

UMA PROPOSTA PARA A SIMULAÇÃO DA DISTRIBUIÇÃO EFICIENTE DE CARGAS ATRAVÉS DE SISTEMAS IMUNOLÓGICOS ARTIFICIAIS

A proposal for simulation of efficient distribution of loads through artificial immune systems

ALVARENGA, Maria Cristina Campos Mendonça

Faculdade de Jaguariúna

PETROLI NETO, Silvio

Faculdade de Jaguariúna

Resumo: O objetivo deste trabalho é o desenvolvimento de um algoritmo, utilizando os princípios dos Sistemas Imunológicos Artificiais, para a simulação de uma distribuição eficiente de cargas. O conceito de carga usado aqui se refere aquilo que é ou pode ser transportado ou suportado por estrutura, pessoa, veículo, aparelho, etc. e que, conseqüentemente, gera um custo econômico, energético ou mesmo de tempo. Através de estudos sobre o Sistema Imunológico Humano, Sistemas Imunológicos Artificiais, Complexidade Computacional, P-Medianas e algoritmos como o ClonalG, pretende-se propor uma modelagem computacional aplicável a sistemas complexos simuláveis que demonstre que o sistema imunológico artificial pode ser usado como ferramenta na busca de “boas soluções”.

Palavras-chave: Sistemas Imunológicos Artificiais, ClonalG, NP-Completo, P-medianas.

Abstract: The main goal of this work is the development of an algorithm using the principles of Artificial Immune Systems for the simulation of an efficient distribution of loads. The concept of loads used here refers to what is or can be transported or supported by structure, person, vehicle, apparatus, etc. consequently generates an economic cost, energy or time. Through studies about Human Immune System, Artificial Immune Systems, Computational Complexity, P-Median and algorithms as ClonalG, we intend to propose a computational model applicable to complex systems that can be simulated to

show that the artificial immune system can be used as tool in pursuit of "good solutions"

Keywords: Artificial Immune Systems, ClonalG, NP-Complete, P-median.

INTRODUÇÃO

A ciência da computação como a conhecemos é uma ciência nova que não tem ainda nem um século de vida, contudo é uma das áreas do conhecimento que mais tem crescido nos últimos anos. Um dos temas mais novos desta ciência é a computação natural. Apesar da ampla abordagem da computação natural, ela pode ser sintetizada na aplicação de modelos de fenômenos e sistemas complexos observados na natureza à ciência da computação.

Em 2006, a Sociedade Brasileira de Computação (SBC) promoveu um seminário onde cientistas e pesquisadores debateram sobre os Grandes Desafios da Pesquisa em Computação no Brasil para os próximos dez anos. Dentre os cinco grandes temas eleitos naquela ocasião está o tema "*Modelagem Computacional de Sistemas Complexos Artificiais, Naturais e Sócio-culturais e da Interação Homem-natureza*". Dentro deste contexto, identifica-se um quadro promissor para a pesquisa científica neste campo, o qual está atrelado a fatores como o aumento da complexidade dos problemas computacionais, a necessidade de se combinar várias áreas de conhecimento, o aumento sistemático do volume de dados e a constante evolução de hardware e softwares.

O objetivo deste trabalho é o desenvolvimento de uma modelagem computacional, utilizando os princípios dos Sistemas Imunológicos Artificiais, para o fornecimento de "possíveis soluções" a problemas que possam ser simulados. É o caso do problema de alocação de clientes a servidores, objetivando-se o menor custo e respeitando-se as capacidades de processamento e de armazenamento destes últimos. O objetivo não é desenvolver um programa para a solução desta questão específica, mas a

partir deste exemplo, modelar um algoritmo baseado em SIA que possibilite a simulação de outros problemas similares a este.

COMPLEXIDADE COMPUTACIONAL E OS PROBLEMAS NP-COMPLETOS

Problemas no contexto da ciência da computação são avaliados de acordo com o seu grau de complexidade. Esta complexidade está relacionada ao consumo de tempo que um sistema computacional leva para solucionar determinados tipos de problemas.

Para a análise desta complexidade, Russel e Norvig (2004, p.943) efetuam uma “primeira divisão bruta” classificando-os em “*problemas que podem ser resolvidos em tempo polinomial*” e aqueles que não podem, independente do algoritmo usado. Estes problemas são considerados tratáveis, enquanto que, uma outra classe denominada NP (polinomiais não determinísticos), que possuem um tempo de solução exponencial, são considerados “intratáveis”.

Os problemas NP-Completo são tidos como os problemas de nível de complexidade mais elevados e somente são provados a partir de outros problemas também NP-Completo já conhecidos. De acordo com Pfützenreuter (2005, p.2), eles possuem três características peculiares que, de certa forma, viabilizam a identificação de um problema como NP-Completo, ou não: se a análise do problema resulta em uma solução ideal não polinomial; ou se soluções não tidas como as ideais são aceitáveis e úteis; ou se estas podem ser obtidas por algoritmos polinomiais, por tentativas aleatórias ou pelo uso de ambas as técnicas; então podemos dizer que o problema é um NP-Completo.

P-MEDIANAS

O problema selecionado descreve um grafo, cabendo portanto, relembrar alguns conceitos da Teoria de Grafos. Um grafo é dado pela

representação $G(V,E)$ onde G é composto por um conjunto finito de vértices V e um conjunto de arestas E . Os vértices são os nodos (nós) que representam pontos fixos, enquanto as arestas são as linhas que ligam estes pontos. Um vértice é chamado de adjacente a outro quando existe uma aresta que os conecta, enquanto uma aresta é dita adjacente a outra quando se conectam ao mesmo vértice.

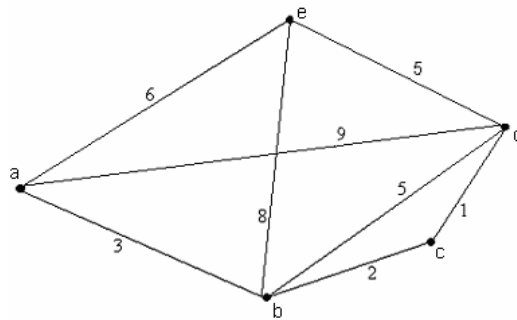


Figura 1: Grafo não orientado. **Fonte:** CAMICIA & VICENTE, 2008, p. 6

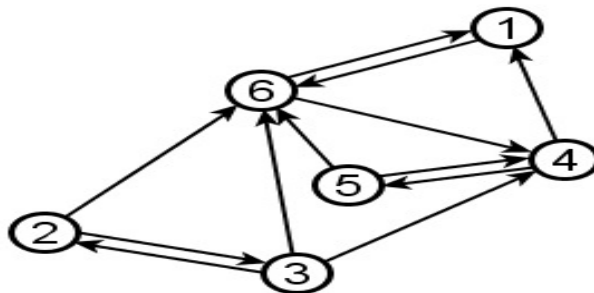


Figura 2: Grafo orientado. **Fonte:** FEOFILOFF, 2010

Um grafo pode ser orientado (figura 1) ou não orientado (figura 2). Chamamos de grafo não orientado àquele onde uma mesma aresta possui um sentido duplo. O grafo orientado ou dirigido, por sua vez, é àquele onde existem arestas que possuem um sentido único definido de um nó para outro, ou seja, possuem orientação e direção.

