

---

# COMPORTAMENTO AUTO-ORGANIZÁVEL EM ALGORITMOS GENÉTICOS APLICADOS A ROBÔS MÓVEIS EM AMBIENTES DINÂMICOS

**Renato Tinós\***

rtinos@ffclrp.usp.br

\*Departamento of Física e Matemática, FFCLRP - Universidade de São Paulo (USP)  
Ribeirão Preto, SP, 14040-901, Brasil

---

## ABSTRACT

In this paper, Genetic Algorithms (GAs) with changing fitness function applied to mobile robots are investigated. The use of GAs with Random Immigrants where the individual with the lowest fitness and its next neighbors are replaced in every generation is proposed to deal with such problems. In order to avoid that the individuals with the best fitness take the new individuals to extinction, these ones are preserved in a subpopulation. This simple approach can take the system to a self-organization behavior, which can be useful to maintain the diversity of the solutions and, then, to allow the GA to escape from local optima induced by the changes in the problem. Simulations of a mobile robot controlled by an Artificial Neural Network configured by the proposed GA are presented.

**KEYWORDS:** Genetic Algorithms, Mobile Robots, Self-Organization, Optimization.

## RESUMO

A aplicação de Algoritmos Genéticos (AGs) em problemas associados a robôs móveis e nos quais a função de aptidão se altera é investigado neste trabalho. Para lidar com tais problemas não-estacionários, o uso de AGs com Imigrantes

Aleatórios em que o indivíduo com a menor aptidão e seus vizinhos próximos são substituídos em cada geração é proposto. Para que melhores indivíduos da população não levem os novos indivíduos à extinção, estes são preservados em uma subpopulação. A estratégia de substituição apresentada pode levar o sistema a um comportamento auto-organizável, permitindo que o nível de diversidade da população aumente e que os indivíduos consigam escapar de ótimos locais induzidos pelas mudanças no robô ou no ambiente. A análise dos dados obtidos em simulações sugere que o AG investigado apresenta um tipo de comportamento auto-organizável conhecido como Criticalidade Auto-Organizada, o qual aparece em diversos fenômenos naturais. Simulações de um robô móvel controlado por uma Rede Neural Artificial cujos pesos são definidos pelo AG proposto são apresentadas.

**PALAVRAS-CHAVE:** Algoritmos Genéticos, Robôs Móveis, Auto-Organização, Otimização.

## 1 INTRODUÇÃO

Em Robótica, a utilização de algoritmos inspirados na evolução das espécies, conhecidos como Algoritmos Evolutivos, tem sido proposta, tanto para a otimização da morfologia, como para o desenvolvimento de leis de controle e de estratégias de navegação (Nolfi and Floreano, 2000). Algoritmos Evolutivos têm sido utilizados em Robótica principalmente porque o projeto de robôs autônomos e de seus controladores para ambientes não-estruturados, flexíveis, e/ou parcialmente desconhecidos é uma tarefa muito difícil para um

---

Artigo submetido em 15/03/2005

1a. Revisão em 31/07/2006

2a. Revisão em 27/02/2007

Aceito sob recomendação do Editor Associado

Prof. Ivan Nunes da Silva

projetista humano (Meyer, 1998). A dificuldade do projeto advém, principalmente, da impossibilidade de prever todas as situações as quais os robôs serão confrontados. Utilizando Algoritmos Evolutivos, o ambiente e a tarefa a ser executada passam a ser os fatores principais no desenvolvimento do robô e de seu controle, tirando o posto que antes cabia à muitas vezes restrigente mente do projetista.

Tal é a perspectiva do uso de Algoritmos Evolutivos em Robótica, que um termo foi proposto especialmente para designar os mecanismos robóticos por eles criados: Robôs Evolutivos (Nolfi and Floreano, 2000). Salienta-se que a conexão entre Robótica e Biologia não têm um sentido único neste caso: robôs autônomos podem servir como uma importante ferramenta para o desenvolvimento e teste de modelos de comportamentos adaptativos, de habilidades cognitivas e de modelos evolutivos dos sistemas biológicos.

As pesquisas em Robôs Evolutivos têm se concentrado quase que exclusivamente em problemas de otimização estacionários. No entanto, a aplicação de Robótica em ambientes flexíveis, não-estruturados, e/ou parcialmente desconhecidos exige que robôs consigam se adaptar às mudanças no ambiente, fazendo com que os problemas de otimização se tornem problemas não-estacionários. Tais mudanças representam variações nas restrições das soluções ou na função de aptidão (*fitness*) associada ao problema que ocorrem durante o processo de otimização. Diversos são os exemplos de mudanças que podem ocorrer em tais ambientes, como: ocorrência de falhas (Tinós et al., 2001), (Terra and Tinós, 2001), mudanças nas características do ambiente (Billard et al., 2000), cooperação e competição entre robôs (Nolfi and Floreano, 2000), mudança de tipo de robô (Floreano and Urzelai, 2000) e mudanças de estratégias.

