

SISTEMA IMUNOLÓGICO ARTIFICIAL PARA RESOLVER O PROBLEMA DA ÁRVORE GERADORA MÍNIMA COM PARÂMETROS *FUZZY*

Tiago Agostinho Almeida *

Universidade Estadual de Campinas (UNICAMP)
Campinas – SP
tiago@dt.fee.unicamp.br

Akebo Yamakami

Universidade Estadual de Campinas (UNICAMP)
Campinas – SP
akebo@dt.fee.unicamp.br

Márcia Tomie Takahashi

Petrobrás
Rio de Janeiro – RJ
mtomie@uefs.br

* *Corresponding author* / autor para quem as correspondências devem ser encaminhadas

Recebido em 04/2006; aceito em 09/2006

Received April 2006; accepted September 2006

Resumo

Neste trabalho é proposta uma meta-heurística baseada em técnicas da computação evolutiva que visa encontrar um conjunto de árvores geradoras mínimas, para problemas de grafos que possuem incertezas em relação às informações associadas aos parâmetros. Resolver problemas dessa natureza é um processo *NP*-Completo, pois envolve um número enorme de comparações. A fim de contornar essa complexidade, é proposto um sistema imunológico artificial capaz de explorar eficientemente o espaço de busca e de obter resultados satisfatórios, sem a necessidade de confrontar todas as soluções entre si.

Palavras-chave: computação evolutiva; grafos *fuzzy*; programação matemática *fuzzy*.

Abstract

This work proposes an heuristical approach based on evolutionary computation, whose goal is to find a set of minimum spanning trees in graphs that contain uncertainties in their parameters. This kind of problem is a *NP*-Hard one, because it involves an enormous number of comparisons. In order to avoid this complexity, this work proposes an artificial immune system that explores efficiently the search space of solutions to looking for satisfactory results, without the necessity of comparing all possible solutions.

Keywords: evolutionary computation; fuzzy graphs; fuzzy mathematical programming.

1. Introdução

A teoria dos grafos é comumente utilizada na área da engenharia para resolver problemas que podem ser representados na forma de redes (Ahuja *et al.*, 1993; Bazaraa *et al.*, 1990). Ela fornece uma modelagem prática e consistente de um problema, facilitando a implementação de algoritmos que auxiliam na obtenção da solução (Ahuja *et al.*, 1993).

Acredita-se que a teoria dos grafos foi introduzida em 1736 quando Eüler propôs uma solução para o problema das Pontes de Königsberg, apresentando uma condição necessária para se percorrer um grafo, sem repetir arestas e retornando ao ponto inicial. Entretanto, foi na segunda metade do século XX que essa teoria teve um enorme crescimento, com inúmeros problemas sendo propostos (Ahuja *et al.*, 1993). Atualmente, problemas bem conhecidos como caminho mínimo, árvore geradora mínima, fluxo máximo, emparelhamento, entre outros, são muito estudados por essa teoria (Goldberg & Luna, 2000).

O problema da árvore geradora mínima aparece em uma série de aplicações (Chunde, 1996; Chang & Lee, 1999; Raidl & Julstrom, 2003), por exemplo, na instalação de linhas telefônicas (ou elétricas) entre um conjunto de localidades, utilizando a infra-estrutura das rodovias com o menor uso de material. Outros casos como análise de *clusters*, armazenamento de informações, dentre outros, também podem ser resolvidos por essa modelagem, que possui eficientes algoritmos como *Kruskal*, *Prim* e *Sollin* (Ahuja *et al.*, 1993).

Em problemas reais é comum encontrar-se incertezas nas informações associadas, pois muitas vezes os dados são imprecisos e conceitos como cheio e vazio, alto e baixo, quente e frio, perto e longe, etc; dependem do ponto de vista do observador. Dessa forma, as tomadas de decisões são baseadas no conhecimento individual sem ter certeza absoluta sobre essas informações (Gomide & Pedrycz, 1998).

Ao trabalhar com dados incertos, uma informação deixa de ser representada por um número e passa a ser representada por um conjunto. Assim, o uso da teoria clássica dos conjuntos torna-se inviável devido a sua ineficiência no tratamento de informações imprecisas. Entretanto, essas incertezas podem ser estudadas e modeladas de forma mais robusta, utilizando a teoria dos conjuntos nebulosos, também conhecida como teoria dos conjuntos *fuzzy* (Zadeh, 1965; Dubois & Prade, 1980; Gomide & Pedrycz, 1998).

A teoria dos conjuntos nebulosos foi introduzida por Lotfi A. Zadeh em 1965 (Zadeh, 1965). A partir de então, essa teoria passou a desempenhar um papel fundamental nas mais diversas áreas do processamento de informação, representando uma ponte de ligação entre o processamento simbólico e o numérico. Com ela é possível utilizar símbolos (termos lingüísticos) aos quais estão associadas semânticas bem definidas que, após serem convertidas em funções de pertinência de conjuntos nebulosos, possibilitam o processamento em forma numérica. Em sistemas baseados em regras nebulosas, as informações imprecisas e incompletas são tratadas de forma lingüística, em analogia às informações sensoriais recebidas pelos órgãos e interpretadas pelo cérebro (Gomide & Pedrycz, 1998).

O tratamento das incertezas em problemas de grafos e as principais definições de grafos *fuzzy* foram propostos por Rosenfeld em 1975 (Rosenfeld, 1975), marco inicial da teoria dos grafos *fuzzy*. Em contraste aos inúmeros trabalhos existentes utilizando a teoria de grafos clássica, o estudo dos grafos *fuzzy* ainda está na sua fase inicial, tanto na teoria quanto na prática. Isso se deve, as diferentes abordagens que um problema de grafos *fuzzy* pode ter, pois os níveis de incerteza podem estar associados tanto na estrutura (nós e/ou arestas) quanto nos parâmetros (custo, capacidade, demanda).

Segundo Ahuja *et al.* (1993), grande parte dos problemas de grafos, embora com número finito de possibilidades, se mostra computacionalmente intratável. Um dos motivos é que problemas desse tipo tornam-se mais complexos conforme se aumenta o número de nós. Em muitos casos, é praticamente impossível garantir a solução ótima global do problema, e somente com a utilização de heurísticas é possível obter soluções aproximadas (Michalewicz, 1996).

Atualmente, existe um grande número de heurísticas que são utilizadas para resolver diversos tipos problemas (Bäck *et al.*, 2000a; De Castro, 2001; Holland, 1992). Muitas delas procuram simular processos evolutivos, por exemplo, os algoritmos genéticos e os sistemas imunológicos artificiais. Essas heurísticas fazem parte de um grupo chamado computação evolutiva.

A computação evolutiva, área da computação que simula os mecanismos de evolução através de computadores, possibilitou o surgimento de uma nova filosofia de máquinas inteligentes. Esse conceito tem por base a capacidade de um sistema adaptar o seu comportamento para atingir os seus objetivos numa variedade de situações.

A simulação de processos evolutivos biológicos permite o entendimento de como a evolução guia os seres vivos na direção de um maior nível de inteligência. A evolução, caracterizada por diferentes níveis de representação (gene, cromossomo, indivíduo, espécie, ecossistema), converge para soluções que apresentam um comportamento mais adaptado ao ambiente.

A modelagem do processo evolutivo emprega uma série de algoritmos de otimização baseados em regras simples, e implementados como métodos de busca em superfícies de resposta de desempenho (*fitness surface*). O processo de otimização, inerente à seleção dos elementos melhores adaptados, aumenta a qualidade média das soluções obtidas ao longo das gerações. Entretanto, o grande potencial dos algoritmos baseados em computação evolutiva surge da integração da ampla exploração do espaço de busca com um processo de busca mais localizada, denominado exploração. Essa integração resulta num alto grau de robustez, que permite aplicar a computação evolutiva em vários problemas práticos, junto aos quais outras estratégias de solução se mostram inócuas (Bäck *et al.*, 2000a; Bäck *et al.*, 2000b; De Castro, 2001).

Neste trabalho, é apresentado o problema da árvore geradora mínima com parâmetros *fuzzy*, a complexidade do problema em questão e uma maneira de resolvê-lo baseando-se em conceitos da teoria de possibilidade (Zadeh, 1978). É proposto um sistema imunológico artificial cujo objetivo é encontrar um conjunto-solução aproximado para o problema, visando contornar a questão da complexidade.

Na Seção 2 é estabelecido o problema da árvore geradora mínima com parâmetros *fuzzy*. Na Seção 3 é feita uma breve revisão bibliográfica sobre o assunto. Na Seção 4 são apresentados os conceitos necessários sobre a teoria da possibilidade. Na Seção 5 são apresentados os principais pontos de motivação que foram encontrados para resolver o problema, bem como a escolha do método de resolução adotado. Na Seção 6, um sistema imunológico artificial é proposto para uma aproximação do problema procurando contornar a questão da complexidade. Na Seção 7 são demonstrados experimentos e seus resultados. Na Seção 8 são apresentadas às conclusões sobre o trabalho.

