265 1267	580,06
abz06	943	945 948 950 952 962 962 962 962 962 966	393,885
la01	666	666 695 718 718 725 728 731 740 746 763	2,659
la06	926	926 936 944 964 965 968 971 975 985 986	17,451
la10	958	958 958 962 972 973 979 979 981 982 992	10,818
la13	1150	1150 1150 1150 1150 1150 1150 1150 1150 1150 1150	119,07
la15	1207	1207 1220 1223 1231 1239 1239 1251 1251 1251 1251	225,947
la19	842	850 856 867 872 872 873 876 878 887 888	163,721
orb02	888	888 895 895 899 902 902 902 903 904 908	822,342
orb05	887	889 889 889 889 901 916 918 918 918 918	702,768
orb07	397	397 400 403 404 404 404 404 404 405 405	805,312
orb10	944	944 944 952 955 955 956 958 958 959 959	973,114

Tabela 6.11 - Tabela sintetizada das k soluções por instância.

Também é mostrada na Figura 6.1 a qualidade dos conjuntos em termos de *makespan*. Tomando as soluções em ordem crescente de custos, pode-se observar que eles estão muito próximos de *(L*)*. Neste gráfico é apresentado por instância a *melhor solução de conjunto (K*)*, *(L*)* e por fim, a *pior solução do conjunto (K`)*.

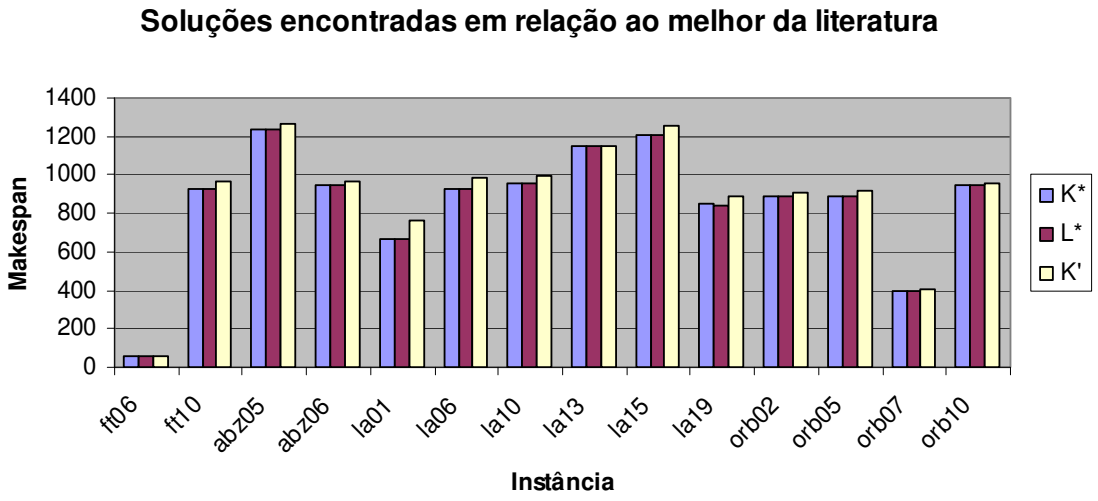


Figura 6.1 - Gráfico com melhor e pior solução de *K* por instância.

Ainda é possível visualizar na Tabela 6.12 os seguintes desvios em relação a L^*: da melhor solução do conjunto $(\delta K^* L^*)$, média dos custos das soluções $(\delta \overline{K} L^*)$ e da pior solução $(\delta K' L^*)$.

Instância	$\delta K^* L^*$	$\delta \overline{K} L^*$	$\delta K' L^*$
ft06	0,00%	5,98%	9,09%
ft10	0,00%	2,82%	3,76%
abz05	0,00%	1,90%	2,67%
abz06	0,21%	1,49%	2,44%
la01	0,00%	8,56%	14,56%
la06	0,00%	3,89%	6,48%
la10	0,00%	1,63%	3,55%
la13	0,00%	0,00%	0,00%
la15	0,00%	2,43%	3,65%
la19	0,95%	3,55%	5,46%
orb02	0,00%	1,33%	2,25%
orb05	0,23%	1,97%	3,49%
orb07	0,00%	1,51%	2,02%
orb10	0,00%	1,06%	1,59%

Tabela 6.12 - Tabela com os desvios em relação a melhor solução da literatura.

Os grupos de soluções tiveram uma média global dos desvios $\delta\overline{KL}^*$ inferior a 3,9%, confirmando o bom desempenho do algoritmo (Tabela 6.12).

