

AUTO-ORGANIZAÇÃO DA POPULAÇÃO EM SISTEMAS IMUNOLÓGICOS ARTIFICIAIS PARA OTIMIZAÇÃO MULTIMODAL CONTÍNUA

Helder Ken Shimo, Renato Tinós

Departamento de Computação e Matemática – FFCLRP, Universidade de São Paulo, shimo.ken@usp.br, rtinos@ffclrp.usp.br

Resumo – Sistemas Imunológicos Artificiais (*Artificial Immune Systems* – AIS) são algoritmos de otimização inspirados na adaptabilidade do sistema imunológico de organismos complexos. No opt-aiNet, um AIS desenvolvido para a otimização de funções multimodais contínuas, soluções candidatas representam células do sistema imunológico e, por ser inspirado na teoria de redes imunológicas, estas tem uma forma de interação entre si. Esta ocorre de forma inibitória em uma etapa de supressão, que busca eliminar soluções semelhantes a fim de evitar a convergência e aumentar a variabilidade, adicionando novas soluções aleatórias. A etapa de supressão é um procedimento custoso, uma vez que calcula a distância euclidiana entre cada possível par de células da população e sendo, desta forma, determinista, é fortemente influenciado pela definição dos parâmetros. Neste trabalho é apresentada uma técnica de supressão inspirada no fenômeno da criticalidade auto-organizada (*Self-Organizing Criticality* – SOC). O método considera que as células estão dispostas em um espaço onde podem se encontrar de forma aleatória e assim, possivelmente, ocorrer interações inibitórias. A supressão é iniciada em uma célula selecionada aleatoriamente, que em seguida pode interagir com um determinado número de células e assim inibi-las com base em seus valores de *fitness*. Desta forma, espera-se que ocorra uma auto-organização da diversidade populacional, que pode ser especialmente útil em problemas multimodais. Adicionalmente, propõe-se uma diferente forma para adição de soluções aleatórias baseada no número de células suprimidas em substituição a utilização de um parâmetro fixo. O algoritmo proposto foi testado em funções de *benchmark* e seus resultados foram comparados com aqueles do opt-aiNet utilizando o teste não-paramétrico de Wilcoxon (*signed-rank Wilcoxon test*).

Palavras-chave – Computação Evolutiva, Sistemas Imunológicos Artificiais, Criticalidade Auto-Organizada, *benchmark*.

1 Introdução

Sistemas Imunológicos Artificiais (*Artificial Immune Systems* – AIS) representam uma classe de Algoritmos Evolutivos (AE), que diferentemente dos métodos tradicionais, inspirados na evolução das espécies, são inspirados na microevolução do sistema imunológico de organismos complexos [1]. Considerando o desenvolvimento do sistema imunológico adaptativo como um processo de otimização, onde suas células adaptam os anticorpos para o reconhecimento dos antígenos buscando inibir a ação dos patógenos, De Castro e Timmis [2] desenvolveram um AIS para otimização de funções multimodais contínuas: o opt-aiNet.

Um problema comum em AE é a convergência prematura da população para ótimos locais, quando se tratando de problemas multimodais. Esta característica torna importante o controle da diversidade populacional, que permite uma melhor exploração do espaço de busca [3], [4], [5]. Para este fim, o opt-aiNet adota uma estratégia inspirada na teoria de redes imunológicas, considerando que as células do sistema imunológico podem suprimir umas as outras. Esta técnica realiza a supressão de soluções semelhantes quando detectada a convergência da população. A avaliação da semelhança dos indivíduos é feita através do cálculo da distância euclidiana. Uma solução é suprimida na detecção de um par que possui a distância menor que um limiar σ_s , parâmetro do opt-aiNet. Após a supressão uma porcentagem de soluções aleatórias é adicionada à população, aumentando a diversidade e permitindo um controle do tamanho populacional. Quando o processo de supressão é aplicado, o custo para computar as distâncias entre todas as células é $O(N^2D)$, sendo N o tamanho da população e D o número de dimensões do problema. Tal atividade é custosa computacionalmente e seu desempenho é dependente de uma definição adequada dos parâmetros envolvidos.

A fim de evitar a convergência para ótimos locais, mantendo a diversidade populacional, e de diminuir a influência da seleção de parâmetros, recentemente diversas estratégias têm sido estudadas em AEs [6], [7], [8], [9], inspiradas no fenômeno conhecido como criticalidade auto-organizada (*Self-Organizing Criticality* – SOC) [10]. Em [9], Tinós e Yang propõem o SORIGA, (*Self-Organizing Random Immigrants Genetic Algorithm*) que utiliza uma técnica de imigrantes aleatórios para a manutenção de diversidade da população em Algoritmos Genéticos (AG), onde, em cada geração o indivíduo de pior *fitness* e seus vizinhos próximos são substituídos por novas soluções geradas aleatoriamente. Com o emprego desta técnica foi observado que, em situações em que a população se encontra com baixa diversidade, a estratégia de substituição causa uma reação em cadeia substituindo grande parte da população por imigrantes aleatórios, aumentando assim a diversidade.

Pode-se observar que os mecanismos de supressão e criação de células aleatórias do opt-aiNet e a técnica de imigrantes aleatórios em AGs utilizam de princípios semelhantes para a manutenção da diversidade. Neste trabalho, buscando diminuir o custo computacional da etapa de supressão do opt-aiNet, e amenizar a influência da escolha dos parâmetros, é proposta, em substituição, uma estratégia auto-organizável inspirada no fenômeno de SOC com base nas idéias aplicadas no SORIGA. A técnica é aqui chamada de SOCopt-aiNet.