2. Apresentação do Problema

Considere um grafo não direcionado $G : (N,A)$ no qual cada aresta $(u,v) \in A$ tem um custo $c(u,v)$ associado. Deseja-se encontrar um subconjunto $T^* \subseteq A$ que conecta todos os nós de G sem formar ciclos e que a soma total dos seus custos:

$$c(T^*) = \sum_{(u,v) \in T^*} c(u,v)$$

é mínima. Como T^* é acíclica e conecta todos os nós de G , T^* forma uma árvore geradora de G . A árvore geradora T^* encontrada desta forma é uma árvore geradora mínima de G (Ahuja *et al.*, 1993; Goldbarg & Luna, 2000).

Na Figura 1 é apresentado um exemplo de um grafo *crisp* (que não há incertezas) e de sua árvore geradora mínima (T^*).

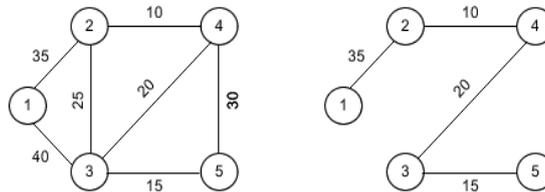


Figura 1 – Exemplo de um problema de árvore geradora mínima em um grafo *crisp*.

A Figura 2 exemplifica um grafo que tem uma estrutura bem conhecida, mas incerteza nos parâmetros associados às arestas (os parâmetros são *fuzzy*). Esses parâmetros são incertos e os seus valores podem ser associados a funções de pertinência, como mostrado na Figura 3.

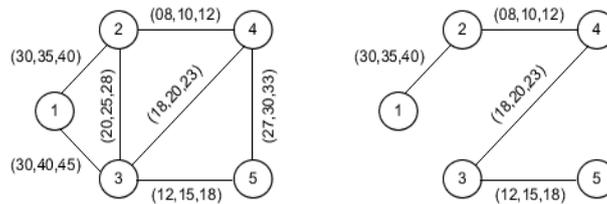


Figura 2 – Exemplo de um problema de árvore geradora mínima em um grafo com parâmetros *fuzzy*.

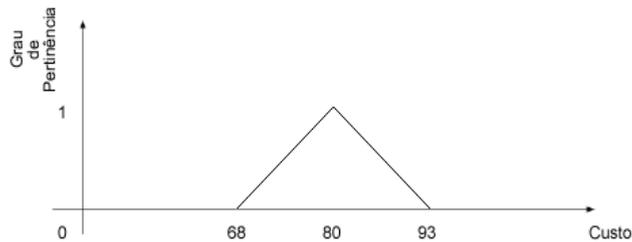


Figura 3 – Função de pertinência associada ao custo total da árvore geradora mínima do grafo da Figura 2.

Para o grafo *crisp* da Figura 1 foi encontrada uma árvore geradora mínima com custo total igual a 80. Para o grafo *fuzzy* da Figura 2 foi obtida uma árvore geradora mínima com custo *fuzzy* igual a (68,80,93) representado pela Figura 3.

Os custos associados as arestas do grafo da Figura 2 são representados por números *fuzzy*. Neste caso, como pode ser observado na função de pertinência da Figura 3, os números *fuzzy* estão representados por uma função triangular, e assim, estes são definidos como “números triangulares *fuzzy*” do tipo (c_i, c_m, c_s) , no qual c_i equivale ao custo (limitante) inferior, c_m ao custo modal, aquele com maior grau de pertinência, e c_s ao custo (limitante) superior. Existem várias formas de representar uma função de pertinência, sendo que, as mais utilizadas na literatura são: Gaussianas, Triangulares e Trapezoidais (Gomide & Pedrycz, 1998).

Considere as árvores geradoras mostradas na Figura 4:

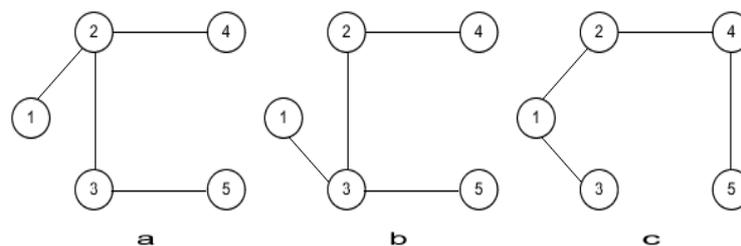


Figura 4 – Árvores geradoras obtidas do grafo da Figura 2.

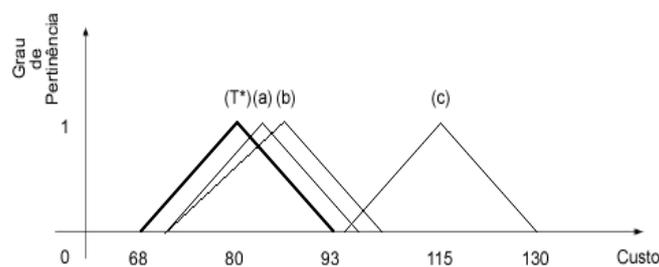


Figura 5 – Comparação dos comprimentos *fuzzy*.

a árvore (a) tem comprimento (70,85,98), (b) tem comprimento (70,90,103) e (c) (95,115,130). Suas funções de pertinência podem ser vistas na Figura 5.

Observando a Figura 5, percebe-se que não há possibilidade da árvore geradora (c) ser menor que a solução encontrada previamente na Figura 2 (T^*). Já para as árvores (a) e (b) existe uma possibilidade de serem menores. Portanto, dizemos que existe um grau de possibilidade de uma árvore pertencer ao conjunto solução do problema. Blue *et al.* (2002) entende esse valor como sendo o grau de pertinência desta solução no conjunto *fuzzy* de soluções. Com isso, fica evidente que uma boa solução para o problema não deve ser formada apenas pela árvore com maior grau de possibilidade de ser a melhor, e sim, por um conjunto-solução composto por árvores que possuem um alto grau de possibilidade de ser a solução do problema.

3. Revisão Bibliográfica

Os trabalhos mais relevantes encontrados na literatura que tratam do problema da árvore geradora mínima *fuzzy* são:

- Chunde (1996) estudou casos de grafos cuja estrutura é constituída por nós *crisp* e arestas associadas a graus de pertinência. São propostos três algoritmos considerando algumas situações da rede, sem considerar o custo das arestas como números *fuzzy*.
- Chang & Lee (1999) analisaram os problemas da árvore geradora mínima, PERT e de caminho mínimo. Eles utilizaram algoritmos clássicos e propuseram um método (OERI), para o ordenamento de números *fuzzy* em problemas de redes. Este resolve o problema utilizando um algoritmo *crisp* para obter a solução.
- Katagiri & Ishii (2000) e Katagiri *et al.* (2004a,b) realizaram um estudo sobre o problema da árvore geradora mínima cujos custos são representados por números *fuzzy* aleatórios. Neste trabalho, os autores consideraram que algumas arestas possuem “gargalos”, ou seja, valores máximos de fluxo (ou custo / demanda) associado a cada aresta. Os métodos de resoluções adotados são baseados em medidas de necessidade e de possibilidade entre os números *fuzzy*.
- Takahashi (2004) desenvolveu um estudo sobre o problema da árvore geradora mínima com estrutura *crisp* e parâmetros *fuzzy*. Sua proposta baseia-se na teoria de possibilidade (Zadeh, 1978), de forma a encontrar o conjunto solução *fuzzy* do problema. Com o objetivo de explorar a interatividade entre as arestas do grafo, foi utilizada a abordagem de Okada (2001) para calcular o grau de possibilidade que cada aresta tem de participar da solução.
- Em Almeida *et al.* (2005a,b) foi realizado um estudo sobre o problema da árvore geradora mínima com parâmetros *fuzzy* e proposto um algoritmo genético com características adaptadas à obtenção do conjunto *fuzzy* de soluções. Nesse trabalho, o grau de possibilidade é calculado utilizando a abordagem de Dubois & Prade (1980) de forma a encontrar o conjunto-solução composto por árvores que tenham alto grau de possibilidade de ser a melhor solução.

4. Conceitos Teóricos

Seja um grafo $G : (N, A)$ no qual a cada aresta $ij \in A$ está associado um custo representado por um número *fuzzy* $c_{ij} \in \mathfrak{R}^n$. Considere duas árvores geradoras T^1 e T^2 , $T^1 \neq T^2$, $T^1 < T^2$. O grau de possibilidade de T^1 ser menor do que T^2 pode ser obtido pela equação definida por Dubois & Prade (1980):

$$w = Poss\left(\sum_{ij \in T^1} c_{ij} \leq \sum_{ij \in T^2} c_{ij}\right) = \sup_{u \leq v} \min\{\mu_{T^1}(u), \mu_{T^2}(v)\}$$

sendo:

- $Poss$: medida de possibilidade;
- μ_{T^1} e μ_{T^2} : as funções de pertinência dos custos totais das árvores T^1 e T^2 respectivamente;

- *sup min*: o valor supremo da t-norma (neste caso, a t-norma escolhida foi o mínimo), ou seja, o maior valor que pode ser obtido no conjunto formado pela intersecção das funções de pertinência μ_{T^1} e μ_{T^2} .