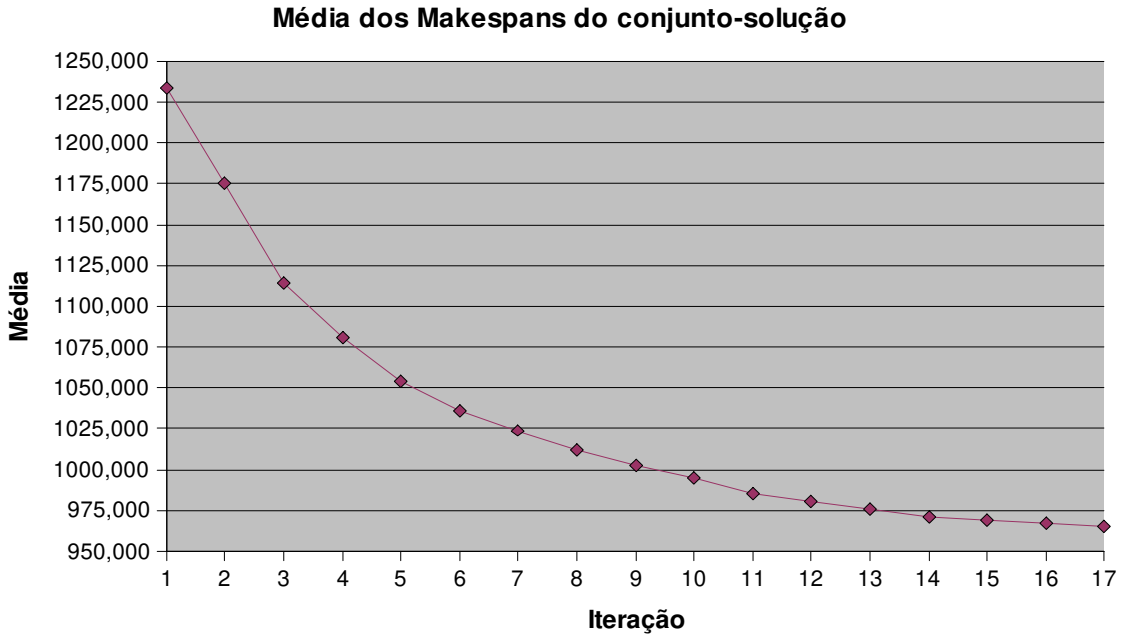


Figura 6.2 - Gráfico com a média do conjunto-solução por iteração; exemplo, la06.

O surgimento e refinamento de anticorpos com alta afinidade são direcionados pelo critério de redução do *makespan*, ou seja, é o guia principal de busca do algoritmo. E, devido a Seleção e Clonagem, a média dos *makespans* do conjunto $(\overline{K})$, será sempre menor ou igual quando comparados à interação anterior. A Figura 6.2 traz um exemplo desse comportamento representado na instância *la06*. Nas demais, este tipo de padrão se repete.

Sendo o objetivo encontrar um grupo de soluções de boa qualidade com uma diversidade acentuada, na tabela 6.13 é possível visualizar por instância, (L^*), (K^*), $(\overline{K})$, *distância máxima entre soluções* $(D_{\max})$, *média das distâncias entre soluções* $(\overline{D})$ e por fim o $AQGS_{10}$. Fica claro que o conjunto *K* possui um bom espalhamento na superfície de busca, uma vez que a *média da distância por grupo* possui valor considerável em relação à *distância máxima entre soluções*.

Instância	L^*	K^*	$\overline{K}$	$D_{\max}$	$\overline{D}$	$AQGS_{10}$
ft06	55	55	58,29	30	20,17	0,387
ft10	930	930	956,2	90	59,4	0,368
abz05	1234	1234	1257,5	90	55,533	0,402
abz06	943	945	957,09	90	53,599	0,419
la01	666	666	723	45	38,288	0,234
la06	926	926	962	70	62,57	0,144
la10	958	958	973,59	70	61,71	0,134
la13	1150	1150	1150	95	58,53	0,383
la15	1207	1207	1236,3	95	60,71	0,385
la19	842	850	871,9	90	46,28	0,52
orb02	888	888	899,799	90	50,333	0,454
orb05	887	889	904,5	90	44,97	0,519
orb07	397	397	403,00	90	42,48	0,543
orb10	944	944	954	90	44,911	0,511

Tabela 6.13 - Tabela de apresentação dos avaliadores de conjunto-solução *K=10*.

Um Sistema Imunológico Artificial possui a característica inerente de proporcionar alta diversidade nos anticorpos de seu repertório (Freitas Filho, 2007). Além disso, políticas adotadas como *Supressão* e *Tratamento a Ótimos Locais* reforçam tal aspecto. Assim, uma vez que a média das distâncias se estabilize no decorrer da busca, o repertório possuirá uma boa diversidade com poucas variações durante as **iterações** de um teste.

Nos gráficos da Figura 6.3 e Figura 6.4 encontram-se exemplos de rodadas diferentes da instância *la06*. A média das distâncias sempre pertence ao intervalo de variação [62,65], onde a última iteração de cada uma delas são respectivamente **62,570** e **62,508**. Ambas representam uma razão em torno de **0,89** de uma diversidade máxima **1**, valores praticamente iguais que corroboram com o comportamento descrito. Isto se repetiu em outras instâncias.