Um grafo é dito valorado quando se atribuem valores as suas arestas. A ordem de um grafo é dada pela cardinalidade do conjunto de vértices. [...] Em um grafo não orientado, o grau de um vértice é o numero de arcos que incidem sobre ele. No caso do grafo ser dirigido, fala-se em grau de entrada e grau de saída. O grau de entrada é o numero de arestas que chegam a

ele, e o grau de saída é o número de arestas que saem dele (CAMICIA & VICENTE, 2010, p. 3).

Uma das formas mais utilizadas de representação de grafos é a matriz de adjacência. Atribui-se a identificação do vértice a uma linha (i) e coluna (j). Se existir aresta que conecte i a j , o “custo” da aresta $E(i, j)$ será 1 e será 0, se os vértices não forem adjacentes.

O grau de um vértice pode ser obtido a partir da soma dos elementos da linha ou coluna da matriz de adjacência, em se tratando de um grafo não orientado. Já no dígrafo ou grafo orientado ou dirigido, “a soma dos elementos na coluna i representa o grau de saída do vértice v_i enquanto a soma dos elementos na linha i representa o grau de entrada de v_i ” (NONATO, 2000).

Duas outras formas de representação são a matriz de incidência e a matriz de custos. Na de incidência cada linha (i) representa cada um dos nós do grafo, e as colunas (j), os seus possíveis arcos (ou arestas). Nos grafos não orientados, ao elemento $M(i,j)$ será atribuído 1 se i for o vértice inicial de j e 0 caso contrário; enquanto nos dígrafos, atribui-se 1 se i for o vértice inicial, -1 se final e 0 se for laço.

A matriz de custos é utilizada na representação de grafos valorados, podendo estes valores representar distância, tempo de viagem, etc. e “são chamados custo, independentemente do que estejam representando”. Em síntese, o elemento $M(i,j)$ será o custo da aresta (i, j) se esta existir, 0 se $i=j$ e ∞ se não existir a aresta (CAMICIA; VICENTE, 2010, p. 4).

A mediana de um grafo corresponde ao nó a partir do qual é possível “visitar” os outros nós com a menor soma de “custos”. Um problema pode possuir como resultado apenas 1 vértice como mediana, 2 ou até um número “ p ” de medianas. Os problemas que possuem um número indefinido de medianas como possíveis resultados são ditos como Problemas de P -medianas.

Em um grafo não orientado onde V é o conjunto dos vértices do grafo “deve-se encontrar um conjunto de vértices $V_p \subset V$ (V_p é o conjunto das medianas do problema) com cardinalidade p , tal que a soma das distâncias de

cada vértice pertencente a V até seu vértice mais próximo em V_p seja a mínima possível". (ROSÁRIO; CARNIERI; STEINER, 2010)

SISTEMA IMUNOLÓGICO HUMANO (SIH)

A computação natural está fundamentada na observação dos fenômenos naturais e de sua aplicação de forma análoga em sistemas computacionais. O Sistema Imunológico Artificial como o próprio nome revela, se baseia nos princípios e conceitos observados no Sistema Imunológico Humano (SIH) sendo, portanto, importante elucidar estes conceitos e princípios no seu contexto original.

Três conceitos fundamentais para a compreensão do mecanismo de defesa do SIH são os patógenos, antígenos e anticorpos. Os patógenos são agentes infecciosos que carregam a substância nociva que chamamos de antígeno, podendo ser extra ou intracelular de acordo com a forma que se alojam no organismo atacado. São exemplos de intracelulares, os vírus, as bactérias e os protozoários, sendo que estes últimos se replicam no interior das células do hospedeiro. Algumas bactérias e parasitas maiores se alojam em tecidos e fluídos corpóreos e por isto são chamadas de extracelulares.

Segundo Trevisan et al (2010), o antígeno é uma *“substância orgânica nociva, de natureza geralmente protéica, que, inoculada no organismo, provoca, como reação, a produção de anticorpos específicos”*. Enquanto o anticorpo é uma *“substância específica de origem celular, que torna inócuas substâncias orgânicas capazes de produzir moléstias, quando introduzidas no organismo”*.

O organismo humano possui um mecanismo de defesa que ocorre em multicamadas e cobre toda a extensão do corpo. Além das barreiras físicas como a pele e barreiras bioquímicas como o nível de pH da saliva, o sistema imunológico pode ser dividido em inato e adaptativo (CASTRO, 2006, p.268).

No processo inato temos a primeira defesa onde o SIH reconhece o antígeno, envia sinais para as células e responde de forma inalterada ao ataque. No adaptativo, o SIH memoriza antígenos já anteriormente

reconhecidos, produzindo anticorpos específicos e se tornando mais eficaz a cada encontro com patógenos do mesmo tipo.

Neste contexto de defesa adaptativa o SIH conta com os linfócitos que, segundo Machado (2005, p.14), são células de glóbulos brancos responsáveis pelas atividades de reconhecimento de patogenicias, respostas e memorização imunológicas. Estas células dão origem a duas importantes células: B e T.

As células B são oriundas da medula óssea e codificam o receptor de superfície específico para o reconhecimento do antígeno. Se este for reconhecido, a célula B se divide e diferencia em plasmócitos e linfócitos de memória. Os plasmócitos são responsáveis pela produção de anticorpos em grande escala e pela distribuição nas células infectadas. Os linfócitos de memória, por sua vez, não produzem anticorpos, vivem por longos períodos e, se forem re-expostos ao antígeno, o reconhecem e novamente se dividem e diferenciam. As células T amadurecem no Timo e daí a origem do seu nome. Elas atuam diretamente nas células atacadas por antígenos, destruindo os patógenos intracelulares. São também responsáveis pela produção e distribuição de anticorpos.

A teoria da seleção clonal está relacionada ao SIH adaptativo e, de acordo com Machado (2005, p.18):

Quando há exposição a um antígeno, uma subpopulação de linfócitos (células B) responde por meio da produção de anticorpos. Cada célula secreta um único tipo de anticorpo, que é relativamente específico para o antígeno. Por meio da ligação do antígeno com o receptor da célula B e, dado um sinal co-estimulatório de células acessórias como a T-Helper, um antígeno estimula a célula B a se proliferar (clones produzidos por meio de divisão celular). Esse processo é denominado expansão clonal e as células B resultantes possuem receptores modificados devido à hipermutação somática ocorrida durante a clonagem.