Considerando T^* a árvore geradora com maior grau de possibilidade de ser a mínima, para encontrar o conjunto das árvores que tem alto grau de possibilidade de ter custo menor ou igual ao de T^* é necessário encontrar todas as árvores geradoras $T^k \in T$, e compará-las com T^* . Fazendo isso, obtém-se o grau de possibilidade dado pela equação (Okada, 2001):

$$D_T = \min_{T^k \in T} \{ Poss(\sum_{ij \in T^k} c_{ij} \leq \sum_{ij \in T^*} c_{ij}) \}$$

Isso torna o problema de difícil solução, pois além de ter que enumerar todas as soluções, a comparação entre elas torna o problema *NP-Completo*.

Proposição 1 de Takahashi (2004) tem-se que, dados dois números triangulares fuzzy, normalizados, $a = (a_i, a_m, a_s)$ e $b = (b_i, b_m, b_s)$. Se $a_m \leq b_m$, então $Poss(a \leq b) = 1$.

Prova: utilizando a definição do grau de possibilidade entre dois números fuzzy:

$$Poss(a \leq b) = \sup_{T^k \in T} \{ \min\{\mu_a(u), \mu_b(v)\} \}$$

Supondo que $u = a_m$ e $v = b_m$, então, se $a_m \leq b_m$, tem-se que $u \leq v$. Sendo os números fuzzy normalizados:

$$\mu_a(u) = \mu_b(v) = 1 \Rightarrow \min\{\mu_a(u), \mu_b(v)\} = 1$$

portanto, $Poss(a \leq b) = 1$. ■

Com isso, se for encontrada a árvore com custo de menor valor modal (a_m), será encontrada a árvore geradora com maior grau de possibilidade de ser a árvore geradora mínima fuzzy. Essa pode ser obtida pelo grafo crisp G_m , com custo das arestas igual ao custo modal $c'_{ij} = (c_m)_{ij}$.

Proposição 2 de Takahashi (2004) tem-se que, T^0 é uma árvore geradora mínima do grafo crisp G_m , com os valores modais (c_m) de G sendo os custos associados às arestas do grafo crisp. Seja a árvore geradora fuzzy T^* , que possui as mesmas arestas de T^0 , com custo igual a $c^* = \sum_{ij \in T^*} c_{ij}$. Se T^* for a solução ótima do problema, então $w = Poss(\sum_{ij \in T^*} c_{ij} \leq \sum_{ij \in T^k} c_{ij}) = 1$,

sendo T^k qualquer árvore geradora fuzzy de G .

Prova: suponha que exista um T^k tal que $w = Poss(\sum_{ij \in T^*} c_{ij} \leq \sum_{ij \in T^k} c_{ij}) < 1$. Pela proposição 1,

temos que $c^* > c^k$. Porém, como c^* é o menor valor encontrado para o grafo G_m , temos que $c^* \leq c^k$, gerando uma contradição. Portanto:

$$w = Poss(\sum_{ij \in T^*} c_{ij} \leq \sum_{ij \in T^k} c_{ij}) = 1 \quad \blacksquare$$

5. Motivação

Encontrar a árvore geradora mínima é um problema cujos métodos envolvem busca em um espaço que cresce conforme se aumenta o número de nós. Além disso, quando se trabalha com grafos cujas arestas possuem custos *fuzzy* a complexidade aumenta, pois o fato de ter que escolher os subconjuntos de arestas a serem retiradas e os grupos que podem substituir cada um dos subconjuntos, faz com que o problema seja combinatorial.

O número de árvores geradoras em um grafo completo é da ordem de m^{m-2} , $m = |N|$ (Raidl & Julstrom, 2001). Por exemplo, de um grafo completo com 50 nós é possível obter $50^{48} \approx 3,55 \times 10^{81}$ árvores possíveis. Uma busca exaustiva é inviável, pois supondo que 1000 computadores analisem 1000 árvores por segundo cada um, levariam aproximadamente 10^{68} anos para analisar todas as árvores possíveis (estima-se que o universo tenha 10^{13} anos). Este é um espaço de busca grande por qualquer parâmetro, mesmo para um problema pequeno. Um problema com 50 nós é pequeno se comparado a casos reais, como o Brasil, por exemplo, que possui mais de 5000 cidades.

Considerando que a solução é composta pelo conjunto de árvores com alto grau de possibilidade de serem menores ou iguais a T^* , e que ainda não existe nenhum método eficiente ou específico para solucionar este tipo de problema, pode-se propor uma heurística para encontrar uma solução aproximada satisfatória tentando contornar a questão da complexidade.

Entre as possíveis heurísticas, estão aquelas que imitam processos naturais em busca de equilíbrio, em geral otimizado, como os sistemas imunológicos artificiais, algoritmos genéticos, estratégias evolutivas, etc.

6. Sistema Imunológico Artificial

6.1 Introdução

A simulação de processos evolutivos naturais objetivando resolver problemas de explosão combinatoria e multimodais demonstrou ser uma estratégia eficaz e robusta. A otimização do comportamento de um sistema, obtida através da simulação de processos evolutivos, representa uma abordagem poderosa para aprendizagem de máquina e estudo de fenômenos auto-organizados. A implementação computacional desses métodos de simulação da evolução, chamada computação evolutiva, possibilita a determinação de soluções para várias classes de problemas (Bäck *et al.*, 2000a,b). Além dessas, outras linhas de pesquisa ainda mais recentes têm surgido como novos paradigmas de computação.

Os sistemas imunológicos artificiais (SIA) (Dasgupta, 1998a), que surgiram a partir de tentativas de modelar e aplicar princípios imunológicos no desenvolvimento de novas ferramentas computacionais, já vêm sendo utilizados em diversas áreas, como reconhecimento de padrões, detecção de falhas e anomalias, segurança computacional, otimização, controle, robótica, *scheduling*, análise de dados, aprendizagem de máquina, dentre outras, como pode ser encontrado em Dasgupta (1998a,b), Bäck *et al.* (2000a,b), Timmis (2000) e em De Castro (2001).

Nas áreas de engenharia e computação, tem surgido um forte interesse pelo estudo dos sistemas imunológicos devido, principalmente, à sua capacidade de processamento de

informação. Sob uma perspectiva de engenharia, existem diversas características do sistema imunológico (SI) que podem ser destacadas:

- **Unicidade:** cada animal possui seu próprio sistema imunológico, com suas capacidades e vulnerabilidades particulares;
- **Reconhecimento de padrões internos e externos ao sistema:** as células e moléculas que não pertencem ao organismo são reconhecidas e eliminadas pelo SI;
- **Deteção de anomalia:** o SI pode detectar e reagir a agentes patogênicos (causadores de anomalias) a que o organismo nunca havia sido exposto anteriormente;
- **Deteção imperfeita (tolerância a ruídos):** um reconhecimento perfeito não é necessário para que o SI reaja contra um elemento causador de patologia (patógeno);
- **Diversidade:** existe uma quantidade limitada de células e moléculas no SI que são utilizadas para se obter o reconhecimento de um número praticamente infinito de elementos, incluindo aqueles sintetizados em laboratório;
- **Aprendizagem por reforço:** a cada encontro com o mesmo patógeno, o sistema imunológico melhora a qualidade de sua resposta; e
- **Memória:** os componentes do SI bem sucedidos no reconhecimento e combate às patologias são armazenados para uma resposta futura mais intensa e efetiva.

6.2 Definição

Segundo Dasgupta (1998a), os sistemas imunológicos artificiais são mecanismos computacionais compostos por metodologias inteligentes, inspiradas no sistema imunológico biológico, para a solução de problemas do mundo real.

6.3 Princípios Fundamentais do Sistema Imunológico

Para facilitar a compreensão da estrutura do sistema imunológico artificial serão apresentados, de forma sucinta, o funcionamento do sistema imunológico nos animais vertebrados e os princípios da seleção clonal. Informações referentes aos elementos que constituem o sistema, bem como definições sobre os termos biológicos utilizados podem ser encontrados nas referências: De Castro (2001), Dreher (1995), Janeway *et al.* (2000) e Timmis (2000).