Considerando a grande variedade de tipos e operadores aplicados a algoritmos de otimização, torna-se importante a utilização de um conjunto de regras para a comparação justa das diferentes técnicas aplicadas. O SOCOpt-aiNet, aqui proposto, foi comparado com opt-aiNet original utilizando o *benchmark* descrito em [11]. Este fornece um conjunto de funções para avaliação de algoritmos de otimização contínua, além de orientações para a comparação entre diferentes técnicas.

O fenômeno de SOC é introduzido na Seção 2. Na Seção 3 é apresentado o opt-aiNet como foi implementado para a comparação dos resultados com o SOCOpt-aiNet, que é descrito na Seção 4. A Seção 5 descreve os experimentos realizados e os resultados obtidos. Por fim, na Seção 6, é realizada uma discussão do trabalho.

2 Criticalidade auto-organizada

Em [10], Bak descreve o fenômeno de SOC, que pode ocorrer em sistemas com diversos componentes que interagem entre si. Em sistemas que apresentam SOC, a resposta a perturbações não pode ser descrita por um valor médio. Desta forma, uma pequena perturbação pode causar pequenas alterações no sistema ou, em um estado crítico, pode causar uma reação em cadeia reorganizando um grande número de componentes retornando o sistema a um estado estável.

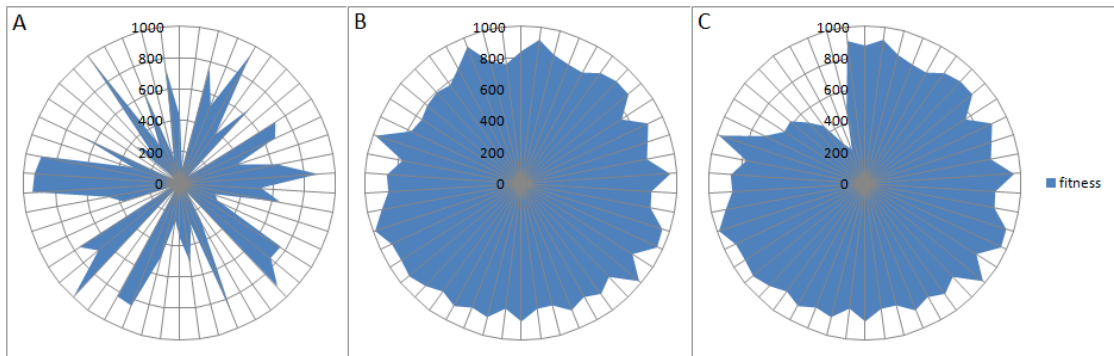


Figura 1 – Exemplo do desenvolvimento do modelos de interação entre espécies.

Um modelo que apresenta SOC é o de interação entre espécies [10] representado na Figura 1. Neste modelo considera-se um vetor circular representando um conjunto de espécies (linhas radiais cinzas) cada uma com um valor de *fitness* (em azul). No instante inicial (A), cada espécie possui um valor aleatório de *fitness*. Em cada instante de tempo, a espécie com menor *fitness* e as duas adjacentes no vetor circular são substituídas por novas espécies com *fitness* aleatório, evento chamado de extinção. Com o passar do tempo, o *fitness* médio tende a aumentar e o desvio padrão diminuir, indicando a convergência e a chegada ao estado crítico (B), onde as espécies têm valores de *fitness* próximos. Nesse ponto, as espécies são comumente substituídas por outras com *fitness* menor. Desta forma em passos seguintes de tempo é mais provável que substituições subsequentes ocorram em espécies em posições substituídas em instantes anteriores, caracterizando uma reação em cadeia que pode levar a um aumento da extinção. Na Figura 1 (C) nota-se a ocorrência de uma reação em cadeia que levou a substituição de um total de oito espécies do conjunto.

As distribuições do tamanho e duração da reação (extinção) podem ser descritas por leis de potência nas formas

$$P(s) \sim s^{-\tau}, \text{ e } P(d) \sim d^{-\alpha} \quad (1)$$

sendo s o número de componentes afetados pela perturbação, d a duração da reação em cadeia, e τ e α constantes reais positivas.

A técnica de manutenção da diversidade do SORIGA [9] é inspirada no modelo de interação entre espécies. Neste, ao invés de selecionar os indivíduos de pior *fitness*, ou indivíduos selecionados aleatoriamente para substituição, como ocorre nas técnicas tradicionais de imigrantes aleatórios, esta técnica considera a população de indivíduos do AG disposta em um vetor circular e, em cada geração, o indivíduo de pior *fitness* e os adjacentes são substituídos. Com a utilização desta estratégia, quando a população converge para um ótimo local, possuindo uma baixa diversidade, uma simples substituição pode levar a uma reação em cadeia em gerações seguintes, substituindo um grande número de indivíduos. Note que a estratégia, embora simples, pode apresentar SOC, controlando a diversidade da população, sem a necessidade de uma intervenção explícita.

3 O opt-aiNet

O opt-aiNet é baseado em três teorias dos sistema imunológico adaptativo: a seleção clonal, a maturação de afinidade e redes imunológicas. Considerando que a população de soluções candidatas é um conjunto de células de memória do sistema imunológico, o princípio de seleção clonal, no opt-aiNet, busca gerar clones e selecionar aqueles com melhor *fitness* para compor a geração seguinte. Na produção dos clones, estes sofrem mutações adaptativas com altas taxas, mas com intensidades inversamente proporcionais ao *fitness* da célula mãe (considerando a utilização em problemas de minimização). A teoria de redes imunológicas indica que no sistema imunológico podem ocorrer interações entre células do próprio sistema, inclusive de forma inibitória, por exemplo, para controle da proliferação, formando uma rede de células [1]. Esta teoria é aplicada na etapa de supressão do algoritmo, buscando eliminar soluções semelhantes.