Algumas soluções foram propostas para tratar dos problemas não-estacionários especificamente em Robôs Evolutivos. Podem ser citadas as técnicas: adição de ruído (Jakobi, 1997), avaliação da função de *fitness* para diferentes condições do ambiente (Thompson, 1998), combinação de Algoritmos Evolutivos e métodos de aprendizado (Nolfi and Floreano, 2000) e manutenção do nível de diversidade por meio de Imigrantes Aleatórios (Floreano et al., 2001). As duas primeiras técnicas implicam no conhecimento *a priori* das características do ambiente, o que muitas vezes é inviável. Já quando se utilizam técnicas de aprendizado associadas com Algoritmos Evolutivos, a capacidade de se alterar (plasticidade) do sistema pode ser reduzida graças ao "Efeito Baldwin", no qual os custos evolutivos associados ao aprendizado dão uma vantagem seletiva para a assimilação das propriedades aprendidas para os indivíduos que estão sendo evoluídos (Marley, 1996) *apud in* (Floreano and Urzelai, 2000).

O Algoritmo Genético (AG) com Imigrantes Aleatórios, o qual é inspirado no fluxo de indivíduos que entram e saem de uma população entre duas gerações na Natureza (Cobb and Grefenstette, 1993), é bastante simples e interessante. Em cada geração do processo de otimização, alguns indivíduos da população corrente são substituídos por indivíduos criados aleatoriamente. Uma estratégia de substituição, como, por exemplo, substituir indivíduos escolhidos aleatoriamente ou indivíduos com os menores valores de *fitness*, define quais indivíduos são substituídos. Através da introdução de novos indivíduos, o método dos Imigrantes Aleatórios tenta manter o nível de diversidade da população. A manutenção da diversidade da população pode ser bastante útil em problemas sujeitos a mudanças (Branke, 2003) para evitar que a população fique presa em ótimos locais (ou seja, soluções que são ótimas em suas vizinhanças locais) representados pelas soluções adotadas antes da mudança no problema.

Contudo, em alguns casos, como quando o número de genes em um indivíduo é alto e/ou quando os ótimos locais em que a população encontra-se têm valores de *fitness* muito maiores que os valores médios de todos os indivíduos possíveis do espaço de busca, a probabilidade que novos indivíduos sejam preservados é geralmente muito pequena. Isso ocorre porque o AG preserva, direta ou indiretamente, os melhores indivíduos da população corrente, e a probabilidade de que novos indivíduos tenham valores de *fitness* próximos ou maiores que os valores dos indivíduos atuais é geralmente baixa.

Neste trabalho, uma nova estratégia para o AG com Imigrantes Aleatórios é investigada em problemas não-estacionários envolvendo robôs móveis. No AG investigado, ao invés de substituir os indivíduos com os menores valores de *fitness* ou indivíduos escolhidos aleatoriamente dentro da população em cada geração como no método padrão dos Imigrantes Aleatórios, o indivíduo com o menor valor de *fitness* e seus dois vizinhos próximos (ou seja, com índices próximos ao índice do indivíduo com o menor valor de *fitness*) são substituídos. Utilizando tal estratégia, os indivíduos de uma população começam a interagir entre eles e, quando os valores de *fitness* dos indivíduos da população são próximos, como no caso em que o nível de diversidade é baixo, uma simples substituição de um indivíduo pode afetar um grande número de outros indivíduos da mesma população através de uma reação de substituições em cadeia. De modo a evitar que os melhores indivíduos da população levem os novos indivíduos à extinção, estes são preservados em uma subpopulação. O número de indivíduos nesta subpopulação não é definido pelo programador, mas sim pelo número de indivíduos criados na reação em cadeia, cujo tamanho depende de várias características da população, principalmente de seu nível de diversidade. É importante observar que esta simples estratégia pode levar o sistema a um comportamento

auto-organizável, o qual pode ser útil em problemas não-estacionários. Simulações de um robô móvel controlado por uma Rede Neural Artificial cujos pesos são definidos pelo AG proposto são apresentadas. Nestas simulações, falhas são introduzidas no robô e o ambiente em que o robô se encontra é alterado.

Os resultados obtidos em simulação apresentados na Seção 4 indicam que o algoritmo investigado, que é descrito na Seção 3, apresenta um tipo de comportamento auto-organizável conhecido como Criticalidade Auto-Organizada (Bak, 1997), o qual é comentado na Seção 2. As considerações finais encontram-se na Seção 5.

## 2 CRITICALIDADE AUTO-ORGANIZADA

Bak, Tang, e Wiesenfeld (Bak, 1997) sugeriram que sistemas constituídos de muitos componentes interagindo entre si podem apresentar um interessante tipo de comportamento auto-organizável (Jensen, 1998). Os autores nomearam tal comportamento como Criticalidade Auto-Organizada (*Self – Organized Criticality* - SOC) e sugeriram que ele estava presente em diversos fenômenos, como pilhas de areia, terremotos, panes em grandes sistemas elétricos e incêndios em florestas. A característica mais fascinante em sistemas que exibem SOC é que eles se auto-organizam em um determinado estado crítico sem a necessidade de qualquer ação de controle externo. Tal estado crítico é caracterizado pela resposta do sistema a perturbações externas. Em um sistema que exhibe comportamento não-crítico, a distribuição das respostas do sistema às perturbações em diferentes posições e instantes é estreita e bem descrita por um valor médio. Em um sistema que exhibe comportamento crítico, não existe uma única resposta característica. Em outras palavras, o sistema exhibe invariância de escala, sendo que uma pequena perturbação em um único componente pode gerar tanto um efeito pequeno apenas nos componentes da sua vizinhança, quanto uma reação em cadeia que afeta todos os componentes constituintes do sistema.