6.3.1 Composição e Funcionamento do Sistema Imunológico

Quando um sistema (corpo) detecta a presença de um antígeno (patógeno), ocorrem duas formas de respostas imunológicas dadas pelos seguintes sistemas (Figura 6):

- **Sistema Imune Inato:** é a primeira linha de defesa do organismo. Ele é formado por células fagocitárias, como macrófagos e os neutrófilos, além de fatores solúveis como o complemento e algumas enzimas. As células do sistema imune inato desempenham um papel crucial na iniciação e posterior direcionamento das respostas imunes adaptativas, principalmente devido ao fato de que as respostas adaptativas demoram período de tempo (da ordem de dias) para exercer seus efeitos. Portanto, a resposta inata apresenta um papel muito importante no controle das infecções durante esse tempo;

- **Sistema Imune Adaptativo:** os linfócitos (produzidos na medula óssea e no timo) são as principais células que compõem esse sistema, presentes apenas nos animais vertebrados, eles desempenham um papel crucial no desencadeamento e posterior regulação das respostas imunes inatas. Cada “linfócito virgem” que penetra na corrente circulatória é portador de receptores de um antígeno com uma única especificidade. Os linfócitos sofrem, então, um processo parecido com a seleção natural durante a vida do indivíduo: somente aqueles que encontram um antígeno com o qual seu receptor pode interagir serão ativados para proliferar e se diferenciar em células efetoras. Após a ligação do anticorpo de superfície ao antígeno, a célula é ativada para proliferar e produzir uma numerosa prole, conhecida como clone. Essas células secretam anticorpos com uma especificidade idêntica à do receptor de superfície. Esse princípio recebeu o nome de teoria da seleção clonal, e constitui a parte central da imunidade adaptativa.

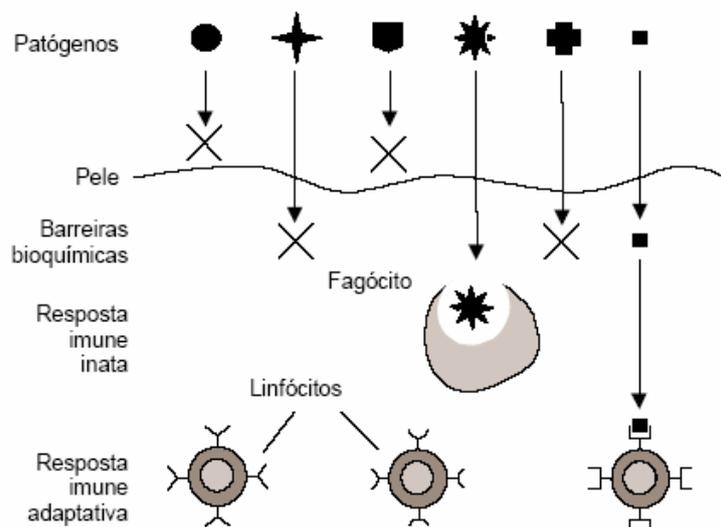


Figura 6 – Estrutura multicamadas do sistema imunológico.

6.4 Princípios da Seleção Clonal

Uma vez que cada célula apresenta um padrão (forma) distinto de receptor antigênico, o número de linfócitos que pode se ligar a um determinado antígeno é restrito. A fim de produzir células efetoras específicas em quantidade suficiente para combater uma infecção, um linfócito ativado deve se proliferar antes que sua prole se diferencie em células efetoras.

O princípio da seleção clonal estabelece que apenas aquela célula capaz de reconhecer um determinado estímulo antigênico irá se proliferar, sendo, portanto, selecionada em detrimento das outras.

Quando um animal é exposto a um antígeno, uma subpopulação de linfócitos responde através da produção de anticorpos. Cada célula secreta um único tipo de anticorpo, que é relativamente específico para o antígeno. Através da ligação do antígeno com o receptor do linfócito e, dado um segundo sinal (ou sinal co-estimulatório) de células acessórias como a

célula T_H , um antígeno estimula o linfócito a se proliferar (dividir) e transformar-se em uma célula terminal capaz de secretar anticorpos em altas taxas. Estas células são chamadas de plasmócitos.

Os linfócitos, além de se proliferar e diferenciar em plasmócitos, também podem se diferenciar em linfócitos de memória, caracterizadas por longos períodos de vida. Esses linfócitos de memória provavelmente não produzem anticorpos, mas quando re-expostas ao mesmo estímulo antigênico começam a se diferenciar em plasmócitos capazes de produzir anticorpos pré-selecionados pelo antígeno específico que estimulou a resposta primária.

6.5 Implementação de um Sistema Imunológico Artificial

Seja dado um grafo com estrutura *crisp* e arestas com custos representados por números triangulares *fuzzy*, deseja-se encontrar um conjunto de árvores geradoras (T) que tenham alto grau de possibilidade de terem seus custos menores ou iguais ao custo da árvore geradora mínima (T^*) associada ao grafo com parâmetros *crisp* (G_m).

Este tipo de problema possui as seguintes características:

- um conjunto *fuzzy* de soluções em que cada árvore geradora (T) possui um grau de possibilidade de ter seu custo menor ou igual ao custo da árvore geradora mínima (T^*);
- necessidade de avaliar uma grande quantidade de árvores geradoras para a construção do conjunto *fuzzy* de soluções.

Essas características nos levaram a desenvolver um sistema imunológico artificial (SIA) nos moldes propostos por Dasgupta, (1998a,b) e De Castro (2001) para tentar encontrar um conjunto-solução para o problema proposto. A população é formada por anticorpos (árvores geradoras) que geram novos indivíduos herdeiros das características dos anticorpos-pai (clones) que evoluem por meio de operadores de maturação. Procurando um subconjunto *fuzzy* de soluções, a medida de afinidade (também conhecida como função de *fitness*) é utilizada de forma a privilegiar a reprodução dos indivíduos mais adaptados ao ambiente (seleção clonal), ou seja, somente aqueles anticorpos que encontram um antígeno com o qual seu receptor pode interagir serão ativados para proliferar e se diferenciar em células efectoras. No caso, a medida de afinidade de cada indivíduo é dada pelo grau de possibilidade da árvore geradora (T) ter custo menor ou igual ao da árvore geradora mínima (T^*) associada ao grafo *crisp* (G_m).

6.5.1 A Estrutura e o Algoritmo

Para facilitar a compreensão do algoritmo, considere o diagrama de blocos da Figura 7.

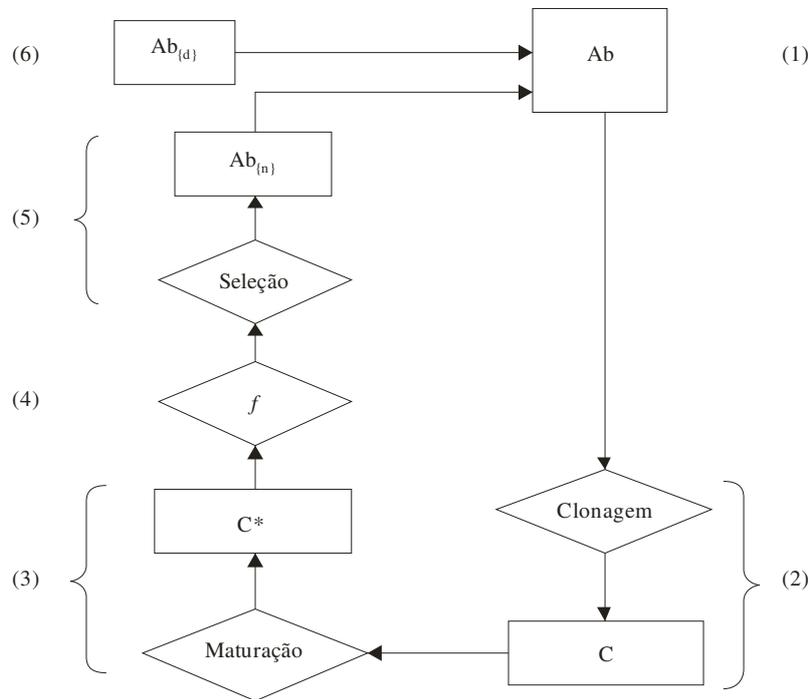


Figura 7 – Diagrama de blocos do Sistema Imunológico Artificial.

A implementação computacional do SIA para encontrar o conjunto-solução pode ser feita utilizando o pseudocódigo descrito no Quadro I, dados os parâmetros de entrada e saída.

Pseudocódigo:

Entradas:

- Grafo $G : (N,A)$ com custos triangulares fuzzy;
- Tamanho p da população de anticorpos;
- Tempo limite t para execução do algoritmo;
- Número máximo de gerações gen ;
- Limiar l para injeção de diversidade populacional;
- Grau de afinidade mínimo gam entre o anticorpo e o antígeno necessário para gerar descendentes;
- Número s de melhores anticorpos encontrados que irão formar o conjunto-solução;
- Número n de clones a serem gerados por cada anticorpo-pai em cada geração.