O Quadro 1 apresenta o pseudo-código do opt-aiNet. O princípio de seleção clonal é aplicado na geração das N_c células clone de cada célula de memória, e na seleção da melhor diferenciando-a, substituindo a célula mãe, tornando-se a nova célula de memória (linhas 5 a 9). No caso em que a célula mãe continua representando a melhor solução, não ocorre diferenciação.

A maturação de afinidade é aplicada na mutação das células clone (linha 6). A mutação é realizada em todos os genes da célula adicionando um valor aleatório, na forma:

$$\mathbf{c}' = \mathbf{c} + \alpha \mathbf{R}_g \quad (2)$$

sendo \mathbf{c}' a célula mutada, \mathbf{c} a célula clone original, \mathbf{R}_g um vetor de valores aleatórios com distribuição gaussiana de média zero e desvio padrão 1 e

$$\alpha = \left(\frac{1}{\beta}\right) \exp(-f_n(\mathbf{c})) \quad (3)$$

onde β é o parâmetro de controle do decaimento da exponencial que controla a pressão da maturação de afinidade da célula e f_n o *fitness* normalizado para o intervalo [0,1], sendo que a melhor célula recebe $f_n=1$. Desta forma a exponencial terá seu valor máximo na mutação da pior solução que será mutada mais intensivamente.

A teoria de redes imunológicas é aplicada na fase de supressão (linhas 10 a 14). Esta etapa ocorre na detecção de estagnação, ou convergência, do *fitness* médio da população. Na implementação utilizada a estagnação ocorre na seguinte condição:

$$\frac{|\bar{f} - \bar{f}_0|}{|\bar{f}_0|} < l_c \quad (4)$$

onde \bar{f} é a média da geração atual, \bar{f}_0 , considerando que a verificação em (4) ocorre em intervalos de g_s gerações, é a média dos *fitness* médios de cada geração no dado intervalo, e l_c o limiar de convergência para que ocorra a supressão.

Quadro 1 – Pseudo-código do opt-aiNet [2].

```

1  Início
2      gere população inicial aleatoriamente
3      enquanto critério de convergência não satisfeito faça
4          determine fitness de cada célula de memória
5          para cada célula de memória faça
6              gere  $N_c$  clones mutados proporcional ao fitness
7              determine fitness de cada clone
8              diferencie clone com melhor fitness para célula
                                     de memória
9          fim para
10         se fitness médio não alterou significativamente então
11             determine afinidade para cada par de células
12             suprima células com afinidade abaixo do limiar  $\sigma_s$ 
13             introduza  $r\%$  de células geradas aleatoriamente
14         fim se
15     fim enquanto
16 Fim
    
```

No caso em que ocorre a estagnação, e consequentemente a supressão, é determinada a afinidade entre cada possível par de células de memória. A afinidade entre as células de memórias é um critério de similaridade entre as células. O algoritmo padrão, porém, utiliza a distância euclidiana para tanto, um critério de dissimilaridade. Desta forma as células suprimidas são aquelas que possuem a distância abaixo do limiar. No caso do uso de um critério de similaridade a afinidade deve ser maior que o limiar. Ao final uma porcentagem r de células geradas aleatoriamente é adicionada à população de células de memória.

4 Algoritmo proposto

A proposta deste trabalho é a utilização de um método auto-organizável, inspirado no modelo de interação entre espécies [10], em substituição à etapa de supressão e adição de soluções aleatórias (linhas 10 a 14 do Quadro 1) utilizados no opt-aiNet. O restante do algoritmo (os processos de seleção clonal e maturação de afinidade) é mantido inalterado.

O algoritmo de supressão do opt-aiNet original assume que cada célula interage com todas as outras em determinado instante. Pode-se argumentar que a técnica proposta faz mais sentido biológico, uma vez que considera que as células do sistema imunológico estão distribuídas aleatoriamente pelo espaço se movimentando de forma casual, o que deve permitir que elas se encontrem e interajam com qualquer outra de forma aleatória.

No Quadro 2 é apresentado em forma de pseudo-código o algoritmo de supressão proposto. Considerando-se a população de células dispostas aleatoriamente no espaço, de forma que podem se encontrar ao acaso, quando a etapa supressão

ocorre, uma célula de índice i (c_i) é selecionada aleatoriamente na população para iniciar a supressão. Em seguida, duas outras células são selecionadas aleatoriamente, como se estivessem se encontrado ao acaso (linha 2). Uma verificação é feita no *fitness* de cada uma das células encontradas verificando se os valores não são significativamente diferentes do *fitness* da célula c_i (linha 4). A verificação é realizada de forma semelhante à equação (4) substituindo \bar{f} pelo *fitness* de uma das duas células encontradas, e \bar{f}_0 pelo da célula de índice i . Caso o valor resultante seja menor que um limiar de supressão σ_s , então, diz-se que a célula verificada é selecionada para supressão (linha 5). Cada célula selecionada para supressão pode interagir recursivamente com duas outras células até que não hajam outras com valores próximos de *fitness*, de acordo com o critério descrito (linha 7). Por fim todos os indivíduos selecionados para supressão e o indivíduo de índice i são suprimidas (linha 10). Para que a melhor solução seja sempre mantida, o indivíduo com melhor *fitness* na população corrente é sempre mantido.