Estes novos linfócitos são também expostos ao antígeno competindo com os anteriores e ocorre uma seleção das células que são capazes de reconhecer o antígeno, determinando assim o princípio da Seleção Clonal.

SISTEMA IMUNOLÓGICO ARTIFICIAL (SIA)

Hoje já existem várias técnicas para o desenvolvimento de algoritmos baseados no Sistema Imunológico Artificial. Entre eles podemos destacar: Medula óssea, Seleção negativa, Seleção clonal, Modelos Contínuos de Rede Imunológica e Modelos Discretos de rede imunológica.

No SIA, os algoritmos de Seleção Clonal possuem três características próprias que os identificam: o encontro com antígenos é sequencial, a taxa de proliferação é proporcional à afinidade e a taxa de mutação é inversamente proporcional à afinidade.

A AFINIDADE ENTRE ANTÍGENOS E ANTICORPOS

Para a representação dos antígenos e anticorpos no SIA, Perelson e Oster desenvolveram o conceito de “espaço de formas” onde a molécula é representada por uma String sendo os mais conhecidos: O espaço de forma Real, Inteiro, Hamming e o simbólico. Os dois primeiros reconhecem valores reais e inteiros respectivamente. Já o de Hamming é baseado em um determinado alfabeto finito e o simbólico aceita quaisquer tipos de atributos, desde que entre eles exista ao menos um símbolo.

A partir do espaço de formas de Hamming, que considera um determinado alfabeto K, é possível estabelecer uma medida de afinidade. Castro (2006, p. 282-283) exemplifica a verificação do grau de afinidade de forma simples utilizando um alfabeto binário, conforme segue:

A Equação (...) descreve a distância Hamming usada para avaliar a afinidade entre duas strings de atributos de tamanho L no espaço de formas de Hamming:

$$D = \sum_{i=1}^L \delta, \text{ onde } \delta = \begin{cases} 1 & \text{se } Ab_i \neq Ag_i \\ 0 & \text{senão} \end{cases}$$

(...) Para poder ilustrar melhor, considere as strings

Ag = [1000111010] e **Ab** = [1010101010].

Ag = [1000111010]

Ab = [1010101010]

Ligação (**Ab, Ag**): 0010010000

Complementaridade: $D(\mathbf{Ab}, \mathbf{Ag}) = \sum \text{Ligação} (\text{Afinidade} = 2)$

Similaridade: $L - D(\mathbf{Ab}, \mathbf{Ag})$ ($\text{Afinidade} = 8$)

ALGORITMO CLONALG

O algoritmo de seleção clonal conhecido por ClonalG foi proposto em 2002 por Castro e Von Zuben e apresenta os passos básicos descritos a seguir de acordo com Castro e Timmis (2002, apud Castro, 2006, p. 298-299):

1. **Inicialização:** cria-se uma população de indivíduos
2. **Apresentação do antígeno:** cada elemento da população é apresentado ao antígeno (padrão)
3. **Avaliação da afinidade:** calcula-se a afinidade de cada anticorpo com o antígeno.
4. **Seleção Clonal e Expansão:** são selecionados os n_1 com maiores afinidades e são clonados proporcionalmente, quanto maior a afinidade, maior o número de cópias.
5. **Maturação da Afinidade:** as cópias são alteradas na proporção inversa de sua afinidade, e adicionadas à população de anticorpos.
6. **Metadinamica:** um número n_2 de indivíduos de baixa afinidade são substituídos por novos indivíduos gerados randomicamente

Ciclo: o Passo 2 é repetido até que um certo critério de parada seja alcançado.

RESULTADOS

A modelagem constante na figura 3 foi implementada em Java, tendo em vista a sua modularidade e a gama de recursos disponíveis (bibliotecas)

favorecendo o desenvolvimento de um código que viabiliza o seu aperfeiçoamento futuro.

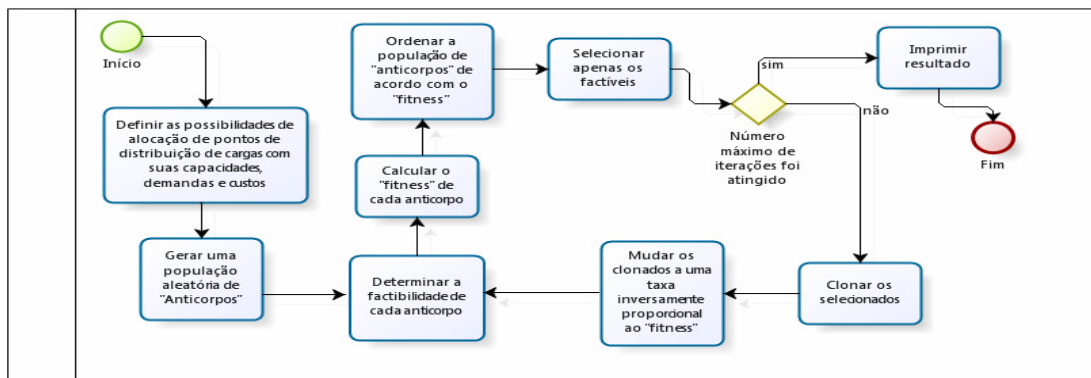


Figura 3: Algoritmo para a alocação de pontos para a distribuição de cargas baseado em princípios da seleção clonal. **Fonte:** construído com a ferramenta freeware *BizAgi Process Modeler* (BIZAGI - BUSINESS AGILITY, 2010)

O primeiro passo foi a definição de um problema com dados determinados e com um volume que permitisse a visualização dos resultados de maneira mais clara. Para tanto, optou-se pela representação de um grafo (figura 4) com dois servidores e cinco estações o que gera no “pior caso” a “ativação” de 10 arestas, conectando cada uma das estações aos dois servidores.

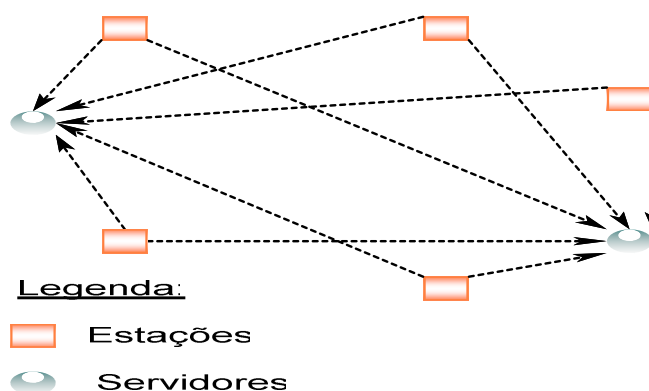


Figura 4: Representação básica do problema.