As distribuições estatísticas que descrevem a resposta de um sistema que exhibe SOC são dadas por leis de potência nas formas

$$P(s) \sim s^{-\tau} \quad (1)$$

e

$$P(d) \sim d^{-\alpha} \quad (2)$$

sendo  $s$  o número de componentes do sistema afetados pela perturbação,  $d$  a duração da reação em cadeia gerada pela perturbação, e  $\tau$  e  $\alpha$  constantes reais.

Como exemplo, considere o modelo da pilha de areia descrito em (Bak, 1997), no qual um grão de areia é adicionado em um posição aleatória em cada intervalo de tempo  $\Delta t$ . Com o

intuito de caracterizar a resposta do sistema, pode-se aferir o número de grãos de areia envolvidos em uma avalanche induzida pela adição de um único grão ( $s$ ), e a duração de cada avalanche ( $d$ ). No estado crítico, as distribuições estatísticas descrevendo a resposta do modelo da pilha de areia para a adição de um único grão são dadas pelas leis de potência apresentadas nas Eqs. 1 e 2, sendo que a adição de um grão em uma dada posição pode provocar desde um pequeno deslizamento que afeta apenas a sua vizinhança até um grande deslizamento que afeta toda a pilha de areia.

Nos últimos anos, pesquisadores sugeriram que a evolução natural pode, também, apresentar SOC (Bak, 1997). Uma evidência de tal afirmação seria o fato de que a evolução ocorre através de períodos curtos de atividade intensa intercalados por períodos calmos, ao invés de continuamente em passos constantes e lentos. Existem muito mais extinções pequenas (ou seja, que afetam poucas espécies) do que extinções grandes, como a grande extinção do Cretáceo que eliminou os dinossauros. Além disso, extinções ocorrem em uma grande variedade de escalas de tamanhos (Raup, 1986). Estes fatos sugerem que as extinções propagam-se através dos ecossistemas, tal qual avalanches em uma pilha de areia, e perturbações de um mesmo tamanho podem desencadear extinções com uma grande variedade de tamanhos. Segundo alguns pesquisadores, isto ocorre porque as espécies co-evoluem em direção a um estado crítico (Kauffman, 1993).

Bak e Sneppen (Bak, 1997) propuseram um modelo de simulação muito simples para estudar a presença de SOC na evolução. Na versão unidimensional do modelo, os indivíduos (ou espécies na terminologia dos autores) são dispostos em um círculo, sendo que um valor aleatório de *fitness* é dado a cada um deles. Em cada geração da simulação, o indivíduo com o valor de *fitness* mais baixo e os indivíduos vizinhos mais próximos a sua direita e a sua esquerda têm seus valores de *fitness* substituídos por novos valores aleatórios. Uma analogia desta conexão entre vizinhos neste modelo seria a interação entre as espécies na natureza. Se, por exemplo, uma espécie de predador tem seu *fitness* alterado, o valores de *fitness* das espécies que são suas presas irão mudar também. O modelo de evolução de Bak-Sneppen pode ser resumido pelo Algoritmo 1.

---

### Algoritmo 1

- 1: Ache o índice  $j$  do indivíduo com o menor valor de *fitness*
  - 2: Substitua o *fitness* dos indivíduos com índices  $j$ ,  $j-1$ , e  $j+1$  por valores aleatórios gerados com distribuição uniforme
- 

Este modelo simples pode levar a um comportamento interessante. No começo da simulação, o *fitness* médio da população é pequeno, mas, conforme o número de geração aumenta, o *fitness* médio aumenta também. Eventualmente, o *fitness* médio para de crescer, indicando que o

estado crítico foi alcançado. No estado crítico, os valores de *fitness* dos vizinhos do pior indivíduo são frequentemente substituídos por valores menores. Então, o pior indivíduo na geração seguinte pode ser um destes vizinhos, o qual é substituído com seus próximos vizinhos, caracterizando uma reação em cadeia, aqui chamada de extinção, que pode afetar todos os indivíduos da população. As extinções exibem invariância de escala e suas distribuições estatísticas são dadas por leis de potência nas formas apresentadas nas Eqs. 1 e 2.

É interessante notar que as grandes extinções geralmente ocorrem neste modelo quando grande parte dos indivíduos da população tem valores de *fitness* altos e similares. Desta forma, SOC evita que as espécies fiquem presas em ótimos locais presentes na superfície de *fitness* (espaço de soluções). A idéia é extremamente interessante e relativamente simples, e logo pesquisadores propuseram seu uso em processos de otimização (Boettcher and Percus, 2003), (Løvbjerg and Krink, 2002). Em AGs, Krink e Thomsen (Krink and Thomsen, 2001) propuseram o uso do modelo de pilhas de areia discutido anteriormente para gerar leis de potência utilizadas para controlar o tamanho das zonas espaciais de extinção em um modelo de difusão. Quando um indivíduo é extinto, uma versão mutada do melhor indivíduo da população é criada em seu lugar. É importante salientar que, no algoritmo proposto em (Krink and Thomsen, 2001), SOC aparece no modelo de pilhas de areia, e não como um resultado da auto-organização dos componentes do sistema (indivíduos do AG).