Diferente do algoritmo original, a etapa de supressão descrita é aplicada a cada intervalo g_s de gerações sem a verificação de convergência da população. Esta somente ocorre na etapa de adição de soluções aleatórias. Note que no algoritmo de supressão apresentado no Quadro 2 é computado o tamanho *ext_size* da extinção (linha 6), ou seja, o número de células suprimidas no processo. Este valor é utilizado para determinar o número de imigrantes aleatórios a serem inseridos na população. Caso o *fitness* médio da população tenha convergido, de acordo com o critério da equação (4), são adicionados *ext_size%* de soluções aleatórias na população. Se o número a ser adicionado for menor que o tamanho da extinção, então são adicionadas soluções aleatórias suficientes para que o tamanho da população se mantenha constante. A estratégia utilizada nesta etapa, busca permitir o controle do tamanho da população para auxiliar a busca de novas regiões promissoras.

Quadro 2 – Algoritmo de supressão proposto.

```
1  Início supressão (célula c)
2      selecione aleatoriamente duas células
3      para cada célula  $c_s$  selecionada faça
4          se  $f(c)$  significativamente próximo de  $f(c_s)$  então
5              selecione célula  $c_s$  para supressão
6               $ext\_size \leftarrow ext\_size + 1$ 
7              supressão ( $c_s$ ) //chamada recursiva
8          fim se
9      fim para
10     suprima células selecionadas para supressão
11 Fim supressão
```

Pode-se observar que o caráter recursivo da etapa de supressão descrita possui grande semelhança com a reação em cadeia que ocorre na técnica utilizada no SORIGA [9]. Portanto, espera-se que com a incorporação deste algoritmo de supressão e adição de soluções aleatórias, ocorra um controle da diversidade da população de forma auto-organizada, o que deve permitir uma melhor exploração do espaço de busca, em especial em funções multimodais, e possivelmente uma diminuição da complexidade do algoritmo. Considerando que o desempenho decepcionante de AEs muitas vezes pode ser justificado pela escolha inadequada dos parâmetros a presença da auto-organização no processo de otimização deve ainda diminuir a influência da escolha de parâmetros, uma vez que esta ocorre de acordo com o desenvolvimento da população no problema.

5 Resultados

Os resultados do algoritmo proposto foram comparados com uma implementação do opt-aiNet baseada em [2] e [12], e para verificar a significância estatística dos resultados, um teste não-paramétrico de Wilcoxon (*signed-rank Wilcoxon test*), com nível de significância de 5%, foi realizado entre os algoritmos, opt-aiNet e o algoritmo SOCOpt-aiNet.

A avaliação do algoritmo proposto foi realizada como descrito em [11], onde é fornecido um conjunto de funções para *benchmark* de otimização contínua. Nele são definidas 25 funções de avaliação divididas em funções unimodais (f1 a f5), multimodais básicas (f6 a f12), funções expandidas (f13 e f14) e composição de funções (f15 a f25). Todas são escaláveis e devem ser executadas para as dimensões $D = \{2, 10, 30, 50\}$. Desta forma cada problema é definido por uma função e o número de dimensões. O algoritmo deve ser executado 25 vezes para cada problema utilizando diferentes sementes aleatórias para inicialização. Considerando que as populações dos algoritmos podem ter tamanho variável, a cada algoritmo é permitido um número limitado de avaliações igual a $10000D$. Neste trabalho foram utilizadas somente as funções unimodais, multimodais básicas e expandidas (f1 a f14), totalizando 56 casos de teste. Na Tabela 1 estão descritas as funções e suas características.

Os parâmetros atribuídos a ambos os algoritmos são:

- Número inicial de células (soluções candidatas): $N=20$;
- Número de clones gerados por cada célula: $N_c=10$;
- Pressão da maturação de afinidade: $\beta=1$;

os parâmetros específicos do opt-aiNet:

- Limiar de convergência: $l_c=0,0001$;

- Limiar de supressão: $\sigma_s=0,2$;
- Porcentagem de soluções aleatórias adicionadas: $r=40\%$;
- Intervalo de supressão: $g_s=5$.

e por fim os parâmetros do algoritmo proposto:

- Limiar de convergência: $l_c=0,001$;
- Limiar de supressão: $\sigma_s=0,05$;
- Intervalo de supressão: $g_s=25$.

Tabela 1 – Funções de teste e suas características. Sendo $f(\mathbf{x}^*)$ o ótimo global, e intervalo os possíveis valores de \mathbf{x}

Função	Nome	$f(\mathbf{x}^*)$	Intervalo	Propriedades ¹
f1	<i>Shifted sphere function</i>	-450	[-100, 100]	U, S
f2	<i>Shifted Schwefel's Problem 1.2</i>	-450	[-100, 100]	U, NS
f3	<i>Shifted rotated high conditioned elliptic function</i>	-450	[-100, 100]	U, NS, RO
f4	<i>Shifted Schwefel's Problem 1.2 with noise in fitness</i>	-450	[-100, 100]	U, NS, RU
f5	<i>Schwefel's Problem 2.6 with global optimum on bounds</i>	-310	[-100, 100]	U, NS
f6	<i>Shifted Rosenbrock's function</i>	390	[-100, 100]	M, NS
f7	<i>Shifted rotated Griewank's function without bounds</i>	-180	$[-\infty, \infty]$	M, NS, RO
f8	<i>Shifted rotated Ackley's function with global optimum on bounds</i>	-140	[-32, 32]	M, NS, RO
f9	<i>Shifted Rastrigin's function</i>	-330	[-5, 5]	M, S
f10	<i>Shifted rotated Rastrigin's function</i>	-330	[-5, 5]	M, NS, RO
f11	<i>Shifted rotated Weierstrass' function</i>	90	[-0.5, 0.5]	M, NS, RO
f12	<i>Schwefel's Prob. 2.13</i>	-460	$[-\pi, \pi]$	M, NS
f13	<i>Shifted expanded Griewank's plus Rosenbrock's function</i>	-130	[-3, 1]	M, NS
f14	<i>Shifted rotated expanded Schaffer's F6 function</i>	-300	[-100, 100]	M, NS