Inicialmente modelou-se as classes estruturais: “Servidor”, “Estacao” e “aresta” conforme diagrama de classe apresentado na figura 5. Os objetos

destas classes foram instanciados na classe “sia” com a atribuição dos valores correspondentes os quais são demonstrados na representação gráfica da figura 6.

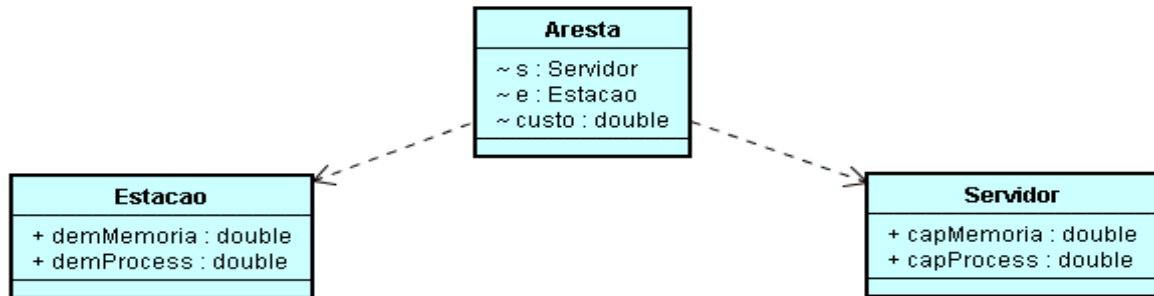


Figura 5: Diagrama de classes estrutural

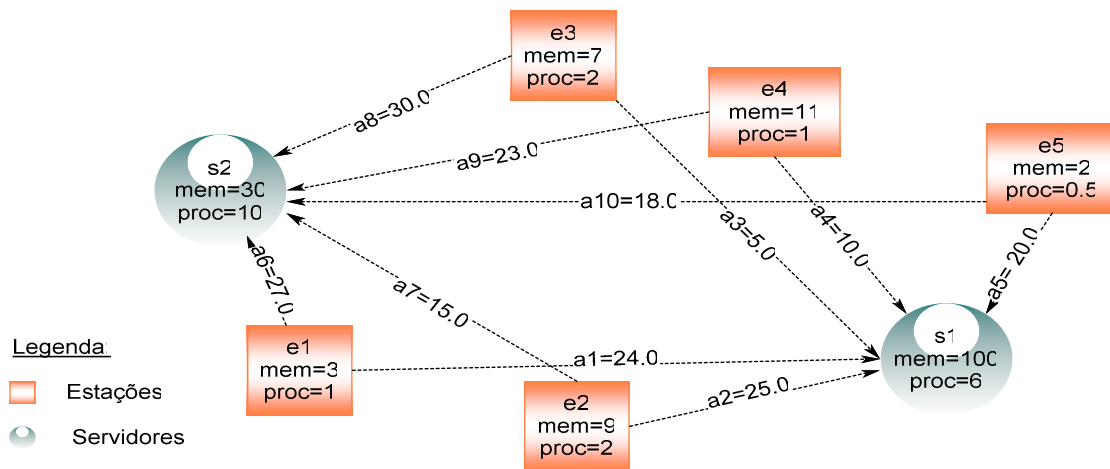


Figura 6: Representação do problema com os valores atribuídos

O anticorpo é representado como um vetor de arestas, onde se considerou todas as arestas possíveis do problema que, neste caso são 10. A cada aresta, é atribuído um dos valores 0 ou 1, sendo que 0 indica que a aresta não compõe aquela solução (ou anticorpo) e 1 indica que compõe. Assim, é

possível a geração aleatória de anticorpos os quais são armazenados em uma matriz conforme demonstrado na figura 7.

run:										
0	0	1	1	0	0	1	1	1	1	=> Anticorpo 1
0	0	1	1	0	0	0	1	0	1	=> Anticorpo 2
1	0	0	1	1	1	0	1	1	1	=> Anticorpo 3
0	0	1	1	0	1	0	0	0	1	=> Anticorpo 4
0	0	1	1	1	0	0	0	1	1	=> Anticorpo 5

Figura 7: Exemplo da matriz de anticorpos (arestas ativas)

Estes anticorpos são verificados quanto à sua factibilidade, ou seja, se os servidores são capazes de absorver a demanda de carga (armazenamento e processamento) das estações a eles associadas. Se alguma das arestas ultrapassar a capacidade do servidor, o anticorpo é considerado ineficaz e terá somado ao seu custo específico o custo total das arestas.

O próximo passo consiste no estabelecimento de um “fitness” que neste problema coincide com o custo. Contudo, como esta modelagem pretende servir como base para a simulação de outros problemas, é importante que este parâmetro seja definido. Assim, apurou-se o fitness de cada anticorpo, atribuindo-se a este, a diferença entre o custo total da população e o seu custo específico. Com isto tem-se que, o anticorpo com menor custo, ou seja a melhor solução, terá o maior *fitness*.

Para a ordenação da população de anticorpos de acordo com o *fitness*, foi necessário agregar à matriz novas colunas contendo os dados de *fitness*, custo e identificação de cada anticorpo. Após sua ordenação uma nova matriz foi gerada, contendo apenas os anticorpos factíveis e que serão submetidos à clonagem e mutação. Estes procedimentos foram definidos dentro da classe “Populacao”.

Neste momento efetua-se a seleção apenas dos anticorpos factíveis, os quais serão clonados e posteriormente mudados. A taxa de mutação é inversamente proporcional ao *fitness*, ou seja, os anticorpos de maior *fitness*

serão os menos alterados, enquanto os de menor *fitness* terão mais elementos (arestas) alteradas.

O resultado da mutação é fornecido novamente para o teste de factibilidade, o cálculo do fitness, nova ordenação e seleção, por N vezes e finalmente, é impresso o resultado.

DISCUSSÃO

Os testes foram efetuados considerando-se três parâmetros variáveis: o número inicial de anticorpos gerados aleatoriamente, o número de iterações do ciclo e o número de resultados capturados.

Os resultados gerados pelo algoritmo implementado foram impressos no formato abaixo, onde estão demonstrados: o número do resultado, a sua classificação dentro de cada conjunto de iterações, a identificação das arestas ativas que compõe a solução e o custo do resultado, conforme exemplo ilustrado na figura 8. Estes resultados foram transportados para uma planilha do Excel para serem ordenados por custo, de forma a facilitar o cálculo dos subtotais.

```
9891, 1, , 1, , 3, 4, , , 7, , , 10, , , 72,  
9892, 1, , 1, , 3, 4, , , 7, , 9, 10, , , 95,  
9893, 1, , 1, , 3, , , , 7, , 9, 10, , , 85,
```

Figura 8: Exemplo da impressão oferecida pelo programa.