### 3 ALGORITMO PROPOSTO

Neste artigo, a substituição do indivíduo com o menor valor de *fitness* (pior indivíduo) da população corrente e de seus vizinhos próximos, como no modelo de evolução de Bak e Sneppen, por novos indivíduos aleatórios em AGs com Imigrantes Aleatórios é investigada. Os vizinhos do pior indivíduo são determinados através do índice de cada indivíduo. Espera-se que, agindo desta forma, o sistema possa exibir SOC, de modo que o nível de diversidade da população se comporte de uma maneira auto-organizada, permitindo que os indivíduos escapem dos ótimos locais presentes na superfície de *fitness* induzidos pelas mudanças inerentes ao problema não-estacionário.

Contudo, este procedimento simples não garante que o sistema exiba SOC, já que os novos indivíduos são geralmente substituídos por indivíduos melhores presentes na população. Propõe-se, então, a adoção de um segundo procedimento, no qual os novos indivíduos criados em uma extinção sejam confinados em uma subpopulação. O tamanho desta subpopulação não é definido pelo programador, mas sim pelo número de indivíduos criados em uma extinção. Os indivíduos que não pertencem a esta subpopulação são proibi-

dos de substituir os indivíduos que estão presentes nela.

Existem duas modificações principais em relação ao AG padrão no algoritmo investigado. Na primeira modificação, a qual é apresentada no Algoritmo 2, o tamanho corrente de cada extinção, denotado por *ext*, é computado. Os índices mínimo e máximo dos indivíduos substituídos ( $i_{min} - 1$  e  $i_{max} + 1$ ) são utilizados para calcular o número de indivíduos afetados pela extinção corrente, ou seja, presentes na subpopulação. Quando uma reação em cadeia é interrompida, isto é, o indivíduo com o valor de *fitness* mais baixo não está dentro da subpopulação ou não é vizinho dos mínimos e máximos índices dos indivíduos que a definem, o tamanho da extinção assume o valor 1.

---

#### Algoritmo 2

```

1: Ache o índice  $j$  do indivíduo com o menor fitness da
   população corrente
2: Substitua os indivíduos com índices  $j$ ,  $j - 1$ , e  $j + 1$  por in-
   divíduos aleatórios
3: if ( $j \geq i_{min} - 1$ ) e ( $j \leq i_{max} + 1$ ) then
4:    $ext \leftarrow ext + 1$ 
5:   if ( $j = i_{min} - 1$ ) then
6:      $i_{min} \leftarrow j$ 
7:   end if
8:   if ( $j = i_{max} + 1$ ) then
9:      $i_{max} \leftarrow j$ 
10:  end if
11: else
12:    $ext \leftarrow 1$ 
13:    $i_{min} \leftarrow j$ 
14:    $i_{max} \leftarrow j$ 
15: end if

```

---

A segunda modificação, a qual é apresentada no Algoritmo 3, indica o mecanismo de seleção para cada indivíduo da nova população. Dois casos podem ser considerados na seleção. No primeiro, quando o índice do indivíduo é diferente dos índices dos indivíduos presentes na subpopulação (ou seja, indivíduos com este índice não sofreram substituição por imigrantes na extinção corrente), o indivíduo é selecionado de acordo com o método de seleção escolhido pelo projetista. Neste caso, o indivíduo é escolhido a partir dos indivíduos presentes em toda a população de acordo com um dado critério, por exemplo, o método da roleta (Goldberg, 1989). Já quando o índice é igual ao índice de um indivíduo que está presente na subpopulação (ou seja, indivíduos com este índice sofreram substituição por imigrantes na extinção corrente), o indivíduo é selecionado a partir dos indivíduos presentes na subpopulação formada pelos indivíduos que foram substituídos na extinção corrente (indivíduos com índices entre  $i_{min} - 1$  e  $i_{max} + 1$ ).

---

### Algoritmo 3

**Require:** índice  $i$  do indivíduo da nova população, sendo  $1 \leq i \leq N$  e  $N$  o tamanho da população

- 1: **if** ( $i < i_{min} - 1$ ) ou ( $i > i_{max} + 1$ ) **then**
- 2: Selecione indivíduo de toda a população, ou seja, dos indivíduos com índices entre 1 e  $N$
- 3: **else**
- 4: Selecione indivíduo da subpopulação composta pelos indivíduos da população com índices entre  $i_{min} - 1$  e  $i_{max} + 1$
- 5: **end if**

---

## 4 SIMULAÇÕES

Dois conjuntos de simulações de Robôs Evolutivos em ambientes dinâmicos são apresentados nesta seção. Em tais simulações, o algoritmo proposto é comparado ao AG padrão, ao AG com Probabilidade de Mutação Dependente do Gene (GAGDM) desenvolvido para problemas não-estacionários (Tinós and Carvalho, 2006) e a duas outras versões de AG com Imigrantes Aleatórios, que diferem na estratégia de substituição dos indivíduos. Na primeira versão, que será aqui chamada de AG com Imigrantes Aleatórios 1, três indivíduos aleatoriamente escolhidos são substituídos por novos indivíduos. Na segunda versão, que será aqui chamada de AG com Imigrantes Aleatórios 2, os três piores indivíduos da população corrente, isto é, os indivíduos com os valores mais baixos de *fitness*, é que são substituídos. O GAGDM permite que as probabilidades de mutação sejam diferentes nos genes de um mesmo indivíduo. Um mecanismo de atualização dos parâmetros que controlam as probabilidades de mutação dos genes, utilizando as taxas de sucesso e fracasso produzidos pelas mutações ao longo das gerações, é empregado.