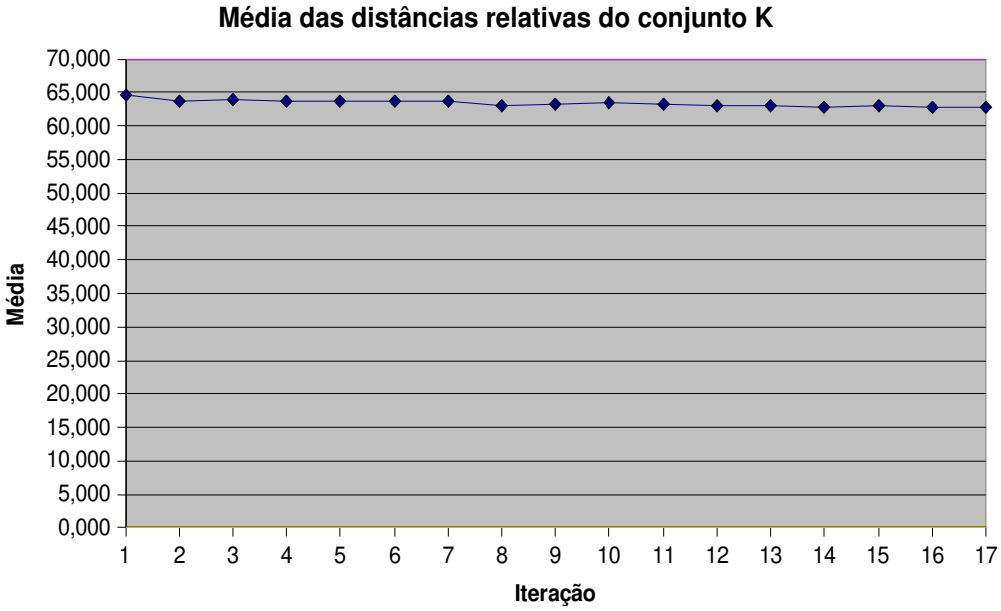


Figura 6.3 - Média das distâncias relativas do conjunto-solução, exemplo *la06*. Se as soluções do conjunto fossem integralmente distintas a média seria 70. Caso as soluções sejam idênticas, a média é 0.

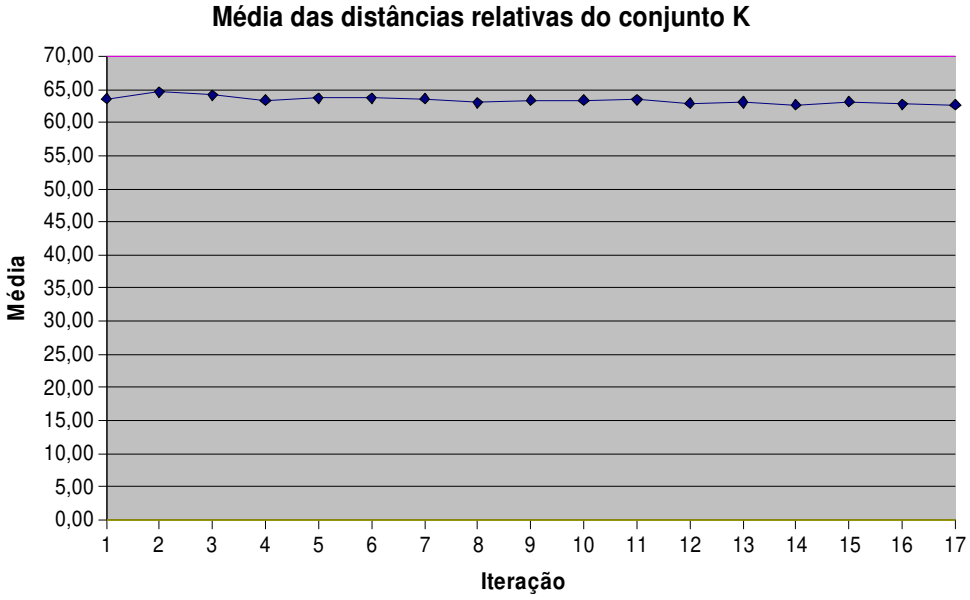


Figura 6.4 - Média das distâncias do conjunto-solução de *la06* (outra rodada). Devido às características de manutenção de diversidade inerentes ao SIA e pelas políticas adotadas para reforçá-las, durante todas as iterações de todas as rodadas deste teste, esta média se estabiliza mantendo-se no intervalo de [62,65].

Outro comportamento observado em algumas instâncias, por exemplo, *orb10*, foi à diminuição da média das distâncias relativa antes da estabilização em torno do valor de baixa variação, Figura 6.5. Isto é devido à especificidade do tipo de instância que exigiu uma queda da diversidade para se obter um bom conjunto-solução. Mesmo assim o SIA foi capaz de encontrar boas soluções mantendo um nível considerável de diversidade entre elas.