Saídas:

- Matriz \mathbf{Am} correspondente ao conjunto-solução formado pelos melhores anticorpos;
- Afinidade (fitness) fm de cada anticorpo da matriz \mathbf{Am} .

```

T* ← encontrar a árvore geradora mínima crisp de G;
Ab ← inicializar população de anticorpos com tamanho p;
f ← afinidade(Ab);
geração ← 1;
enquanto tempo <= t e geração <= gen faça
  para cada anticorpo i de Ab faça
    se afinidade(Ab(i)) >= gam então
      C ← clonar(Ab(i), n);
      C* ← maturar(C);
      f ← afinidade(C*);
      Ab[n] ← selecionar melhor clone de C*;
      se Ab[n] tiver maior afinidade que Ab(i) então
        Ab(i) ← Ab[n]; //substitui o pai pelo clone
      fim-se;
    fim-para;
  Atualizar Am e fm com os s melhores anticorpos de Ab;
  se diversidade de Ab < l então
    I ← selecionar indivíduos semelhantes de Ab;
    Ab[d] ← maturar(I);
    Ab ← substituir os indivíduos ∈ I de Ab por Ab[d]
  fim-se;
  geração ← geração + 1;
fim_enquanto;

```

Quadro I – Pseudocódigo do Sistema Imunológico Artificial.

Representação da População de Anticorpos (Ab): como descrito anteriormente, em um grafo $G : (N, A)$, sendo $m = |N|$, uma árvore geradora tem $m - 1$ arestas que conectam todos os nós e não formam ciclos (Ahuja *et al.*, 1993; Bazaraa *et al.*, 1990). Assim sendo, cada anticorpo é representado por um vetor-linha com $m - 1$ posições de números inteiros no qual cada posição armazena um índice que representa uma aresta do grafo. Desta forma, cada anticorpo representa uma árvore geradora. A Figura 8 mostra um exemplo de anticorpo.

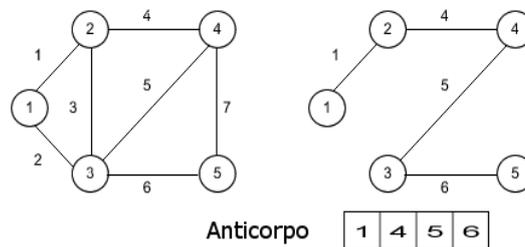


Figura 8 – Exemplo de anticorpo representando uma árvore.

A população de anticorpos (**Ab**) (Tabela 1) é formada por árvores geradoras. Cada posição **Ab**(*i*) corresponde a uma árvore geradora composta pelo índice das arestas (**Ab**^{*p*}_{*q*}).

Veja o exemplo seguinte:

dado o grafo $G : (N,A)$ da Figura 8, sua representação é dada pela Tabela 1:

Tabela 1 – Representação do Grafo $G : (N,A)$ da Figura 8.

Índice	Nó <i>i</i>	Nó <i>f</i>
1	1	2
2	1	3
3	2	3
4	2	4
5	3	4
6	3	5
7	4	5

Uma população (Tabela 2) deve ser composta por *n* anticorpos, no qual cada um deles corresponde a uma árvore geradora.

Tabela 2 – População de anticorpos (árvores geradoras).

Anticorpo **Ab**(*I*)

1	4	5	6
1	3	4	7
...
2	3	6	7

O anticorpo **Ab**¹ que representa a árvore da Figura 8 é formado pelos índices [1,4,5,6] do grafo $G : (N,A)$ (Tabela 1). Isso significa que esse anticorpo representa a árvore formada pelas arestas (1,2)(1,3)(2,4)(3,4)(3,5).

1) Inicialização: a inicialização da população é feita por meio de uma busca local ao redor da árvore-base (T^*). O operador de maturação (ver item número 3, Figura 9) é aplicado em T^* para gerar novos indivíduos. Dessa forma, *n* arestas de T^* são selecionadas e substituídas por outras arestas pertencentes ao conjunto de corte para formar uma nova árvore. Esse procedimento é repetido *p* vezes, até que a população inicial seja totalmente preenchida. Fazendo dessa maneira, espera-se aumentar o desempenho do algoritmo, pois árvores que compartilham um maior número de arestas de T^* possuem maior grau de possibilidade de pertencer ao conjunto solução. Logo, como a população inicial geralmente é preenchida com árvores que possuem alto grau de possibilidade, acelera-se a convergência do algoritmo.

2) Clonagem: a clonagem da população é feita gerando-se *n* clones idênticos de cada indivíduo da população. Esses anticorpos clonados irão formar a população **C** de clones.

3) Maturação: a população **C** de clones sofre um processo de maturação de afinidade (mutação) gerando uma nova população **C*** de clones maturados. Essa mutação ocorre sorteando-se *n* arestas que compõem o anticorpo e trocando-as por outras que pertençam ao conjunto de corte como mostrado na Figura 9.

4) Medida de Afinidade (*fitness*): o grau de afinidade do indivíduo (*i*) é dado pelo valor de possibilidade da árvore (anticorpo) em questão (T^k) ter custo total menor ou igual à árvore geradora mínima de maior grau de possibilidade (T^* – anticorpo com maior afinidade), conforme descrito na Seção 4 e obtido pela seguinte equação:

$$fitness(i) = Poss\left(\sum_{ij \in T^k} c_{ij} \leq \sum_{ij \in T^*} c_{ij}\right)$$

5) Seleção: o processo de seleção dos clones é inspirado nos princípios da seleção clonal. Portanto, o método utilizado é elitista (Michalewicz, 1996), pois privilegia o clone com maior grau de afinidade. Assim, o clone com maior *fitness* é selecionado entre os indivíduos da população de clones e comparado com o seu anticorpo-pai e apenas o melhor indivíduo deve ser preservado na população.

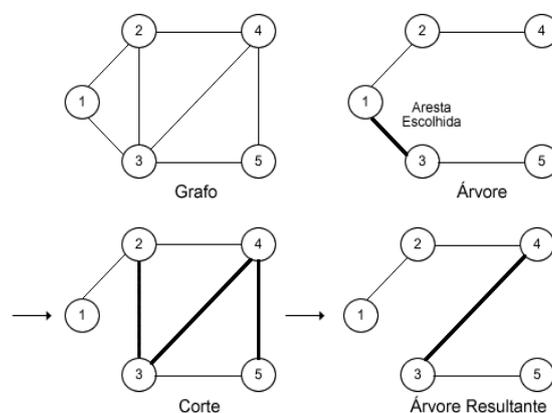


Figura 9 – Exemplo de maturação de um anticorpo (árvore).

6) Diversidade: a chance de obter anticorpos com alto grau de afinidade aumenta em uma população de soluções mais diversificadas. Para estudar o comportamento da diversidade da população, foi definida uma medida:

Diversidade populacional é a razão entre o número de indivíduos únicos (aqueles que não se repetem na população) e o tamanho da população.

Uma possível injeção de diversidade pode ocorrer em momentos pré-determinados, tais como:

- quando a variação da média de *fitness* (medida de afinidade) for menor que uma determinada taxa;
- quando a taxa de diversidade for menor que um determinado limiar.

Quando um desses fatores ocorrer significa que a população está se tornando homogênea prejudicando o desempenho do algoritmo. Neste caso, há a necessidade de se fazer uma injeção de diversidade populacional. Isso pode ser feito de diversas maneiras, como:

- cruzar elementos da população para gerar novos indivíduos;
- aplicar mutação em indivíduos semelhantes;
- retirar uma parcela de anticorpos da população e preenchê-la com novos indivíduos.

7. Resultados Experimentais

Dois grafos foram utilizados para avaliar o desempenho do Sistema Imunológico Artificial (SIA), e os resultados obtidos foram confrontados com os obtidos pelo Algoritmo Genético (AG) proposto em Almeida *et al.* (2005a) e em Almeida *et al.* (2005b).

Para cada grafo foram realizadas 10 rodadas. Cada uma dessas rodadas resulta em um conjunto-solução formado pelos n melhores anticorpos (árvores) encontrados, organizados em ordem crescente de grau de possibilidade (*fitness*). Desta forma, o conjunto-solução final é formado pelos valores máximos de cada linha dos conjuntos-soluções encontrados em cada rodada, ou seja, avalia-se a primeira linha de todos os conjuntos-soluções obtidos e armazena-se o maior valor (o melhor anticorpo daquela posição) na primeira linha do conjunto-solução final e assim sucessivamente até a última linha. Fazendo assim, no final do processo obtém-se um conjunto-solução final formado pelos melhores indivíduos de cada posição, por exemplo, o indivíduo (i) (linha i) da solução final será o melhor indivíduo da i -ésima linha dentre todas as soluções obtidas em cada rodada.