### 4.1 Descrição das Simulações

Simulações baseadas no experimento proposto em (Floreano and Mondada, 1996) foram utilizadas para testar o AG proposto. Nas simulações apresentadas aqui, uma versão modificada do simulador EVOROBOT desenvolvido por S. Nolfi (Nolfi and Floreano, 2000) é utilizada. No simulador utilizado, Redes Neurais Artificiais (RNAs) cujos pesos são ajustados por um AG são utilizadas para controlar um robô simulado cujas características são semelhantes às de um robô Khepera com 8 sensores de distância infravermelhos, 2 sensores de luz ambiente (um de cada lado, colocados sobre o robô), e 1 sensor para detectar a luminosidade do chão colocado sob o robô.

Cada indivíduo do AG é composto por um vetor contendo todos os pesos da RNA. A RNA utilizada para controlar os robôs é do tipo *feedforward*, com uma única camada oculta composta por 5 neurônios com conexões recorrentes (Rede

de Elman). A RNA possui duas saídas (saídas o0 e o1), cada uma ligada a um motor responsável pelo controle de uma das rodas do robô. As entradas da RNA são os sinais provenientes de:

- 8 sensores de distância infravermelhos colocados ao redor do robô (entradas I0 a I7). A Figura 1 mostra a disposição dos sensores de distância no robô. Com o objetivo de aumentar a robustez do sistema de controle a incertezas, os sinais provenientes dos sensores infravermelhos e de luminosidade são acrescidos de um ruído branco com limiar de  $\pm 0,05$ .
- 2 sensores de luz ambiente colocados na parte de cima do robô (entradas I8 e I9).
- 1 sensor para detectar a luminosidade do chão colocado sob o robô (entrada I10).
- 1 sensor para detectar o nível de energia da bateria (entrada I11).

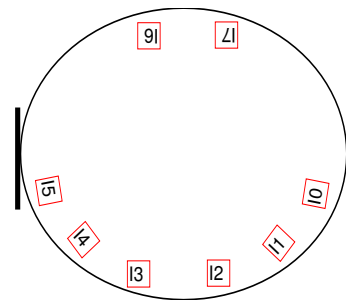


Figura 1: Disposição dos sensores de distância no robô simulado.

Nas simulações realizadas, o robô deve navegar por um ambiente desconhecido sem que ocorra choques com obstáculos. O robô possui uma quantidade limitada de energia, a qual é recarregada sempre que atravessa uma área de recarga de bateria. Esta área é caracterizada por possuir uma cor diferente no chão, o que altera as respostas do sensor colocado sob o robô, e por possuir uma torre com uma fonte de luz (ver Figura 4).

Desta forma, o *fitness* do indivíduo é definido como a velocidade média de rotação das duas rodas acumulada durante a vida do robô, i.e. enquanto sua bateria tem energia ou até que ele se choque com um obstáculo, considerando um limite máximo de 1 minuto. Como a energia da bateria totalmente carregada dura 20 segundos, o robô tem que voltar pelo menos duas vezes para a área de recarga para que a duração de sua vida seja máxima. Vale ressaltar que o *fitness* do robô não é acumulado quando ele se encontra na área de recarga

de bateria. Apesar de a função de *fitness* não especificar explicitamente que o robô deve retornar para a área de recarga de bateria quando esta estiver se exaurindo, os indivíduos que desenvolvem esta habilidade enquanto estiverem explorando o ambiente sem se chocarem com obstáculos apresentam valores maiores de *fitness*. Em cada simulação, a posição inicial do robô é fixa, sendo que sua orientação inicial varia aleatoriamente em 10 graus.

Dois conjuntos de simulações com 2000 gerações cada foram utilizados para testar o AG proposto. Os dois conjuntos de simulações foram propostos para testar o desempenho do AG proposto em tarefas em que o ambiente do robô se modifica e que envolvem tolerância a falhas.

No primeiro conjunto (simulação 1), o ambiente utilizado para a evolução dos robôs é alterado após 1000 gerações. Esta situação é frequente em problemas reais, quer devido a alterações do ambiente, quer devido às situações em que os robôs são primeiramente evoluídos em simulações, e, quando um desempenho satisfatório é alcançado, são transferidos para um ambiente real. Nas simulações aqui apresentadas, os robôs são evoluídos em uma arena de  $40 \times 45$  cm livre de obstáculos (salvo as paredes) nas primeiras 1000 gerações, e em uma arena de  $60 \times 35$  cm com quatro objetos cilíndricos com posições fixas durante as últimas 1000 gerações (ver figuras 4 e 5).

No segundo conjunto de simulações (simulação 2), o robô é afetado por uma falha de funcionamento em seis sensores de distância após 1000 gerações. Para simular esta falha, os valores dos sinais provenientes dos sensores de distância do robô correspondentes às entradas I0, I1, I2, I5, I6 e I7 são igualados a zero (ver Figura 1). A simulação 2 foi proposta para que o problema de reconfiguração após falhas seja investigado. Os robôs evoluem durante todo o tempo em uma arena de  $60 \times 35$  cm com três objetos cilíndricos fixos (ver figuras 6 e 7).