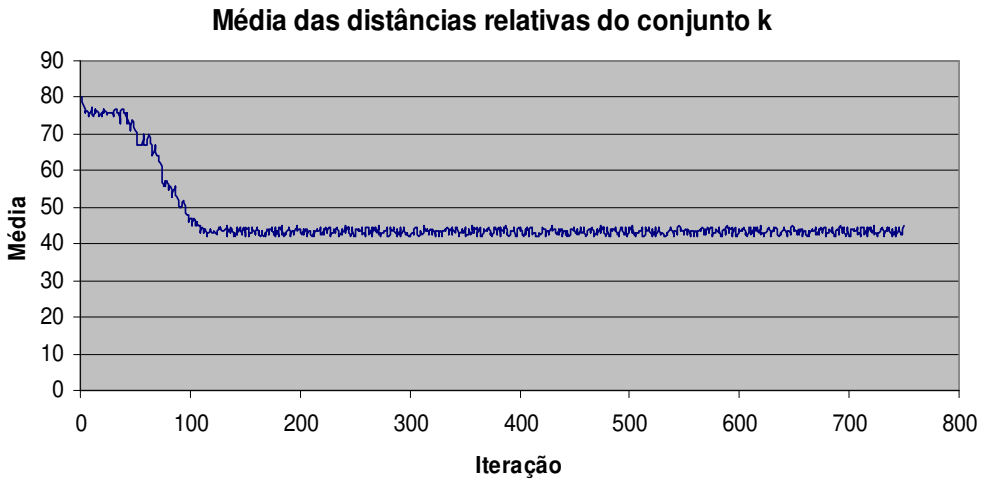


Figura 6.5 - Gráfico com a avaliação do conjunto solução por iteração; instância exemplo *orb10*.

É considerado como grau de *diversidade máxima 1* quando cada par de solução do conjunto possui a distância máxima definida pela métrica. Por exemplo, se em um conjunto as solução fossem 100% diferentes entre si, o $\overline{D}$ seria igual ao $D_{\max}$. A Figura 6.6 apresenta o grau de diversidade do conjunto-solução encontrado em relação ao grau máximo. Pode-se observar a alta diversidade entre as soluções promovida pelo sistema imunológico aplicado.

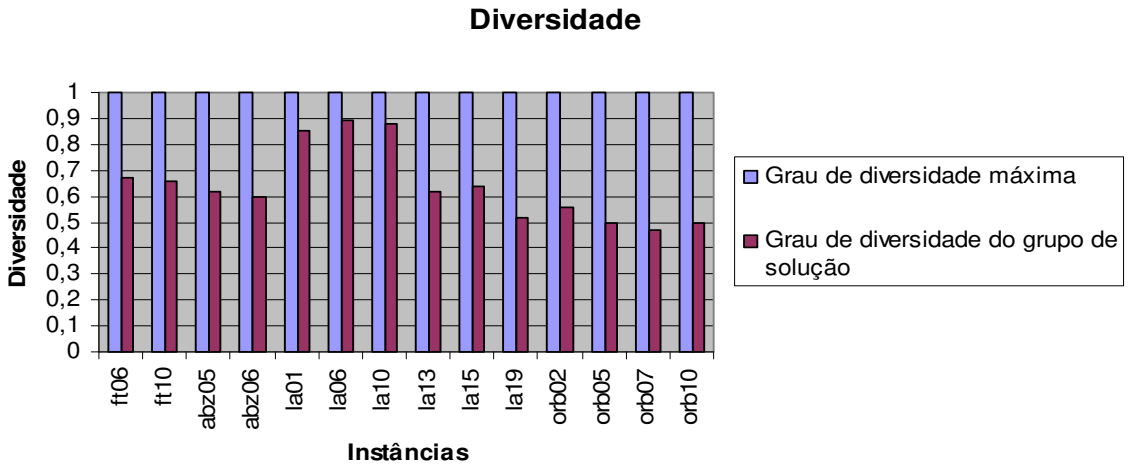


Figura 6.6 - Gráfico de diversidade das soluções.

A característica de diversidade também pode ser visualizada nas *Matrizes de Distâncias Relativas* apresentadas nos blocos de testes por autor. Lá está explicitado o quanto as soluções **não** estão concentradas em uma única região, mas distribuídas ao máximo levando em consideração a qualidade dos *makespans*.

Por fim, é esperada a redução do fator $AQGS_K$ durante as iterações devido à forte interferência de redução de $\overline{K}$ (Figura 6.7).