Primeiramente optou-se por fixar o número de anticorpos em 500, e de iterações em 100, alterando apenas o número de resultados recuperados gerando os dados conforme demonstrado na tabela 1, onde é possível observar que as 7(sete) melhores soluções aparecem em todas as simulações, independente do número de resultados recuperados.

Tabela 1. Resultados obtidos **variando o número de resultados** e fixando o número de anticorpos iniciais em 500 e de iterações em 100

50		100		500		1000		2000		5000		10000	
Qua	Cus	Qua	Cus	Qua	Cus	Qua	Cus	Qua	Cus	Qua	Cus	Qua	Cus
19	72	33	72	157	72	356	72	660	72	1656	72	3365	72
13	74	21	74	98	74	231	74	458	74	1094	74	2144	74
3	75	8	75	80	75	148	75	322	75	743	75	1516	75
5	77	11	77	60	77	93	77	181	77	489	77	990	77
3	82	9	82	45	82	63	82	111	82	334	82	673	82
3	85	11	85	35	85	63	85	150	85	383	85	725	85
3	87	5	87	10	87	26	87	66	87	169	87	321	87
1	95	1	97	3	88	7	88	20	88	40	88	82	88
-	-	1	98	5	90	4	90	9	90	31	90	53	90
-	-	-	-	2	92	3	92	7	92	20	92	36	92
-	-	-	-	4	95	3	95	11	95	27	95	71	95
-	-	-	-	1	97	1	97	3	97	12	97	21	97
-	-	-	-	-	-	1	98	1	98	2	98	1	98
-	-	-	-	-	-	1	99	1	99	-	-	2	99

Os gráficos apresentados na figura 9 demonstram que à medida que o número de resultados recuperados aumenta, a representação de uma curva exponencial decrescente se torna mais evidente.

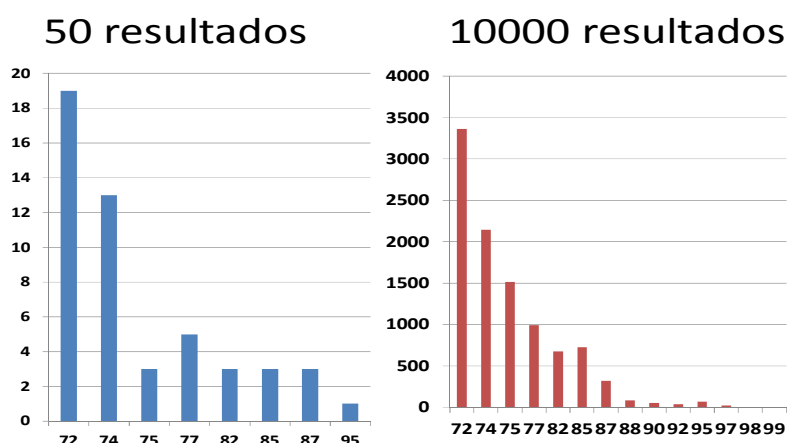


Figura 9: Distribuição de Quantidade de Resultados Obtidos (eixo y) por custo (eixo x)

Estes dados também foram analisados através de um gráfico de ajuste de curva, de forma a demonstrar se o algoritmo seguia um padrão de

comportamento. Para tanto foi utilizada a versão demo da ferramenta computacional DataFit (OAKDALE ENGINEERING, 2010).

No caso da variação do número de resultados gerados foram obtidos os gráficos de ajustes de curva conforme a figura 10 para o menor (50) número de resultados verificados e a figura 11 para o maior (10000).

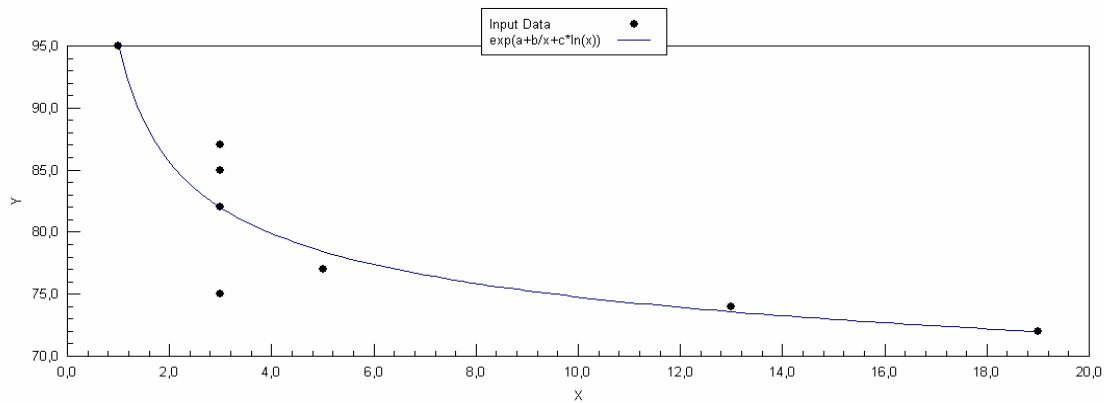


Figura 10: Ajuste de curva para a distribuição de custos (y) por quantidade de resultados recuperados (x), considerando 50 anticorpos iniciais, 100 iterações e 50 resultados recuperados.

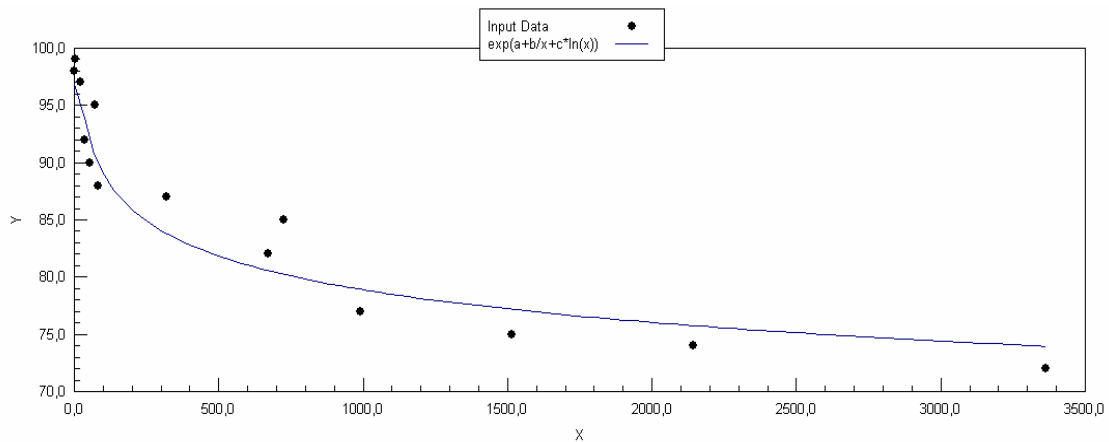


Figura 11: Ajuste de curva para a distribuição de custos por quantidade de resultados recuperados, considerando 500 anticorpos iniciais, 100 iterações e 10000 resultados recuperados.