Devido ao fato do SIA trabalhar com múltiplas populações (populações de clones), ele consome mais tempo de processamento que o AG para realizar cada experimento (mediante a igualdade dos parâmetros de número de gerações e do tamanho da população). Então, a fim de obter uma comparação mais justa, cada experimento do AG foi testado utilizando o mesmo tempo de processamento consumido pelo SIA.

Os seguintes parâmetros foram utilizados para realizar os experimentos:

Parâmetros do SIA:

- (Critério de parada) Número máximo de gerações: 50;
- Tamanho da população de anticorpos: 50;
- Grau de afinidade para geração de clones: 50%;
- Número de clones gerados por anticorpo: 10;
- Taxa mínima de diversidade populacional: 50%;
- Número de indivíduos armazenados no conjunto-solução: 10.

Parâmetros do AG:

- (Critério de parada) Tempo consumido pelo SIA;
- Tamanho da população de cromossomos: 50;
- Taxa de cruzamento: 50%;
- Taxa de mutação: 3%;
- Preservação: melhor e pior indivíduos;
- Número de melhores cromossomos armazenados no conjunto-solução: 10.

Esses parâmetros foram calibrados exaustivamente de forma a obter o melhor desempenho para os dois métodos utilizados.

Quanto à implementação dos algoritmos e à realização dos testes foi utilizada a ferramenta MatLab 7 na seguinte plataforma: Processador Athlon XP 2200 – 2.0 GHz – 256 Mb.

7.1 Experimento I – BAYS29

Para o primeiro experimento foi utilizada a instância Bays29 retirada da base de dados *Traveling Salesman Problem Library* (TSPLIB). O TSPLIB é um repositório público de instâncias para o problema do caixeiro viajante na internet onde, além das instâncias, consta também o ótimo global para algumas delas. Ele está disponível em: <http://www.iwr.uni-heidelberg.de/groups/comopt/soft/TSPLIB95/TSPLIB.html>.

O grafo em questão representa as possíveis conexões existentes entre 29 cidades do estado da Bavária – Alemanha, com as respectivas distâncias geográficas entre elas. O custo de cada aresta foi estabelecido como um número triangular *fuzzy* com valor modal igual à distância de uma cidade a outra e desvio de 10% para os limitantes inferiores e superiores. Esse desvio corresponde à incerteza em relação aos custos associados às arestas do grafo, e como é aplicado o mesmo valor para todos eles, fica caracterizado que a incerteza é expressada de uma forma geral em relação às arestas. Quando um desvio diferente é calculado para cada custo, diz-se que existe uma incerteza diferente associada a cada aresta do grafo.

O grafo em questão é completo, formado por 29 nós e 406 arestas. Ele possui em torno de $3,05 \times 10^{39}$ candidatos a solução e a sua principal característica é conter diversas árvores com grau de possibilidade igual a 1. Este fato influencia diretamente na diversidade da população, aumentando a possibilidade dela se tornar homogênea com o passar das gerações, pois os indivíduos mais adaptados ao ambiente tendem a sobreviver em todas as gerações, além de possuírem chances maiores de gerar descendentes.

Os resultados encontrados pelos algoritmos estão exibidos na Tabela 3.

A convergência e o comportamento da diversidade populacional foram observados em cada um dos testes tanto para o SIA (Figura 10) quanto para o AG (Figura 11).

Tabela 3 – Comparação dos resultados obtidos pelos métodos para o Experimento I.

Nº Sol.	SIA <i>Fitness</i>	SIA <i>Custo Fuzzy</i>	AG <i>Fitness</i>	AG <i>Custo Fuzzy</i>
1	0,9712	[1409,40 1566,00 1722,60]	0,9936	[1403,10 1559,00 1714,90]
2	0,9872	[1404,90 1561,00 1717,10]	0,9968	[1402,20 1558,00 1713,80]
3	1,0000	[1401,30 1557,00 1712,70]	0,9968	[1402,20 1558,00 1713,80]
4	1,0000	[1401,30 1557,00 1712,70]	0,9968	[1402,20 1558,00 1713,80]
5	1,0000	[1401,30 1557,00 1712,70]	1,0000	[1401,30 1557,00 1712,70]
6	1,0000	[1401,30 1557,00 1712,70]	1,0000	[1401,30 1557,00 1712,70]
7	1,0000	[1401,30 1557,00 1712,70]	1,0000	[1401,30 1557,00 1712,70]
8	1,0000	[1401,30 1557,00 1712,70]	1,0000	[1401,30 1557,00 1712,70]
9	1,0000	[1401,30 1557,00 1712,70]	1,0000	[1401,30 1557,00 1712,70]
10	1,0000	[1401,30 1557,00 1712,70]	1,0000	[1401,30 1557,00 1712,70]

Ambos os algoritmos conseguiram encontrar boas soluções (árvores), pois todas elas possuem alto grau de possibilidade de serem melhores que T^* . No entanto, por uma pequena diferença, o SIA obteve um desempenho melhor, pois enquanto que este encontrou 8 árvores com grau máximo de possibilidade, o AG encontrou 6. Essa diferença pode ser explicada

pelo fato do SIA trabalhar com múltiplas populações (clones), o que faz com que ele consiga manter a diversidade da população (Figura 10) de maneira mais eficaz que o AG (Figura 11), permitindo-lhe obter um melhor desempenho em problemas multimodais. No AG, o melhor indivíduo tem uma grande chance de sobreviver em todas as gerações, e durante essas gerações, esse indivíduo tende a transmitir parte do seu código genético para outros indivíduos da população, fazendo com que a diversidade reduza gradativamente. Por outro lado, o SIA mantém populações paralelas de anticorpos, e permite que os pais transmitam os seus códigos genéticos somente para os seus clones, visando manter a diversidade da população. A diferença de desempenho na manutenção da diversidade populacional entre os dois métodos é clara, pois analisando esse experimento, nota-se que enquanto que o AG manteve, em média, 38% de indivíduos únicos na população, o SIA conseguiu manter 96%.

O tempo total consumido para a realização de cada rodada, tanto para o AG quanto para o SIA, foi de 21 segundos, em média. Nesse tempo, enquanto o SIA executou 50 gerações, o AG executou 250. O número de indivíduos avaliados pelos algoritmos tem como limitante superior 25.000 para o SIA e 12.500 para o AG.

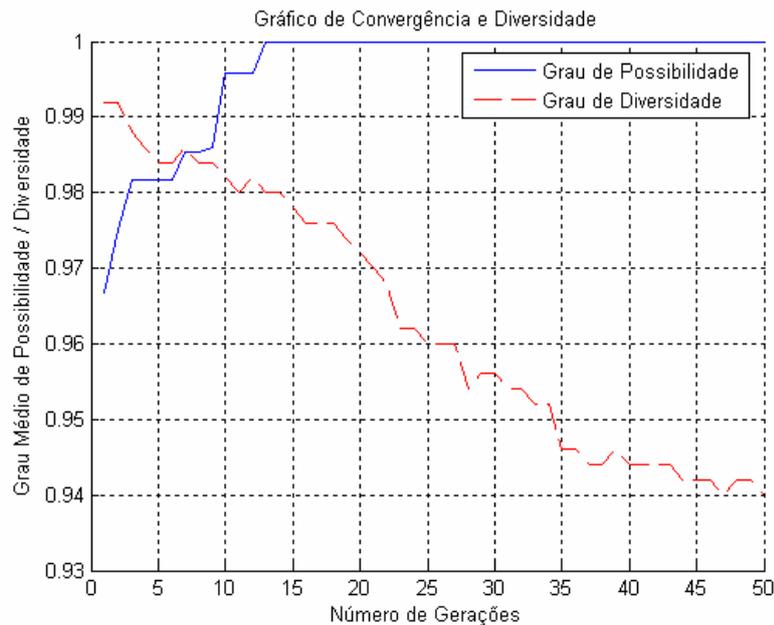


Figura 10 – Gráfico de convergência e diversidade populacional obtidos pelo SIA para o Experimento I.

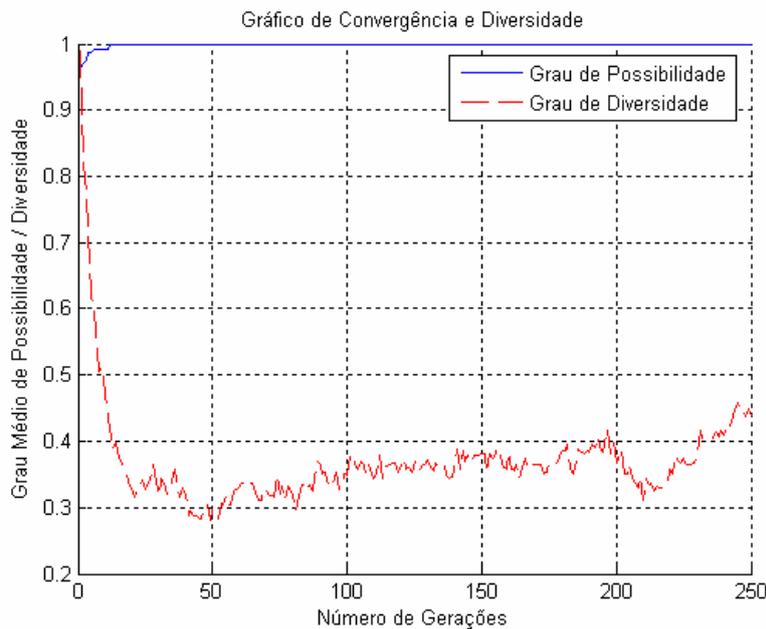


Figura 11 – Gráfico de convergência e diversidade populacional obtidos pelo AG para o Experimento I.