¹U unimodal, M multimodal, S separável, NS não separável, RO rotacionada, RU com ruído

Os resultados obtidos estão nas tabelas 2, 3, 4 e 5 referentes às funções de dimensão 2, 10, 30 e 50 respectivamente. São apresentadas as médias, desvios padrão, medianas e o melhor resultado, isto é, o mínimo dos erros obtidos pelos algoritmos nas diferentes execuções, além do p-valor do teste de Wilcoxon com nível de significância 5%. Dado que as funções de *benchmark* têm seus valores ótimos conhecidos, os erros são computados simplesmente pelo valor absoluto da diferença do resultado obtido com o valor do ótimo da função. Nas colunas do p-valor estão indicados os casos em que o SOCOpt-aiNet superou o opt-aiNet (+), ou foi superado (-). Casos em que a diferença tem relevância estatística estão marcados com (S).

Tabela 2 – Resultados referentes aos problemas de dimensão 2.

Função	Algoritmo	Média	Desvio padrão	Mediana	Mínimo	p-valor
f1	<i>opt-aiNet</i>	5,56E-04	1,67E-03	1,56E-04	2,02E-05	4,58E-01 (+)
	<i>SOCOopt-aiNet</i>	7,05E-04	2,35E-03	1,51E-04	3,94E-06	
f2	<i>opt-aiNet</i>	3,45E-04	3,59E-04	1,43E-04	2,54E-05	2,78E-01 (-)
	<i>SOCOopt-aiNet</i>	3,24E-04	4,14E-04	1,97E-04	4,38E-06	
f3	<i>opt-aiNet</i>	2,40E+02	3,37E+02	1,60E+02	8,73E+00	5,45E-01 (-)
	<i>SOCOopt-aiNet</i>	2,66E+02	3,52E+02	1,79E+02	3,16E+00	
f4	<i>opt-aiNet</i>	1,88E+02	5,04E+02	7,76E-03	4,12E-05	6,38E-01 (+)
	<i>SOCOopt-aiNet</i>	1,64E+02	4,89E+02	2,90E-03	8,18E-05	
f5	<i>opt-aiNet</i>	2,32E+02	4,92E+02	1,38E+00	9,79E-02	1,58E-01 (+)
	<i>SOCOopt-aiNet</i>	2,64E+02	5,31E+02	9,78E-01	7,29E-02	
f6	<i>opt-aiNet</i>	8,80E+00	1,90E+01	1,06E-02	1,68E-05	6,19E-01 (-)
	<i>SOCOopt-aiNet</i>	7,97E+00	1,86E+01	1,73E-02	8,52E-04	
f7	<i>opt-aiNet</i>	1,07E+02	2,92E+01	1,05E+02	6,59E+01	6,00E-01 (-)
	<i>SOCOopt-aiNet</i>	1,07E+02	3,02E+01	1,07E+02	6,59E+01	
f8	<i>opt-aiNet</i>	1,45E+01	6,04E+00	1,68E+01	1,38E+00	3,91E-01 (+)
	<i>SOCOopt-aiNet</i>	1,34E+01	6,75E+00	1,55E+01	1,67E+00	
f9	<i>opt-aiNet</i>	8,24E-03	9,82E-03	5,62E-03	2,38E-04	1,02E-03 (-)
	<i>SOCOopt-aiNet</i>	3,60E-02	4,59E-02	1,90E-02	1,60E-03	
f10	<i>opt-aiNet</i>	1,46E-02	1,46E-02	1,01E-02	7,02E-04	2,13E-03 (-)
	<i>SOCOopt-aiNet</i>	4,09E-02	4,22E-02	1,69E-02	5,19E-03	
f11	<i>opt-aiNet</i>	1,50E-01	4,39E-02	1,48E-01	8,08E-02	1,66E-01 (-)
	<i>SOCOopt-aiNet</i>	1,70E-01	6,08E-02	1,79E-01	3,81E-02	

f12	<i>opt-aiNet</i>	7,16E-02	6,74E-02	4,57E-02	1,33E-02	8,31E-01 (+)
	<i>SOCopt-aiNet</i>	6,86E-02	6,61E-02	4,40E-02	1,27E-03	
f13	<i>opt-aiNet</i>	6,16E-05	7,43E-05	4,10E-05	2,00E-08	2,53E-01 (-)
	<i>SOCopt-aiNet</i>	7,84E-05	8,45E-05	4,62E-05	3,00E-08	
f14	<i>opt-aiNet</i>	2,98E-01	2,09E-01	2,61E-01	1,94E-02	1,23E-01 (-)
	<i>SOCopt-aiNet</i>	4,13E-01	2,33E-01	4,55E-01	1,94E-02	

Tabela 3 – Resultados referentes aos problemas de dimensão 10.