Deve-se perceber que a assíntota foi obtida em ambos os resultados, tendo sido observado também nos testes com valores intermediários denotando um padrão de comportamento exponencial do algoritmo no tratamento do problema piloto. Na sequência, fixou-se o número de resultados

em 5000 e de anticorpos em 500, alterando apenas o número de iterações gerando os dados conforme demonstrado na tabela 2:

Tabela 2. Resultados obtidos variando o número de iterações e fixando o número de anticorpos iniciais e de resultados capturados em 100

5		10		100		1000	
Qtde	Custo	Qtde	Custo	Qtde	Custo	Qtde	Custo
1577	72	1667	72	1743	72	1702	72
1108	74	1078	74	1066	74	1103	74
800	75	739	75	776	75	741	75
496	77	492	77	447	77	473	77
333	82	376	82	324	82	326	82
391	85	346	85	356	85	351	85
164	87	165	87	152	87	164	87
45	88	40	88	45	88	41	88
29	90	38	90	32	90	39	90
26	92	17	92	17	92	17	92
25	95	37	95	29	95	29	95
4	97	4	97	13	97	11	97
2	99	1	100	-	-	2	98
-	-	-	-	-	-	1	100

Estes testes também geraram gráficos de colunas com a mesma tendência exponencial tanto nos gráficos de coluna como no gráfico de ajuste de curva. São apresentados exemplos destes gráficos para 5 e 1000 iterações respectivamente nas figuras 12, 13, 14 e 15.

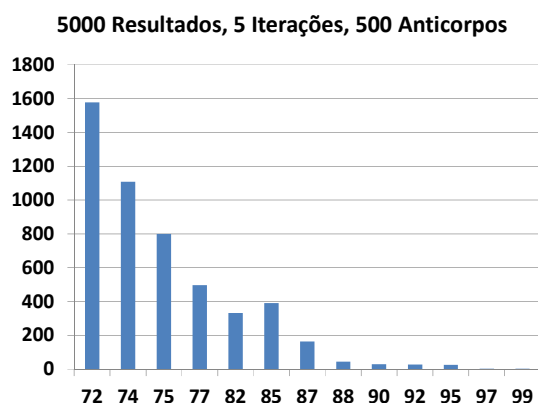


Figura 12 Distribuição de quantidade de resultados obtidos (eixo y) por custo (eixo x).

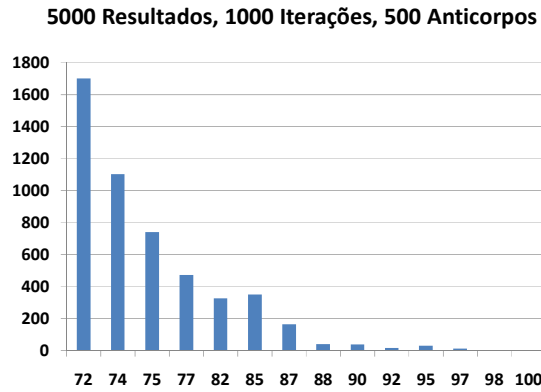


Figura 13: Distribuição de quantidade de resultados obtidos (eixo y) por custo (eixo x)

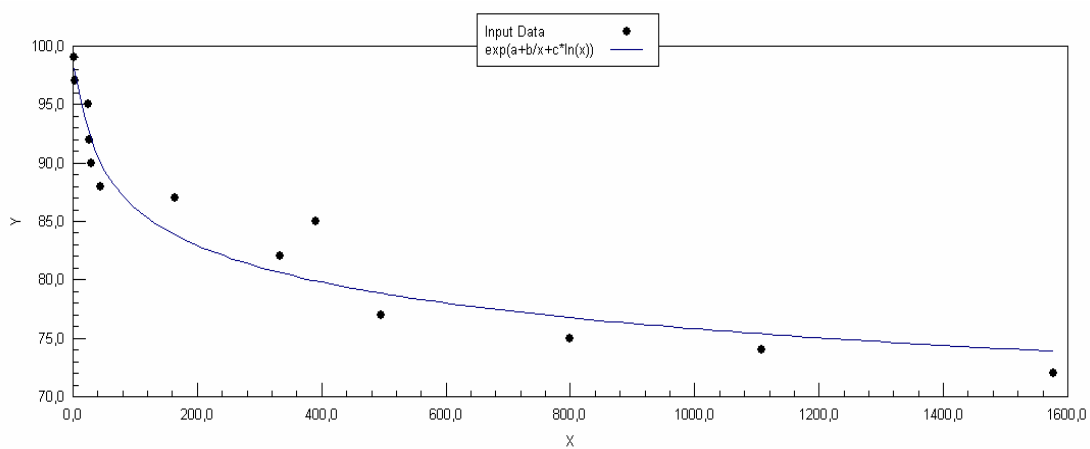


Figura 14: Ajuste de curva para a distribuição de custos (y) por quantidade de resultados recuperados (x), considerando 500 anticorpos iniciais, **5 iterações** e 5000 resultados recuperados.

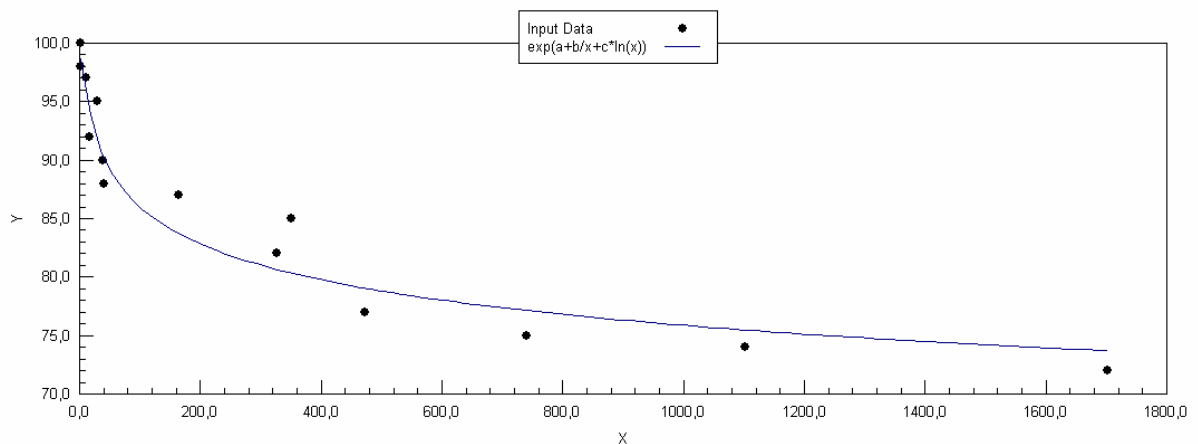


Figura 15: Ajuste de curva para a distribuição de custos (y) por quantidade de resultados recuperados (x), considerando 500 anticorpos iniciais, **1000 iterações** e 5000 resultados recuperados.