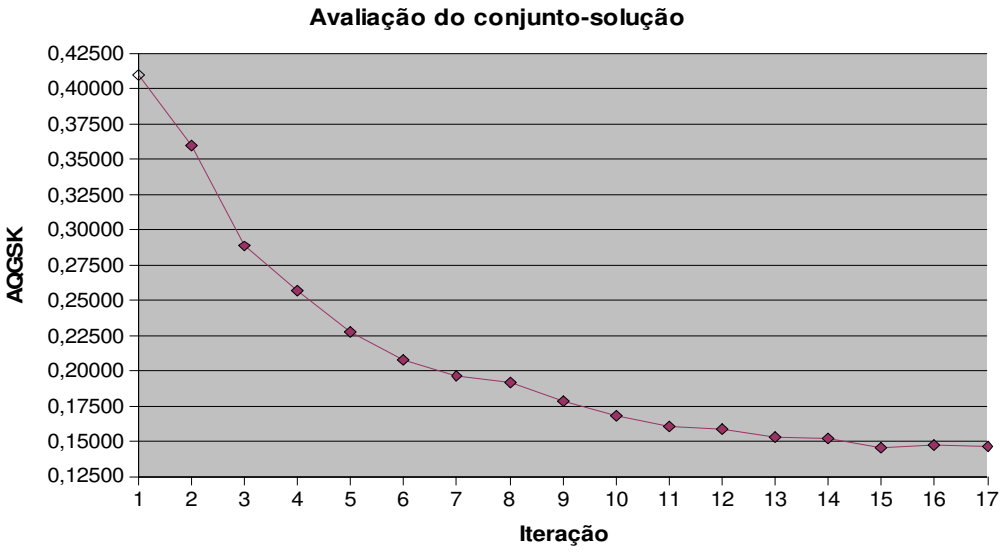


Figura 6.7 - Avaliação do conjunto solução por iteração; instância exemplo *la06*.

Também é comum existir pequenos aumentos seguidos de reduções causados por leves variações de $\overline{D}$ em instantes que $\overline{K}$ não teve redução significativa (Figura 6.7). Isso não causa grandes oscilações e nem provoca impacto no comportamento de redução deste fator de avaliação.

Nas demais instâncias o decaimento de $AQGS_K$ teve um comportamento semelhante.

Capítulo 7

Conclusão

7.1 Considerações Finais

A simulação de processos evolutivos naturais permite a compreensão, por exemplo, de como os seres vivos são direcionados em busca de um maior nível de inteligência. A evolução encontra, ao final do processo, soluções que apresentam um comportamento mais adaptado ao ambiente. Esta modelagem emprega uma série de algoritmos de otimização baseados em simples regras, implementados como método de busca em superfícies de resposta de desempenho. O processo de otimização, inerente à seleção dos elementos mais adaptados, melhora a qualidade das soluções obtidas ao longo das gerações. Entretanto, o grande potencial dos algoritmos baseados em computação natural vem da integração de ampla exploração do espaço de busca junto a processos de busca mais localizada. Essa integração resulta em uma robustez acentuada que permite a aplicação em diversos problemas práticos.

A transcrição dos aspectos imunológicos na construção dos escalonamentos tornou o processo de otimização robusto e eficiente. A representação da solução, através de grafos, deu subsídio para o desenvolvimento de procedimentos (movimentação, vizinhança e diversidade) que promoveram alta exploração e explotação da superfície de busca.

Os resultados obtidos pelo algoritmo proposto estão dentro dos objetivos inicias deste trabalho. Tais objetivos não se resumem apenas na elaboração de uma ferramenta de resolução do escalonamento de um o *Job Shop* com o objetivo de minimizar o *makespan*, mas um sistema capaz de demonstrar eficiência em diferentes especificidades de teste, encontrando soluções diversificadas e de alta qualidade. Vale ressaltar que um dos principais fatores que contribuiu com a qualidade do algoritmo, dentre outros observados neste trabalho, foi à exploração/explotação do espaço de busca proporcionadas pelas características inerentes ao sistema como diversidade imunológica e seleção clonal.

Não foi possível estabelecer comparações diretamente a outros resultados, pois não foram encontrados trabalhos que tratassem concomitantemente qualidade e diversidade de conjunto de soluções para o *Job Shop*. Sendo assim, torna-se contribuição considerável a abordagem imunológica realizada aqui.

7.2 Perspectivas futuras

Para testar o Sistema Imunológico Artificial implementado foram usadas instâncias clássicas da literatura. Devido ao grande número de trabalhos e aplicações de outros autores, já são conhecidas soluções muito boas. Mas se for necessário à aplicação em novas instâncias onde não é conhecida nenhuma solução, torna-se necessária uma definição mais robusta para a afinidade. Uma maneira de contornar tal situação é definir $\tilde{A}$ como o melhor anticorpo encontrado até o momento, assim a definição da afinidade torna-se adaptativa de acordo com a evolução do algoritmo.