Função	Algoritmo	Média	Desvio padrão	Mediana	Mínimo	p-valor
f1	<i>opt-aiNet</i>	1,03E-01	2,89E-02	1,00E-01	4,99E-02	4,46E-04 (S-)
	<i>SOCopt-aiNet</i>	1,41E-01	2,73E-02	1,42E-01	6,68E-02	
f2	<i>opt-aiNet</i>	2,58E-01	8,78E-02	2,46E-01	1,20E-01	2,54E-05 (S-)
	<i>SOCopt-aiNet</i>	6,55E+00	2,12E+01	7,91E-01	1,62E-01	
f3	<i>opt-aiNet</i>	1,15E+05	5,84E+04	1,16E+05	1,37E+04	2,96E-04 (S-)
	<i>SOCopt-aiNet</i>	1,95E+05	1,03E+05	1,79E+05	4,32E+04	
f4	<i>opt-aiNet</i>	1,89E+04	9,04E+03	1,69E+04	5,13E+03	4,43E-01 (-)
	<i>SOCopt-aiNet</i>	1,64E+04	8,11E+03	1,72E+04	4,37E+03	
f5	<i>opt-aiNet</i>	1,47E+03	7,32E+02	1,34E+03	6,68E+01	3,62E-05 (S-)
	<i>SOCopt-aiNet</i>	2,43E+03	1,09E+03	2,26E+03	4,95E+02	
f6	<i>opt-aiNet</i>	1,94E+01	3,04E+00	1,98E+01	1,22E+01	7,33E-04 (S-)
	<i>SOCopt-aiNet</i>	2,68E+01	9,87E+00	2,44E+01	1,67E+01	
f7	<i>opt-aiNet</i>	2,31E+03	4,15E+02	2,45E+03	1,35E+03	1,09E-01 (+)
	<i>SOCopt-aiNet</i>	2,26E+03	4,08E+02	2,24E+03	1,35E+03	
f8	<i>opt-aiNet</i>	2,03E+01	7,94E-02	2,03E+01	2,02E+01	2,83E-02 (S-)
	<i>SOCopt-aiNet</i>	2,04E+01	4,80E-02	2,04E+01	2,03E+01	
f9	<i>opt-aiNet</i>	3,41E+01	4,01E+00	3,34E+01	2,51E+01	6,53E-02 (-)
	<i>SOCopt-aiNet</i>	3,62E+01	3,56E+00	3,59E+01	3,04E+01	
f10	<i>opt-aiNet</i>	3,31E+01	4,45E+00	3,44E+01	2,34E+01	7,98E-01 (-)
	<i>SOCopt-aiNet</i>	3,32E+01	4,55E+00	3,47E+01	2,29E+01	
f11	<i>opt-aiNet</i>	8,31E+00	5,48E-01	8,36E+00	6,72E+00	3,03E-02 (S-)
	<i>SOCopt-aiNet</i>	8,65E+00	5,22E-01	8,57E+00	7,46E+00	
f12	<i>opt-aiNet</i>	2,10E+03	8,49E+02	1,88E+03	8,91E+02	5,78E-02 (+)
	<i>SOCopt-aiNet</i>	1,76E+03	6,79E+02	1,72E+03	8,51E+02	
f13	<i>opt-aiNet</i>	3,18E+00	5,00E-01	3,36E+00	1,86E+00	3,82E-03 (S-)
	<i>SOCopt-aiNet</i>	3,71E+00	4,84E-01	3,78E+00	2,82E+00	
f14	<i>opt-aiNet</i>	3,91E+00	2,11E-01	3,92E+00	3,34E+00	8,40E-01 (-)
	<i>SOCopt-aiNet</i>	3,90E+00	2,28E-01	3,93E+00	3,31E+00	

Tabela 4 – Resultados referentes aos problemas de dimensão 30.

Função	Algoritmo	Média	Desvio padrão	Mediana	Mínimo	p-valor
f1	<i>opt-aiNet</i>	1,76E+00	1,42E-01	1,79E+00	1,41E+00	2,70E-03 (S-)
	<i>SOCopt-aiNet</i>	1,96E+00	2,33E-01	1,97E+00	1,45E+00	
f2	<i>opt-aiNet</i>	1,24E+01	3,40E+00	1,14E+01	7,31E+00	1,39E-05 (S-)
	<i>SOCopt-aiNet</i>	2,36E+01	6,42E+00	2,20E+01	1,29E+01	
f3	<i>opt-aiNet</i>	1,70E+06	3,84E+05	1,79E+06	9,16E+05	2,54E-05 (S-)
	<i>SOCopt-aiNet</i>	2,56E+06	7,69E+05	2,27E+06	1,14E+06	
f4	<i>opt-aiNet</i>	8,95E+04	1,89E+04	9,08E+04	5,42E+04	9,68E-01 (+)
	<i>SOCopt-aiNet</i>	8,93E+04	2,22E+04	9,03E+04	5,14E+04	
f5	<i>opt-aiNet</i>	5,87E+03	5,13E+02	5,87E+03	4,89E+03	5,76E-05 (S-)
	<i>SOCopt-aiNet</i>	7,77E+03	1,51E+03	7,81E+03	4,44E+03	
f6	<i>opt-aiNet</i>	2,49E+02	3,41E+01	2,48E+02	1,81E+02	8,09E-05 (S-)
	<i>SOCopt-aiNet</i>	3,96E+02	1,77E+02	3,26E+02	2,62E+02	
f7	<i>opt-aiNet</i>	4,85E+03	2,77E+02	4,82E+03	4,34E+03	1,23E-05 (+)