Foram realizados ainda testes fixando-se o número de resultados e de iterações em 100, alterando apenas o número de anticorpos iniciais gerando os dados conforme a tabela 3. Observa-se que com um número baixo de

anticorpos iniciais a diversidade de anticorpos encontrados é grande e que quanto maior o “número de anticorpos iniciais”, maior a concentração de resultados na “melhor solução”, o que também é mostrado nas figuras, 16 e 17.

Tabela 3. Resultados obtidos variando o número de anticorpos e fixando o número de iterações e de resultados capturados em 100.

50		250		500		1000		2500		5000	
Qtde	Custo	Qtde	Custo	Qtde	Custo	Qtde	Custo	Qtde	Custo	Qtde	Custo
2	72	20	72	31	72	56	72	85	72	99	72
6	74	18	74	26	74	26	74	15	74	1	74
6	75	9	75	12	75	11	75	-	-	-	-
2	77	4	77	7	77	3	77	-	-	-	-
6	82	7	82	9	82	2	82	-	-	-	-
5	85	14	85	10	85	2	85	-	-	-	-
5	87	12	87	2	87	-	-	-	-	-	-
4	88	3	88	2	90	-	-	-	-	-	-
5	90	2	90	1	95	-	-	-	-	-	-
5	92	2	92	-	-	-	-	-	-	-	-
6	95	5	95	-	-	-	-	-	-	-	-
4	97	3	97	-	-	-	-	-	-	-	-
3	98	1	100	-	-	-	-	-	-	-	-
6	99	-	-	-	-	-	-	-	-	-	-
3	100	-	-	-	-	-	-	-	-	-	-
4	102	-	-	-	-	-	-	-	-	-	-
2	104	-	-	-	-	-	-	-	-	-	-
4	105	-	-	-	-	-	-	-	-	-	-
1	107	-	-	-	-	-	-	-	-	-	-
3	109	-	-	-	-	-	-	-	-	-	-
4	110	-	-	-	-	-	-	-	-	-	-
2	112	-	-	-	-	-	-	-	-	-	-
2	113	-	-	-	-	-	-	-	-	-	-
1	115	-	-	-	-	-	-	-	-	-	-
1	117	-	-	-	-	-	-	-	-	-	-
1	124	-	-	-	-	-	-	-	-	-	-
1	125	-	-	-	-	-	-	-	-	-	-
1	126	-	-	-	-	-	-	-	-	-	-
2	127	-	-	-	-	-	-	-	-	-	-
1	128	-	-	-	-	-	-	-	-	-	-
1	132	-	-	-	-	-	-	-	-	-	-
1	144	-	-	-	-	-	-	-	-	-	-

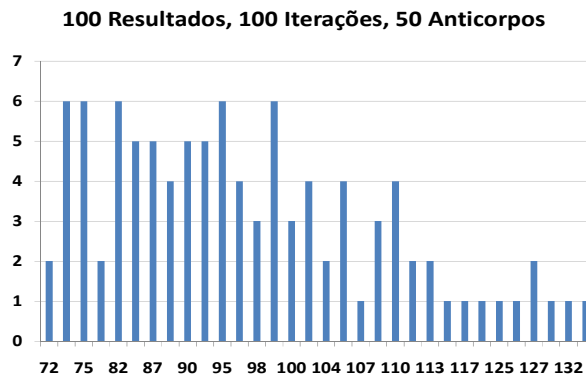


Figura 16: Distribuição das quantidades de resultados obtidos (y) por custo (x).

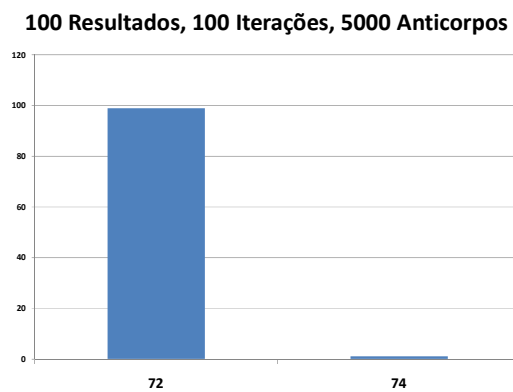


Figura 17: Distribuição das quantidades de resultados obtidos (y) por custo (x).

Estes testes geraram ajustes de curva, cujos gráficos também demonstraram a mesma tendência de curva exponencial assintótica como exemplificado nas figuras 18 e 19.

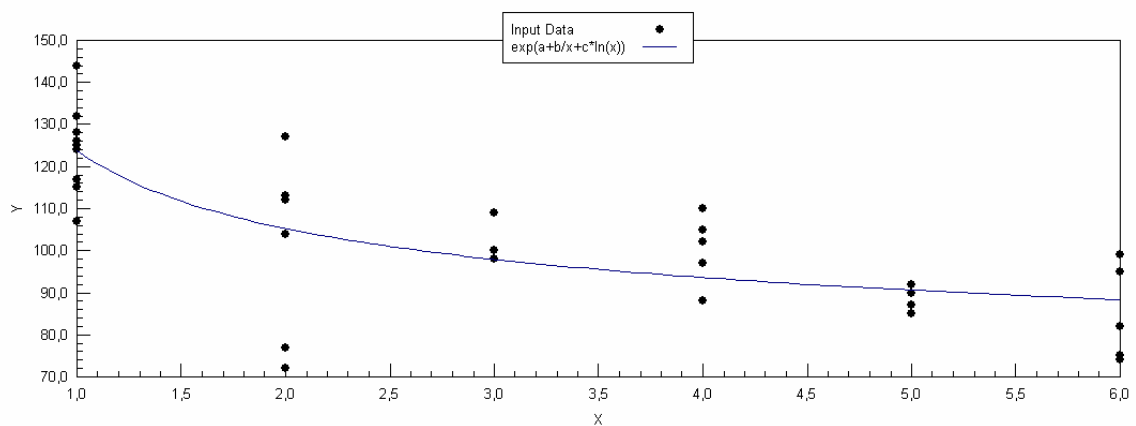


Figura 18: Ajuste de curva para os resultados obtidos com 100 iterações, 100 resultados e 50 anticorpos iniciais.