Para cada problema, os cinco AGs (AG padrão, GAGDM, AGs com Imigrantes Aleatórios 1 e 2 e AG proposto) foram simulados 20 vezes cada com diferentes sementes aleatórias. No início de cada simulação, os indivíduos foram gerados aleatoriamente. Em cada geração, os 20 melhores indivíduos da população corrente foram selecionados e cada um gerou 5 descendentes, considerando uma taxa de mutação igual a 0,01 (o operador *crossover* não foi utilizado). Para o GAGDM, os parâmetros são:  $v_{max} = 20,00$ ;  $\alpha = 0,010$  e  $\beta = 10,00$ .

## 4.2 Resultados

A Tabela 1 apresenta os resultados de adaptabilidade, que é definida como a diferença média entre o *fitness* do melhor

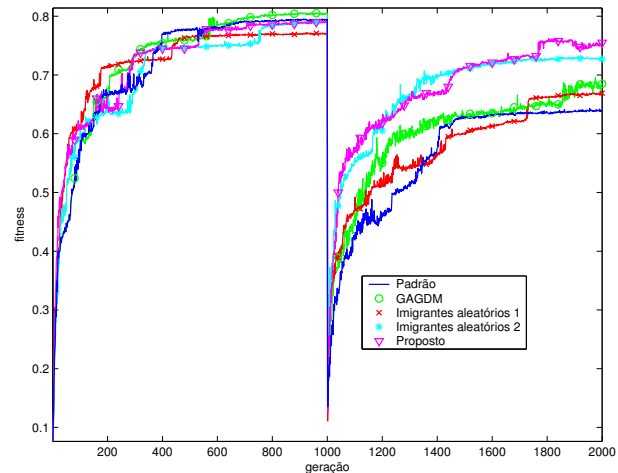


Figura 2: *Fitness* do melhor indivíduo da geração corrente para a simulação 1.

indivíduo corrente e o respectivo valor ótimo (definido aqui como 1) (Trojanowski and Michalewicz, 1999), o *fitness* médio da população, e o *fitness* do melhor indivíduo após 2000 gerações. Enquanto que os melhores resultados para a adaptabilidade são aqueles com os menores valores, os melhores resultados para o *fitness* são aqueles com os maiores valores. Os resultados apresentados na Tabela 1 correspondem à média sobre as 20 simulações de cada algoritmo para as simulações 1 e 2.

Testes de hipótese *t* indicam que o *fitness* do melhor indivíduo após 2000 gerações é maior para o algoritmo proposto, quando comparado com o AG padrão, com níveis de significância iguais a 0,04 para a simulação 1 e 0,1 para a simulação 2. As Figuras 2 e 3 mostram, respectivamente, os valores médios (sobre as 20 simulações) de *fitness* dos melhores indivíduos para as simulações 1 e 2.

As Figuras 4 e 5 mostram, respectivamente, simulações de indivíduos obtidos em uma geração anterior à mudança do ambiente e na geração final da sétima execução da simulação 1. Observe que na simulação apresentada na Figura 4, a estratégia encontrada pelo AG é a de andar em linha reta e girar para a direita sempre que encontrar uma parede, detectada através dos sensores correspondentes às entradas I6 e I7. Quando a área de recarga é encontrada, que é detectada através do sensor de luminosidade correspondente à entrada I10 e cuja aproximação e afastamento são detectados respectivamente através dos sensores de luz ambiente (entradas I8 e I9), o robô muda de direção. É interessante observar que o robô somente volta para a área de recarga quando o nível de energia da bateria, que é medido pelo sensor correspondente à entrada I11, está quase em zero. Esta estratégia, que permite que o *fitness* acumulado seja alto já que o robô navega

Tabela 1: Resultados

AG		Simulação 1 (mudança do ambiente)	Simulação 2 (reconfiguração após falhas)
Adaptabilidade	Padrão	0,362 (36,14 %)	0,600 (60,02%)
	GAGDM	0,339 (33,88 %)	0,574 (57,41 %)
	Imigrantes Aleatórios 1	0,351 (35,08 %)	0,520 (52,01 %)
	Imigrantes Aleatórios 2	0,313 (31,29 %)	0,627 (62,68 %)
	Proposto	0,302 (30,22 %)	0,519 (51,91 %)
<i>Fitness</i> Médio	Padrão	0,299	0,186
	GAGDM	0,300	0,183
	Imigrantes Aleatórios 1	0,341	0,232
	Imigrantes Aleatórios 2	0,354	0,184
	Proposto	0,348	0,230
<i>Fitness</i> Final do Melhor Indivíduo	Padrão	0,638	0,284
	GAGDM	0,685	0,425
	Imigrantes Aleatórios 1	0,669	0,388
	Imigrantes Aleatórios 2	0,727	0,359
	Proposto	0,755	0,441

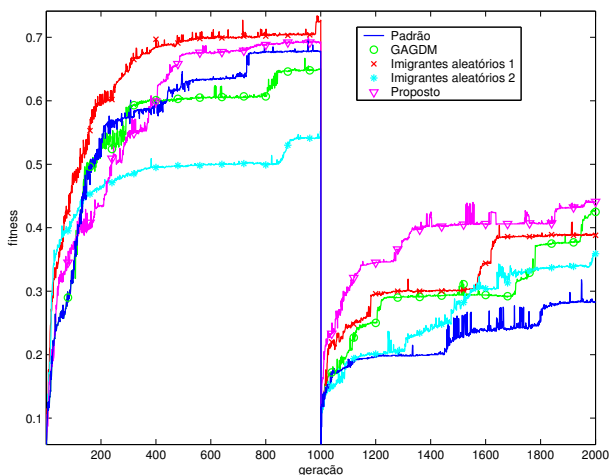


Figura 3: *Fitness* do melhor indivíduo da geração corrente para a simulação 2.

o maior tempo possível fora da área de recarga, foi encontrada pelo AG sem que o projetista a especificasse.