	<i>SOCopt-aiNet</i>	6,41E+03	4,54E+02	6,37E+03	5,39E+03	
f8	<i>opt-aiNet</i>	2,10E+01	5,46E-02	2,10E+01	2,08E+01	1,74E-01 (+)
	<i>SOCopt-aiNet</i>	2,10E+01	4,34E-02	2,10E+01	2,08E+01	
f9	<i>opt-aiNet</i>	2,40E+02	1,07E+01	2,39E+02	2,15E+02	8,27E-02 (+)
	<i>SOCopt-aiNet</i>	2,30E+02	1,75E+01	2,32E+02	1,94E+02	
f10	<i>opt-aiNet</i>	2,29E+02	1,50E+01	2,31E+02	1,99E+02	8,61E-01 (+)
	<i>SOCopt-aiNet</i>	2,28E+02	1,59E+01	2,29E+02	2,00E+02	
f11	<i>opt-aiNet</i>	3,94E+01	8,74E-01	3,94E+01	3,79E+01	8,19E-01 (+)
	<i>SOCopt-aiNet</i>	3,92E+01	1,32E+00	3,92E+01	3,57E+01	
f12	<i>opt-aiNet</i>	1,47E+05	1,96E+04	1,44E+05	1,11E+05	5,82E-03 (+)
	<i>SOCopt-aiNet</i>	1,32E+05	1,33E+04	1,31E+05	1,02E+05	
f13	<i>opt-aiNet</i>	2,30E+01	1,38E+00	2,33E+01	1,95E+01	8,61E-01 (+)
	<i>SOCopt-aiNet</i>	2,31E+01	1,56E+00	2,32E+01	2,03E+01	
f14	<i>opt-aiNet</i>	1,35E+01	1,74E-01	1,35E+01	1,31E+01	2,31E-01 (+)
	<i>SOCopt-aiNet</i>	1,34E+01	2,40E-01	1,34E+01	1,29E+01	

Tabela 5 – Resultados referentes aos problemas de dimensão 50.

Função	Algoritmo	Média	Desvio padrão	Mediana	Mínimo	p-valor
f1	<i>opt-aiNet</i>	5,25E+00	3,83E-01	5,29E+00	4,47E+00	4,93E-04 (S-)
	<i>SOCopt-aiNet</i>	5,93E+00	5,96E-01	5,96E+00	4,43E+00	
f2	<i>opt-aiNet</i>	1,14E+02	1,73E+01	1,13E+02	7,55E+01	1,23E-05 (S-)
	<i>SOCopt-aiNet</i>	1,93E+02	3,80E+01	1,85E+02	1,19E+02	
f3	<i>opt-aiNet</i>	3,95E+06	6,31E+05	3,82E+06	2,56E+06	7,22E-05 (S-)
	<i>SOCopt-aiNet</i>	5,07E+06	8,45E+05	5,05E+06	3,82E+06	
f4	<i>opt-aiNet</i>	1,84E+05	3,18E+04	1,77E+05	1,44E+05	1,10E-02 (S-)
	<i>SOCopt-aiNet</i>	2,16E+05	5,90E+04	2,19E+05	1,15E+05	
f5	<i>opt-aiNet</i>	1,27E+04	8,88E+02	1,27E+04	1,05E+04	2,96E-04 (S-)
	<i>SOCopt-aiNet</i>	1,45E+04	2,02E+03	1,44E+04	1,09E+04	
f6	<i>opt-aiNet</i>	8,84E+02	1,09E+02	8,64E+02	6,25E+02	2,70E-03 (S-)
	<i>SOCopt-aiNet</i>	1,21E+03	4,86E+02	1,04E+03	7,18E+02	
f7	<i>opt-aiNet</i>	4,08E+03	1,98E+02	4,12E+03	3,74E+03	1,23E-05 (S-)
	<i>SOCopt-aiNet</i>	7,05E+03	3,88E+02	7,11E+03	6,08E+03	
f8	<i>opt-aiNet</i>	2,11E+01	3,32E-02	2,11E+01	2,10E+01	6,15E-02 (-)
	<i>SOCopt-aiNet</i>	2,11E+01	3,67E-02	2,11E+01	2,10E+01	
f9	<i>opt-aiNet</i>	5,08E+02	2,42E+01	5,08E+02	4,52E+02	4,22E-02 (S+)
	<i>SOCopt-aiNet</i>	4,94E+02	3,03E+01	5,01E+02	4,24E+02	
f10	<i>opt-aiNet</i>	4,97E+02	1,91E+01	4,93E+02	4,69E+02	4,12E-01 (+)
	<i>SOCopt-aiNet</i>	4,94E+02	2,10E+01	4,92E+02	4,61E+02	
f11	<i>opt-aiNet</i>	7,27E+01	1,46E+00	7,29E+01	6,89E+01	9,46E-01 (-)
	<i>SOCopt-aiNet</i>	7,25E+01	1,93E+00	7,30E+01	6,80E+01	
f12	<i>opt-aiNet</i>	8,47E+05	6,93E+04	8,42E+05	7,10E+05	1,49E-02 (S+)
	<i>SOCopt-aiNet</i>	7,81E+05	9,79E+04	7,92E+05	5,91E+05	
f13	<i>opt-aiNet</i>	5,10E+01	2,57E+00	5,17E+01	4,56E+01	8,40E-01 (+)
	<i>SOCopt-aiNet</i>	5,10E+01	2,62E+00	5,16E+01	4,37E+01	
f14	<i>opt-aiNet</i>	2,31E+01	2,38E-01	2,31E+01	2,24E+01	1,58E-01 (+)
	<i>SOCopt-aiNet</i>	2,30E+01	2,83E-01	2,31E+01	2,23E+01	

Em 20, de um total de 56 casos de teste, o algoritmo SOCopt-aiNet superou o opt-aiNet dos quais 2 apresentaram diferença significativa pelo teste estatístico. Dos demais, o opt-aiNet o superou em 36 dos casos dos quais 24 apresentaram significância estatística. Embora na maioria dos casos, o opt-aiNet apresente melhores resultados em geral, com o aumento da complexidade das funções nota-se uma aparente tendência do SOCopt-aiNet apresentar melhor desempenho. Isto pode ser percebido pelo número de ocorrências em que este apresenta a mediana menor que do algoritmo original: dos 36 casos multimodais, em 17 o SOCopt-aiNet apresentou mediana menor quando comparado com o opt-aiNet, dos quais 13 são dos problemas de dimensão 30 e 50.