A implementação de uma busca local dentro do sistema imunológico artificial, logo após a maturação da população de clones, pode fazer com que o processo de exploração do espaço de busca seja otimizado, possibilitando obter resultados melhores (Almeida, 2006).

Sistemas Imunológicos Artificiais aplicados em outras variações do *Job Shop* como *Job Shop* com parâmetros incertos (Fuzzy) e *Job Shop* Dinâmico, podem ser métodos eficientes de resolução. O SIA demonstra bom comportamento quando tratados grafos CRISP e Fuzzy (Almeida, 2006). No *Job Shop* Dinâmico os custos de operação mudam durante o tempo, o SIA por promover alta diversidade, tende a ser robusto a essa variação devido à alta capacidade de mobilidade em busca de novas soluções (De Castro, 2001).

Referências

1. **Ada, G. L. & Nossal, G. J. V. (1987*),*** 'The Clonal Selection Theory', *Scientific American*, 257(2), pp. 50-57.

2. **Adams, J., Balas E., Zawack D. (1988),** 'The shifting bottleneck procedure for job shop scheduling', *Management Science* 34, 391-401.

3. **Alisantaso, D., Koo, L. P., Jiang, P. Y. (2003),** 'An immune algorithm approach to the scheduling of a flexible PCB flow shop', Int J Adv Manuf Technol, 22: 819–827.

4. **Almeida, T., A.** *Computação Evolutiva Aplicada à Resolução do Problema da Árvore Geradora Mínima com Parâmetros Fuzzy.* Campinas: Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas, 2006, Tese (Mestrado).

5. **Applegate, D., W. Cook W., (1991),** 'A computational study of the job-shop scheduling instance', *ORSA Journal on Computing* 3, 149-156.

6. **Bagchi, S., Uckun, S., Miyabe, Y., & Kawamura, K. (1991)**, Exploring problem-specific recombination operators for job shop scheduling. *Proceedings of the Fourth International Conference on Genetic Algorithms*, San Mateo: Morgan Kaufmann, pp. 10–17.

7. **Balas, E. (1969),** 'Machine Scheduling via disjunctive graphs'*, Operations Research,* 17: 941-957.

8. **Binato, S., Hery, W.J., Loewenstern, D.M. and Resende, M.G.C., (2002),** A GRASP for Job Shop Scheduling. In: Essays and Surveys in Metaheuristics, Ribeiro, Celso C., Hansen, Pierre (Eds.), Kluwer Academic Publishers.

9. **Bierwirth, C. (1995). In E. Pesch, & S. Vo. (1995),** 'A generalized permutation approach to job shop scheduling with genetic algorithms'. *OR-Spektrum* , pp. 87–92. *Special issue: Applied Local Search*, 17(213).

10. **Bonfim, Tatiane Regina.** *Escalonamento Memético e Neuro-Memético de Tarefas*. Campinas: Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas, 2006, Tese (Doutorado).

11. **Brucker, P., Jurisch, B. and Sievers, B., (1994),** *A Branch and Bound Algorithm for Job-Shop Scheduling Problem*, Discrete Applied Mathematics, Vol 49, pp. 105-127.

12. **Burnet, F. M. (1978***), "Clonal Selection and After, Theoretical Immunology"*, (Eds.) G. I. Bell, A. S. Perelson & G. H. Pimbley Jr., Marcel Dekker Inc., pp. 63-85.

13. **Caldeira J. P., Melicio F., Rosa A, C. (2004***),* 'Using a hybrid Evolutionary –Taboo Algorithm to solve Job Shop Problem'. *SAC 2004*: 1446-1451.

14. **Carlier, J. and Pinson, E., (1989),** 'An Algorithm for Solving the Job Shop Problem'. *Management Science*, Feb, 35(29), pp.164-176.

15. **Cheng, R., Gen, M. and Tsujimura, Y. (1999),** 'A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies', *Computers & Industrial Engineering*, Vol. 36, pp. 343-364.

16. **Coello Coello, C. A. and Cortés, N.C. (2005), '**Solving Multiobjective Optimization Problems Using an Artificial', Immune System, *Genetic Programming and Evolvable Machines*, 6, 163-190.

17. **Costa, A. M., Vargas P. A., Von Zuben f. J., França P. M. (2002), '**Makespan Minimization on Parallel Processors: An Immune-Based Aproache', *IEEE.*

18. **Dasgupta, D. (1998),** 'Artificial Immune Systems and Their Applications', *Springer-Verlag,* Nova York.

19. **Davis, L. (1985)**, Job shop scheduling with genetic algorithms. *Proceedings of the International Conference on Genetic Algorithms and their Applications*, Hillsdale: Lawrence Erlbaum, pp. 136–149.

20. **De Castro, L.N. ,** *Engenharia Imunológica: Desenvolvimento e Aplicação de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais.*

Campinas: Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 2001, Tese (Doutorado)

21. **De Castro, L.N. & Von Zuben, F.J. (1999)**, 'Artificial Immune Systems: Part I – Basic Theory and Applications'*, Technical Report* – RT DCA 01/99.