Pode-se observar que a estratégia adotada na simulação apresentada na Figura 5 é diferente. Nesta estratégia, o robô se movimenta em uma área menor, já que obstáculos estão presentes no ambiente. Observe que após a recarga da bateria, o robô faz um círculo completo, uma curva para a direita e caminha até encontrar a parede, detectada através dos sensores correspondentes às entradas I6 e I7. É interessante notar que o círculo é feito para que o robô possa navegar o maior tempo possível fora da área de recarga.

Já nas simulações apresentadas nas Figuras 6 e 7 para a

décima execução da simulação 2, o robô praticamente só anda em linha reta. A simulação apresentada na Figura 6 utiliza uma estratégia de controle obtida pelo AG proposto em uma geração anterior à introdução da falha. Observe que, ao encontrar uma parede, o robô faz pequenos movimentos em que gira para a direita sob seu eixo, até que fique com suas rodas paralelas à parede. Isto pode ser observado pelas ativações dos sensores correspondentes às entradas de I1 à I4. Então, o robô caminha de forma quase paralela à parede, utilizando o sensor lateral correspondente à entrada I0, até que a área de recarga ou outra parede seja encontrada. Uma vez encontrada a área de recarga, o robô gira para a direita e anda em linha reta até a parede oposta. Novamente observe que a bateria somente é recarregada quando o seu nível está quase em zero.

Uma estratégia diferente é adotada na simulação apresentada na Figura 7, que utiliza um indivíduo obtido pelo AG proposto na geração final da sétima execução da simulação 2. Nesta simulação, o robô apresenta uma falha que o faz perder os sinais de todos os sensores de distância, salvo os sensores correspondentes às entradas I3 e I4. Pode-se observar nos gráficos apresentados, que somente estes dois sensores apresentam ativação diferente de zero. Na estratégia encontrada pelo AG proposto, o robô caminha em direção à área de recarga utilizando um dos sensores de luz ambiente (entrada I8). Antes de chegar na área de recarga, o robô desvia de um obstáculo utilizando os sensores de distância que ainda funcionam (entradas I3 e I4). Ao chegar na área de recarga, o robô passa a fazer pequenos movimentos lineares para frente e para trás, saindo e entrando na área de recarga (observe a entrada I10 correspondente ao sensor de luminosidade colocado sob o robô). Desta forma, sua bateria fica em um nível

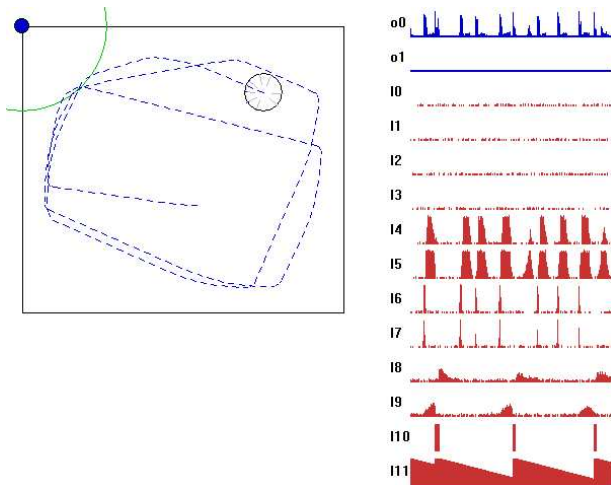


Figura 4: Simulação de um indivíduo obtido na geração anterior à mudança no ambiente na sétima execução da simulação 1 utilizando o algoritmo proposto. O robô é representado através de um círculo de cor branca e sua trajetória através de uma linha tracejada. A área de recarga de bateria, representada por um semi-círculo no canto superior da arena, possui uma cor diferente no chão. Uma torre de iluminação, representada por um pequeno círculo hachurado, se encontra no canto superior da arena. Nos gráficos à direita, os sinais de saída para os motores (o0 e o1) e as entradas da rede provenientes dos sensores (I0 a I11) são representados nos eixos verticais e o tempo no eixo horizontal. Esta figura foi obtida através do simulador EVOROBOT.

quase estável (entrada I11). Apesar de não acumular valores altos de *fitness* já que robô não navega pelo ambiente, esta estratégia faz com que o robô não se choque com os obstáculos do ambiente já que possui pouca capacidade sensorial.

### 4.3 Análise dos Resultados

Nos conjuntos de simulações apresentados na Seção 4.2, os AGs encontram, ao longo das gerações, pesos da RNA que definem a estratégia de navegação adotada. Para isso, mapas topológicos do ambiente são criados internamente e são utilizados para a navegação no ambiente de modo a aumentar ao máximo o *fitness* acumulado durante a vida do robô. A criação destes mapas topológicos internos nestas simulações é semelhante àquela encontrada nos resultados experimentais apresentados em (Nolfi and Floreano, 2000).