7.2 Experimento II – GRAFO274

Para este segundo experimento foi utilizada a instância Grafo274 retirada da base de dados *Operational Research Library* (Beasley, 1990). O OR-LIBRARY é um banco de dados público, disponível na Internet, que contém diversas instâncias de problemas clássicos na área de pesquisa operacional.

O Grafo274 é composto por 50 nós e 274 arestas. Um grafo completo com esse mesmo número de nós tem aproximadamente $3,55 \times 10^{81}$ soluções candidatas.

O custo de cada aresta foi estabelecido como um número triangular *fuzzy* do tipo (c_i, c_m, c_s) , com valor modal (c_m) igual ao custo da aresta que conecta dois nós, limitante inferior $c_i = c_m - (0,1 \times c_m) - a$ e limitante superior $c_s = c_m + (0,1 \times c_m) + a$, onde a é um número aleatório entre 1 e 5. Fazendo dessa forma, fica caracterizado que a incerteza, nesta instância, é expressada diferentemente para cada aresta do grafo.

A principal característica do Grafo274 é ser um problema multimodal que contém vários ótimos globais, ou seja, existem diversas árvores geradoras mínimas (T^*) que podem ser a solução ótima do problema, pois todas têm grau de possibilidade igual a 1. Essa característica influencia diretamente na eficiência dos métodos, pois o fato de existir diversos “picos” candidatos à solução pode induzir os indivíduos da população se aglomerarem em um pico específico fazendo com que o espaço de busca não seja explorado de forma eficiente. Assim, é importante que o algoritmo seja capaz de realizar eficientemente manutenção de diversidade populacional durante as gerações, para que o espaço de busca seja bem explorado possibilitando uma boa performance.

Os resultados encontrados estão exibidos na Tabela 4.

A convergência e o comportamento da diversidade populacional foram observados em cada um dos testes tanto para o SIA (Figura 12) quanto para o AG (Figura 13).

O tempo total consumido em cada rodada, tanto para o AG quanto para o SIA, foi de 2 minutos, em média, sendo que enquanto que o SIA executou 50 gerações, o AG executou 1220.

O número de indivíduos avaliados pelos algoritmos tem como limitante superior 25.000 para o SIA e 61.000 para o AG.

Tabela 4 – Comparação dos resultados obtidos pelos métodos para o Experimento II.

Nº Sol.	SIA <i>Fitness</i>	SIA <i>Custo Fuzzy</i>	AG <i>Fitness</i>	AG <i>Custo Fuzzy</i>
1	1,0000	[1822,00 2085,00 2355,00]	0,9981	[1822,00 2086,00 2356,00]
2	1,0000	[1823,00 2085,00 2353,00]	0,9981	[1822,00 2086,00 2354,00]
3	1,0000	[1824,00 2085,00 2352,00]	0,9981	[1822,00 2086,00 2355,00]
4	1,0000	[1821,00 2085,00 2356,00]	0,9981	[1821,00 2086,00 2355,00]
5	1,0000	[1821,00 2085,00 2355,00]	0,9981	[1820,00 2086,00 2355,00]
6	1,0000	[1821,00 2085,00 2356,00]	1,0000	[1824,00 2085,00 2352,00]
7	1,0000	[1823,00 2085,00 2354,00]	1,0000	[1824,00 2085,00 2352,00]
8	1,0000	[1821,00 2085,00 2355,00]	1,0000	[1827,00 2085,00 2350,00]
9	1,0000	[1823,00 2085,00 2353,00]	1,0000	[1823,00 2085,00 2352,00]
10	1,0000	[1822,00 2085,00 2354,00]	1,0000	[1822,00 2085,00 2353,00]

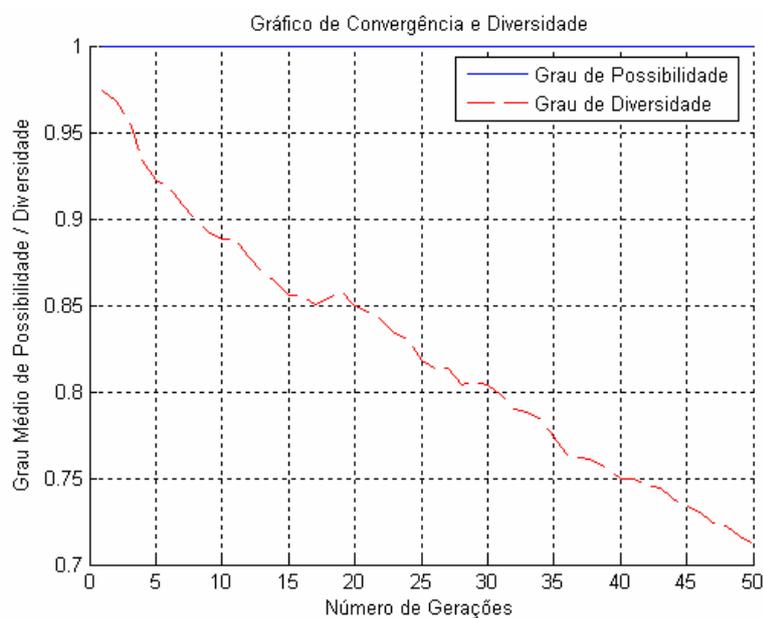


Figura 12 – Gráfico de convergência e diversidade populacional obtidos pelo SIA para o Experimento II.

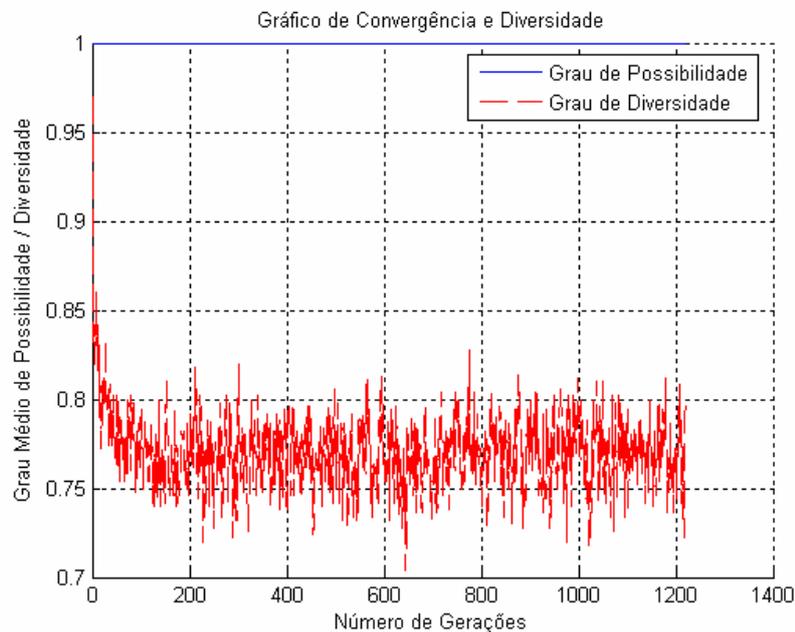


Figura 13 – Gráfico de convergência e diversidade populacional obtidos pelo AG para o Experimento II.

Fazendo uma análise de desempenho de cada algoritmo, pode-se concluir que em relação aos resultados, os dois métodos conseguiram encontrar árvores com grande chance de serem soluções ótimas, pois todas elas têm graus de possibilidade próximos ou iguais a 1.

Enquanto que, em média, 83% da população manteve-se heterogênea durante as gerações quando o SIA foi utilizado, 77% da população (em média) manteve-se distinta quando utilizado o AG. Observando os gráficos das Figura 12 e Figura 13, fica evidente que o SIA não precisou fazer injeção de diversidade em nenhum momento, enquanto que o AG necessitou fazer injeções constantemente e, ao passo que a queda da diversidade populacional ocorreu de forma lenta e gradativa quando o SIA foi utilizado, pelo AG a queda foi bastante acentuada nas primeiras gerações e manteve-se oscilando por meio de injeções constantes de diversidade durante as demais gerações.

Para esse experimento, enquanto que o AG encontrou 5 soluções com *fitness* igual a 1, o SIA conseguiu encontrar todas as 10 soluções com grau máximo de possibilidade.