22. **Dorndorf, U., & Pesch, E. (1993),** 'Combining genetic and local search for solving the job shop scheduling problem', *APMOD93 Proc. Preprints*, Budapest, Hungary, pp. 142–149.

23. **Dreher, H. (1995)**, 'The Immune Power Personality'*, Penguim Books.*

24. **Fisher, H., Thompson, G.L. (1963),** *Probabilistic learning combinations of local job-shop scheduling rules.* Industrial Scheduling. Prentice-Hall, Englewood Cliffs, NJ, pp. 225–251

25. **Freitas Filho, L. J. R., Berbert, P. C., Almeida, T. A., Yamakami, A & Carvalho M. B. (2007), '**Artificial Immune System to Find a Set of k-Spanning Trees with Low Costs and Distinct Topologies', *International Conference on Artificial Immune Systems – Santos/Brazil.*

26. **George, A. J. T. & Gray, D. (1999***),* 'Receptor Editing during Affinity Maturation, Imm'. *Today*, 20(4), pp. 196.

27. **Giffler, B. and Thompson, G.L., (1960),** 'Algorithms for Solving Production Scheduling Problems', *Operations Research*, Vol. 8(4), pp. 487-503.

28. **Gonçalves, J.F. and Mendes, J.M., (1994).** A Look-Ahead Dispatching Rule for Scheduling Operations. In: VII Latin-Ibero-American Conference on Operations Research and Systems Engineering, University of Chile, Santiago, Chile.

29. **Gonçalves, J.F., Mendes, J.M., and Resende, M.C., (2002)**, 'A Hybrid Genetic Algorithm for the Job Shop Scheduling Problem', *AT&T Labs Research Technical Report* TD-5EAL6J.

30. **Gonçalves, J.F, Mendes, J.J.M, Resende, M.G.C. (2005),** 'A hybrid genetic algorithm for the job shop scheduling problem'. *European Journal of Operational Research* 167, pages 77–95.

31. **Gray, C. and Hoesada, M. (1991),** 'Matching Heuristic Scheduling Rules for Job Shops to the Business Sales Level', *Production and Inventory Management Journal*, Vol. 4, pp. 12-17

32. **Hart, E., Ross P.,Nelson J. (1998), '**Producing Robust Schedulings Via an Artificial Immune System'. *Proc. do ICEC'98*.

33. **Hightower, R. R., Forrest S. & Perelson, A. S. (1996),** *The Baldwin Effect in the Immune System: Learning by Somatic Hypermutation*, R. K. Belew and M. Mitchel (Eds.), Adaptive Individuals in Evolving Populations, Addison-Wesley, Reading, MA, pp. 159-167.

34. **Hightower, R. R., Forrest, S. A & Perelson, A. S. (1995)**, *The Evolution of Emergent Organization in Immune System Gene Libraries*, Proc. of the Sixth ICGA, L. J. Eshelman (Ed.), Morgan Kaufmann, San Francisco, CA, pp. 344-350.

35. **Janeway, C. A., P. Travers, Walport, M. & Capra, J. D. (2000*),*** 'Imunobiologia: O Sistema Imunológico na Saúde e na Doença', *Artes Médicas*, 4a Ed.

36. **Kobayashi, S., Ono I., Yamamura M. (1995),** 'An eficient genetic algorithm for job shop scheduling problems'. In: 6th ICGA, pages 506 - 511.

37. **Laguna, M., Glover, F. (1997),** 'Tabu Search', *Kluwer Academic Publishers*, Norwell, MA.

38. **Lawrence, S. (1984),** 'Resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques', *GSIA*, Carnegie Mellon University, Pittsburgh, PA.

39. **Lederberg, J. (1988),** 'Ontogeny of the Clonal Selection Theory of Antibody Formation', *Annals of the New York Ac. of Sc.*, 546, 175-182.

40. Lenstra, J.K., Rinnooy K. A. H. G., Brucker P (1977), 'Complexity machine scheduling problems'. *Ann. Discrete Math.*, 1:343–362, 1977.

41. Lourenço, H.R. (1995), 'Local optimization and the job-shop scheduling problem', *European Journal of Operational Research*, Vol. 83, pp. 347-364.

42. Lourenço, H.R. and Zwijnenburg, M. (1996), 'Combining the large-step optimization with tabu-search: Application to the job-shop scheduling problem'. In I.H. Osman and J.P. Kelly, editors, Metaheuristics: Theory and Apllications, pp. 219-236, Kluwer Academic Publishers.

43. Mahfoud, S. W. (1995), *Niching Methods for Genetic Algorithms*, Tese de Doutorado, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.

44. Melián, B., Pérez, J. A. M., Vega, J. M. M. Metaheuristics (2003), 'A Global View In Revista Iberoamericana de Inteligencia Artificial', *Asociación Española de Inteligencia Artificial*, 2003. v. 2, n. 19.