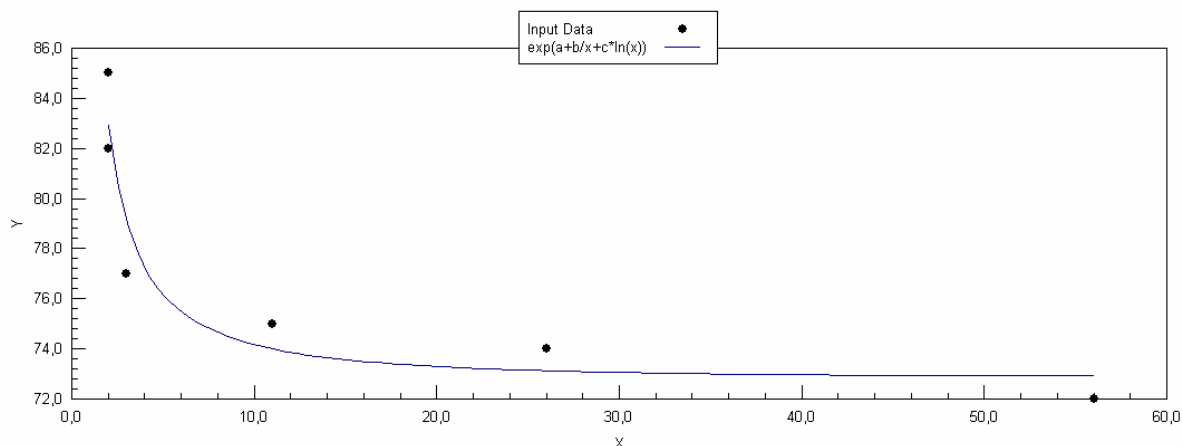


Figura 19: Ajuste de curva para os resultados obtidos com 100 iterações, 100 resultados e 1000 anticorpos iniciais.

Por fim, observa-se na tabela 4 que o tempo varia de acordo com a quantidade de ciclos solicitados, porém as mesmas melhores soluções foram obtidas em todos os testes realizados.

Tabela 4: Tempo de processamento

Anticorpos (A)	Iterações (I)	Resultados Recuperados (RR)	A x I x RR	Tempo de processamento (em segundos)
50	100	100	500000	1
250	100	100	2500000	6
500	100	50	2500000	7
500	100	100	5000000	13
500	100	100	5000000	13
1000	100	100	10000000	27
500	5	5000	12500000	44
500	100	500	25000000	70
500	10	5000	25000000	82
2500	100	100	25000000	93
500	100	1000	50000000	139
5000	100	100	50000000	301
500	100	2000	100000000	295
500	100	5000	250000000	780
500	100	5000	250000000	668
500	100	10000	500000000	1358
500	1000	5000	2500000000	6011

CONSIDERAÇÕES FINAIS

Considerando que a alocação de clientes a servidores a custo mínimo, a exemplo do problema do caixeiro viajante, é tido como um problema NP-completo, a implementação de um algoritmo que trate qualquer volume maior de dados, poderia inviabilizar o trabalho. A representação de um problema fictício de pequeno porte permitiu a observação da tendência dos resultados e do comportamento do algoritmo proposto. Foi possível a obtenção de soluções ótimas para um volume de dados determinados e “tratáveis”, o que vislumbra uma necessidade de se explorar a aplicação deste modelo computacional na simulação de novos problemas com níveis de complexidade variados e com volumes de dados maiores em futuras pesquisas. É importante observar ainda que os resultados apresentados demonstram a característica de multimodalidade dos “Sistemas Imunológicos Artificiais”, ou seja, não apenas a solução ótima global do problema é obtida, mas também soluções locais de otimalidade. Essa característica permite explorar soluções que, apesar de não serem a melhor, continuam sendo viáveis e factíveis.

REFERÊNCIAS

- ALMEIDA, T. A.; YAMAKAMI, A. and TAKAHASHI, M. T. **Sistema imunológico artificial para resolver o problema da árvore geradora mínima com parâmetros fuzzy**. *Pesqui. Oper.* [online]. 2007, v. 27, n.1, pp. 131-154. ISSN 0101-7438.
- BIZAGI - BUSINESS AGILITY. **BizAgi Process Modeler**. Disponível em: <<http://www.bizagi.com/>>. Acesso em: 24 mar. 2010.
- CAMICIA, R. G. M.; VICENTE, A. **Grafos e Mediana**. Disponível em: <<http://projetos.unioeste.br/cursos/cascavel/matematica/xxiisam/artigos/15.pdf>>. Acesso em: 01 maio 2010.
- CASTRO, L. N. **Fundamentals of Natural Computing: basic concepts, algorithms, and applications**. Boca Raton, FL, USA: Chapman & Hall/crc, 2006. 662 p.
- CEZAR, C. C. **Estudo da Otimização de Funções através de Sistemas Imunológicos Artificiais**. 2009. 22 f. Trabalho de Conclusão de Curso (Graduação) - Curso de Ciência da Computação, Faculdade de Jaguariúna, Jaguariúna, 2009.

FEOFILOFF, P. **Análise de Algoritmos**. Disponível em: <http://www.ime.usp.br/~pf/analise_de_algoritmos/> Acesso em: 3 abr. 2010.

NONATO, L. G. **Matriz de Adjacência**. 2000. Disponível em: <<http://www.lcad.icmc.usp.br/~nonato/ED/Grafos/node73.html>>. Acesso em: 01 maio 2010.

OAKDALE ENGINEERING (USA). **DataFit**: version 9.0.59. Evaluation Copy. Disponível em: <<http://www.curvefitting.com/datafit.htm>>. Acesso em: 13 nov. 2010.

PFÜTZENREUTER. **O que são problemas Np-Completos**. 2005. Disponível em: <<http://epx.com.br/artigos/npcompleto.php>>. Acesso em: 17 abr. 2010.

ROSÁRIO, R. R. L.; CARNIERI, C.; STEINER, M. T. A. **Proposta de Solução para o Problema das P-Mediana na Localização de Unidades de Saúde 24 Horas**. Disponível em: <http://www.abepro.org.br/biblioteca/ENEGEP2002_TR61_0949.pdf>. Acesso em: 29 ago. 2010.

RUSSEL, S.; NORVIG, P. **Inteligência Artificial**. 2. ed. Rio de Janeiro: Campus, 2004. 1021 p.

TREVISAN, R. et al. **Michaelis – Moderno Dicionário da Língua Portuguesa**. Editora Melhoramentos Ltda. Disponível em: <<http://michaelis.uol.com.br/moderno/portugues/index.php>> Acesso em: 18 jun. 2010.