8. Conclusões

Uma grande quantidade de problemas pode ser representada na forma de redes utilizando a teoria dos grafos. Essa abordagem permite obter uma melhor manipulação das entidades envolvidas, de forma que a implementação de métodos para solucioná-los torna-se mais simples e intuitiva. O problema de encontrar a árvore geradora mínima em grafos é bastante estudado dentro dessa teoria, pois o mesmo possui diversas aplicações práticas.

Freqüentemente, em casos reais, as informações associadas aos problemas possuem valores incertos (ou imprecisos), fazendo com que a teoria clássica dos conjuntos torne-se ineficaz para representá-los. Assim, é necessário utilizar uma representação mais robusta, que pode ser obtida por meio da teoria dos conjuntos nebulosos.

Neste trabalho, foi proposto um Sistema Imunológico Artificial, baseado em técnicas da computação evolutiva, para resolver o problema da árvore geradora mínima em grafos cujas arestas possuem custos *fuzzy* associados. Esse método está fundamentado na teoria da possibilidade (Zadeh, 1978) e faz uso de técnicas de seleção clonal, para explorar eficientemente o espaço de busca de soluções, de forma a obter um conjunto-solução aproximado, tentando contornar a complexidade do problema.

Enquanto que o Algoritmo Genético (AG) evolui os candidatos buscando soluções ao redor dos indivíduos mais adaptados ao ambiente, o Sistema Imunológico Artificial (SIA) evolui múltiplas populações (clones). Em geral, o SIA consegue manter um melhor controle sobre a diversidade da população, possibilitando obter maior desempenho quando aplicado em grafos que possuem diversas árvores com o mesmo custo total *fuzzy* de T^* . Isso torna o método proposto competitivo e ao mesmo tempo promissor para a resolução do problema da árvore geradora mínima com parâmetros *fuzzy*.

O método proposto pode ser adaptado para resolver outros problemas de grafos que tenham parâmetros *fuzzy*, tais como: caminho mínimo, problema do caixeiro viajante, problema de fluxo em redes, roteamento, etc.

Algumas técnicas podem ser sugeridas para serem implementadas no SIA de forma a aperfeiçoá-lo para a resolução de certos problemas, como por exemplo: flexibilizar o número de clones gerados por cada anticorpo, ou seja, ao invés de cada anticorpo gerar um número fixo de clones, quanto maior for o grau de afinidade do anticorpo com o antígeno maior pode ser o número de clones gerados por esse anticorpo. Além disso, a implementação de uma busca local logo após a maturação da população de clones, pode fazer com que o processo de exploração do espaço de busca seja otimizado, possibilitando um melhor desempenho.

Agradecimentos

Os autores são gratos à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo apoio financeiro à realização desta pesquisa.

Referências Bibliográficas

- (1) Ahuja, R.K.; Magnati, T.L. & Orlin, J.B. (1993). *Network Flows: Theory, Algorithms and Applications*. Prentice-Hall.
- (2) Almeida, T.A.; Prado, F.M.S.; Sousa, V.N.; Takahashi, M.T. & Yamakami, A. (2005a). A Genetic Algorithm to solve the Minimum Spanning Tree Problem with Fuzzy Parameters using Possibility Measure. *Proceedings of the 24th International Conference North American Fuzzy Information Processing Society*, Ann Arbor, Michigan, USA, 627-632.

- (3) Almeida, T.A.; Takahashi, M.T. & Yamakami, A. (2005b). An Evolutionary Approach to Solve Minimum Spanning Tree Problem with Fuzzy Parameters. **In:** *International Conference on Computational Intelligence for Modeling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA & IAWTIC'06)* [edited by J. D. Pinter, PCS Inc. & Dalhousie, Christina Meier, Robert John and Masoud Mohammadian], Computer Society, **2**, 203-208.
- (4) Bazaraa, M.; Jarvis, J. & Sherali H.F. (1990). *Linear programming and network flows*. 2 ed., John Wiley & Sons, NY, USA.
- (5) Bäck, T.; Fogel, D.B. & Michalewicz, Z. (2000a). *Evolutionary Computation 1 Basic Algorithms and Operators*. Institute of Physics Publishing, Bristol, UK.
- (6) Bäck, T.; Fogel, D.B. & Michalewicz, Z. (2000b). *Evolutionary Computation 2 Advanced Algorithms and Operators*. Institute of Physics Publishing, Bristol, UK.
- (7) Beasley, J.E. (1990). OR-Library: Distributing Test Problems by Electronic Mail. *Journal of the Operational Research Society*, **41**(11), 1069-1072. <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>
- (8) Blue, M.P.; Bush, B.W. & Puckett, J. (2002). Unified approach to fuzzy graph problems. *Fuzzy Sets and Systems*, **125**(3), 355-368.
- (9) Chang, P.T. & Lee, E.S. (1999). Fuzzy decision Networks and Deconvolution. *Computers and Mathematics with Applications*, **37**(11), 53-63.
- (10) Chunde, Y. (1996). On the Optimization Problem of Spanning Tree in Fuzzy Network. *The Journal of China Universities of Posts and Telecommunications*, **3**(2), 22-25.
- (11) Dasgupta, D. (1998a). *Artificial Immune Systems and Their Applications*. Springer-Verlag, Berlin, Germany.
- (12) Dasgupta, D. (1998b). An Overview of Artificial Immune Systems and Their Applications. *Artificial Immune Systems and Their Applications*. Springer-Verlag, Berlin, Germany.
- (13) De Castro, L.N. (2001). Engenharia Imunológica: Desenvolvimento e Aplicação de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais. Tese de Doutorado, Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e Computação.
- (14) Dreher, H. (1995). *The Immune Power Personality*. Penguin Books, USA.
- (15) Dubois, D. & Prade, H. (1980). *Fuzzy Sets and Systems*. Academic Press, NY, USA.
- (16) Goldberg, M.C. & Luna, H.P. (2000). *Otimização Combinatória e Programação Linear: Modelos e Algoritmos*. Editora Campus, RJ, Brasil.
- (17) Gomide, F. & Pedrycz, W. (1998). *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press, MA, USA.
- (18) Holland, J.H. (1992). *Adaptation in Natural and Artificial Systems*. 2 ed., MIT Press, MA, USA.

- (19) Janeway, C.A.; Travers P.; Walport, M. & Capra, J.D. (2000). *Imunobiologia: O Sistema Imunológico na Saúde e na Doença*. 4 ed., Artmet, Porto Alegre, RS, Brasil.
- (20) Katagari, H. & Ishii, H. (2000). Chance Constrained Bottleneck Spanning Tree Problem with Fuzzy Random Edge Costs. *Journal of the Operations Research Society of Japan*, **43**, 128-137.
- (21) Katagari, H.; Sawada, M. & Ishii, H. (2004a). Fuzzy Random Bottleneck Spanning Tree Problems using Possibility and Necessity Measures. *European Journal of Operational Research*, **152**, 88-95.
- (22) Katagari, H.; Mermri, E.B.; Ishii, H. & Kat, K. (2004b). A Study on Fuzzy Random Minimum Spanning Tree Problems through Possibilistic Programming and the Expectation Optimization Model. *Proceedings of the 47th IEEE International Midwest Symposium on Circuits and Systems*, Hiroshima, Japan, 49-52.
- (23) Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, and Heidelberg GmbH & Co. K, Berlin, Germany.
- (24) Okada, S. (2001). Interactions among Paths in Fuzzy Shortest Path Problems. *Proceedings of the 9th International Fuzzy Systems Associations World Congress*, Vancouver, BC, Canada, 41-46.
- (25) Raidl, R.G. & Julstrom, B.A. (2001). Edge-Sets: An Effective Evolutionary of Spanning Trees. *IEEE Transactions on Evolutionary Computation*, **7**(3), 225-239.
- (26) Rosenfeld, A. (1975). Fuzzy graphs, Fuzzy Sets and Their Applications. **In:** *Proceeding of US-Japan Sem., University of California, Berkeley, CA, 1974* [edited by L.A. Zadeh, K.S. Fu and M. Shimura], Academic Press, NY, USA, 77-95.
- (27) Takahashi, M.T. (2004). Contribuições ao Estudo de Grafos Fuzzy: Teoria e Algoritmos. Tese de Doutorado, Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, SP, Brasil.
- (28) Timmis, J. (2000). Artificial Immune Systems: A Novel Data Analysis Technique Inspired by the Immune Network Theory. Tese de Doutorado, University of Wales, Department of Computer Science, Aberystwyth, Ceredigion, Wales.
- (29) Zadeh, L.A. (1965). Fuzzy Sets. *Information and Control*, **8**(3), 338-353.
- (30) Zadeh, L.A. (1978). Fuzzy Sets as a Basis for a Theory of Possibility. *Fuzzy Sets and Systems*, **1**(1), 3-28.