45. Michalewicz, Z. (1996), 'Genetic Algorithms + Data Structures = Evolution Programs', 3rd edn, *Springer-Verlag*, Berlin Heidelberg New York.

46. **Mori, K., Tsukiyama M., Fukuda T. (1998),** 'Adaptive Scheduling System Inspired by Immune System'*, IEEE*, 3833-3837.

47. **Nakano, R. and Yamada, T. (1991).** Conventional Genetic Algorithm for Job-Shop Problems. In Kenneth, M. K. and Booker, L. B. (eds) Proceedings of the 4th International Conference on Genetic Algorithms and their Applications, San Diego, USA, pp. 474-479.

48. **Nawaz, M., Enscore, JR., and Ham, I. (1983),** 'A Heuristic Algorithm for the *m*-machine, *n*-job Flow-shop Sequencing Problem', OMEGA International J. Management Sci., 11, 1, 91-95.

49. Nicholson, T. (1971), *Optimization in industry: Optimization techniques*. Longman press, Vol. 1, Chapter 10, London, UK.

50. Nowicki, E., Smutnicki, C. (1996), 'A fast taboo search algorithm for the job-shop problem', *Management Science* 42 (6), 797-313.

51. Pinson, E., (1995). The job shop scheduling problem: A concise survey and some recent developments. *In*: Chrétienne, P., Coffman Jr., E.G., Lenstra, J.K., Liu, Z. (Eds.), Scheduling theory and its application, John Wiley and Sons, pp. 277-293.

52. Resende, M. G. C., (1997), 'A GRASP for Job Shop Scheduling', *INFORMS Spring Meeting*, San Diego, California, USA.

53. Scrich, C. R. *Busca Tabu para programação de tarefas em Job Shop com datas de entrega.* Campinas: Faculdade de Engenharia Elétrica e de Computação, Univ. Estadual de Campinas, 1997, Tese (Doutorado)

54. Storb, U. (1998), 'Progress in Understanding the Mechanism and Consequences of Somatic Hypermutation'*, Immun. Rev.*, 162, pp. 5-11.

55. Sutton, R. S. & Barto, A. G. (1998), 'Reinforcement Learning an Introduction', *A Bradford Book*.

56. Tamaki, H. and Nishikawa, Y. (1992), 'A Parallelled Genetic Algorithm Based on a Neighbourhood Model and its Application to the JobShop Scheduling'. In Männer, R. and Manderick, B. (eds) PPSN'2 *Proceedings of the 2nd International Workshop on Parallel Problem Solving from Nature*, Brussels, Belgium, pp. 573-582.

57. Tizard, I. R. (1995), 'Immunology An Introduction', *Saunders College Publishing*, 4th Ed.

58. Vaessens, R.J.M., Aarts, E.H.L. and Lenstra, J.K., (1996), 'Job Shop Scheduling by local search', *INFORMS Journal*.

59. Van Laarhoven, P. J.M., Aarts E.H. L., Lenstra J.K. (1992), 'Job Shop Scheduling by Simulated Annealing', *Operations Research*, volume 40, página 113-125.

60. Wang, L. and Zheng, D. (2001), 'An effective hybrid optimisation strategy for job-shop scheduling problems', *Computers & Operations Research*, Vol. 28, pp. 585-596.

61. Warner F., Winkler A. (1992), 'Insertion Techniques for the Heuristic Solution of the Job Shop Problem', *Report*, Technische Universität "Otto von Guericke", Magdeburg, Germany.

62. Williamson, D. P., Hall, L. A., Hoogeveen, J. A., Hurkens, C. A. J., Lenstra, J. K., Sevast'janov, S. V. and Shmoys, D. B. (1997), 'Short Shop Schedules', *Operations Research*, March - April, **45**(2), pp. 288-294.

63. Yamada, T. and Nakano, R. (1992), 'A Genetic Algorithm Applicable to Large-Scale Job-Shop Problems', in Männer, R. and Manderick, B. (eds) PPSN'2 *Proceedings of the 2nd International Workshop on Parallel Problem Solving from Nature*, Brussels, pp. 281-290

64. Yamada, T. and Nakano, R. (1995a), 'Job-Shop Scheduling by Simulated Annealing Combined with Deterministic Local Search', *MIC'95 Meta-heuristics International Conference*, Hilton, Breckenridge, Colorado, USA, July 22-26, pp. 344-349.

65. **Yamada, T., Nakano, R. (1997),** 'Genetic Algorithms for Job-Shop Scheduling Problems'. *Proceedings of Heuristic for Decision Support*, pp 67-81, UNICOM seminar, 18-19 March, London.

66. **Wikipédia, (2006),** Site internacional de pesquisa na internet. http://pt.wikipedia.org/wiki/Imunologia