Em uma análise mais apurada dos resultados, nota-se que o algoritmo SOCOpt-aiNet apresenta melhores resultados nos problemas multimodais, e mesmo quando o opt-aiNet original o supera a diferença não é grande em geral. Porém, em algumas das funções multimodais, em f_6 , f_7 , ambos os algoritmos apresentaram resultados decepcionantes. Considerando isto, pode-se argumentar uma possível causa relacionada aos operadores utilizados.

Nas funções f_6 e f_7 , os gradientes são relativamente baixos, o que dificulta a busca dos algoritmos, uma vez que esta é feita utilizando somente o operador de mutação com distribuição normal, o que dificulta para as soluções darem grandes saltos. O operador de supressão e adição de soluções aleatórias permite o algoritmo a explorar novas regiões, porém, é difícil em problemas com baixo gradiente ocorrer convergência, tornando, nesses casos, incomum a adição de soluções aleatórias.

Durante os experimentos, também foram computados o tamanho da população em cada geração dos algoritmos. Foi verificado que o tamanho das populações no SOCOpt-aiNet não se altera significativamente quando comparado aos tamanhos das populações atingido pelo opt-aiNet, mesmo nos problemas multimodais, indicando que o critério de convergência é raramente atingido, um indício de que o intervalo de supressão que pode ser insuficiente dificultando a convergência.

6 Conclusões

A manutenção da diversidade populacional tem um papel de grande importância em algoritmos evolutivos, em especial na aplicação em funções multimodais, uma vez que deve promover a exploração do espaço de busca, evitando avaliações desnecessárias, por exemplo, examinando regiões não promissoras ou já exploradas.

Tanto o opt-aiNet como o SOCOpt-aiNet, proposto neste trabalho, apresentaram fraco desempenho em alguns problemas em que há um baixo gradiente quando se aproximando do ótimo global. Tal característica representa uma grande dificuldade para a otimização com os operadores de mutação utilizados. Uma possível forma para auxiliar neste problema é utilizando uma técnica de controle do parâmetro β . Este, com valores menores, deve permitir maiores saltos pelo espaço de busca, permitindo uma convergência mais rápida em problemas com baixo gradiente.

Por outro lado, o algoritmo proposto apresentou uma tendência a obter bons resultados nos problemas multimodais em geral. Considerando que o tempo de convergência é dependente da complexidade da função de *fitness*, em trabalhos futuros, pode-se utilizar uma diferente abordagem, por exemplo, pela definição de um intervalo de supressão dependente do número de dimensões do problema, ou até mesmo um controle dinâmico do parâmetro. Com um melhor critério de controle do intervalo de supressão, que deve possibilitar a convergência da população, pode-se diminuir também o valor da variável de limiar de supressão (σ_s) do SOCOpt-aiNet. Esta, com um valor alto, dificulta a convergência uma vez que elimina soluções com *fitness* dentro de um intervalo maior, além de facilitar a supressão de soluções que ainda não atingiram um ótimo local.

7 Referências

- [1] L. N. De Castro, J. I. Timmis, Artificial immune systems as a novel soft computing paradigm, **Soft Computing**, 7 (2003), 526-544.
- [2] L. N. De Castro, J. I. Timmis, An Artificial immune network for multimodal function optimization, **Evolutionary Computation**, 1 (2002), 699-704.
- [3] H. G. Cobb, J. J. Grefenstette, Genetic algorithms for tracking changing environments, **Proc. of the 5th Int. Conf. on Genetic Algorithms**. (1993), 523-530.
- [4] A. E. Eiben, R. Hinterding, Z. Michalewicz, Parameter control in evolutionary algorithms, **IEEE Trans. on Evol. Computation**, 3 (1999), 124-141.
- [5] F. Vavak, T. C. Fogarty, A comparative study of steady state and generational genetic algorithms for use in non-stationary environments, **Lecture Notes in Computer Science**. Springer. 1143 (1996), 297-304.
- [6] C. M. Fernandes, J. J. Merelo, V. Ramos, A. C. Rosa, A self-organized criticality mutation operator for dynamic optimization problems, **Proc. of the 10th Genetic and Evolutionary Computation**, Atlanta, Georgia, USA, (2008), 937-944.
- [7] T. Krink, P. Rickers, R. Thomsen, Applying self-organized criticality to evolutionary algorithms, **Proc. of the 6th international conference on parallel problem solving**, London, UK, (2000), 375-384.
- [8] T. Krink, P. Thomsen, Self-organized criticality and mass extinction in evolutionary algorithms, **Proc. of the 2001 congress on evolutionary computation**, Seoul, South Korea, 2 (2001), 1155-1161.
- [9] R. Tinós, S. Yang, Self-organizing genetic algorithms for dynamic optimization problems, **Genetic Programming and Evolvable Machines**, 8(2007), 255-286.
- [10] P. Bak, How nature works: the science of self-organized criticality, **Oxford University Press** (1997).
- [11] P. N. Suganthan, N. Hansen, J. J. Lian, K. Deb, Y. – P. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, **Technical report, Nanyang Technological University, Singapore**, (2005).
- [12] J. Timmis, C. Edmonds, A comment on opt-aiNet: an immune network algorithm for optimization, **Genetic and evolutionary computation (GECCO)**, 3012 (2004), 308-317.