Quando mudanças induzidas nos problemas são drásticas, como aquelas apresentadas na seção anterior, novas estratégias muito diferentes das antigas têm que ser encontradas. Desta forma, o robô não pode mais confiar nas representações internas do ambiente previamente encontradas. Na simulação 1, o ambiente mudou de tal forma, que

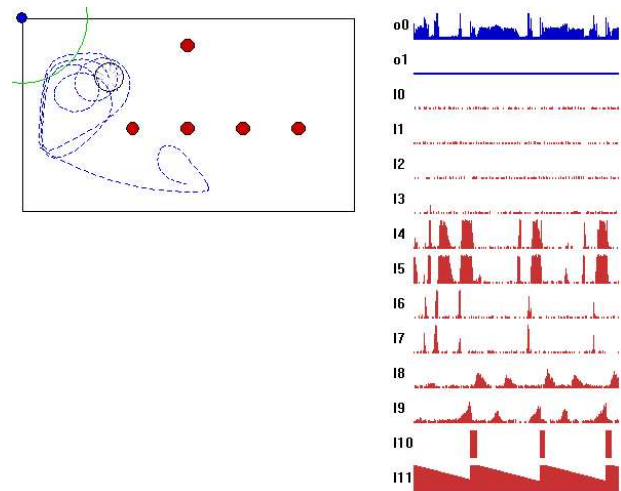


Figura 5: Simulação de um indivíduo obtido na geração final na sétima execução da simulação 1 utilizando o algoritmo proposto. A arena simulada tem tamanho diferente daquela apresentada na Figura 4, além de possuir quatro obstáculos cilíndricos, representados na figura por círculos hachurados.

o robô teve que abandonar a estratégia previamente encontrada e buscar uma outra (veja Figuras 4 e 5). O mesmo vale para a simulação 2, já que as falhas introduzidas mudam a forma que o robô enxerga o ambiente (veja Figuras 6 e 7). Desta forma, o robô precisou desenvolver estratégias que não utilizassem os sinais provenientes aos sensores com falha. As alterações de estratégia provocadas pelas mudanças no problema foram produzidas por grandes alterações na representação da RNA responsável pelo controle do robô.

No AG padrão, e mesmo no GAGDM, uma nova estratégia para o problema é mais difícil de ser encontrada após a mudança no problema, já que o algoritmo muitas vezes fica preso em ótimos locais dados pela solução antiga. Este problema pode ser evitado quando novas soluções aleatórias são geradas, como nos AGs com Imigrantes Aleatórios. Os resultados dos AGs com Imigrantes Aleatórios e do AG proposto serão agora analisados. Primeiro, o modo com o AG proposto funciona será comentado.

Nas primeiras gerações das execuções do AG proposto, os indivíduos geralmente apresentam valores de *fitness* baixos e um nível de diversidade alto. No AG proposto, os novos indivíduos que substituem o pior indivíduo e seus dois vizinhos próximos também têm valores de *fitness* baixos. Como diversos indivíduos da população têm valores baixos de *fitness*, a probabilidade que o novo pior indivíduo seja um dos vizinhos do pior indivíduo anterior é baixa. Consequentemente, substituições de um único indivíduo dificilmente geram grandes reações em cadeia.



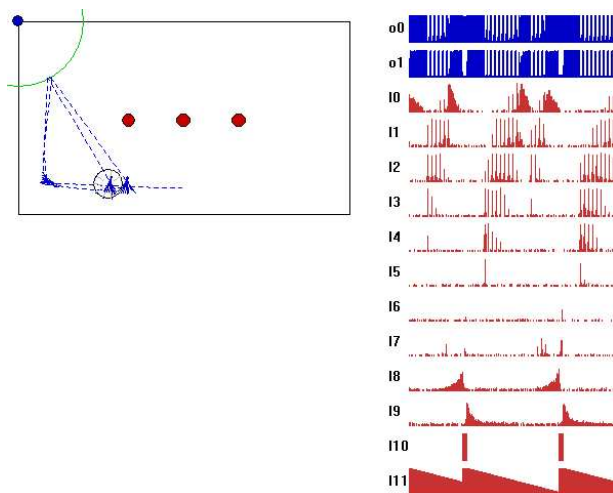


Figura 6: Simulação de um indivíduo sem falhas obtido na geração anterior à mudança no problema na décima execução da simulação 2 utilizando o algoritmo proposto.

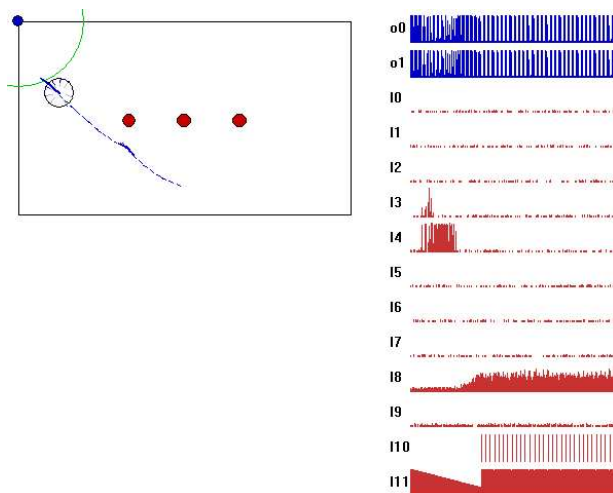


Figura 7: Simulação de um indivíduo com falhas obtido na geração final na décima execução da simulação 2 utilizando o algoritmo proposto.

Conforme o *fitness* médio da população aumenta, eventualmente um ótimo local é alcançado. Então, conforme aumenta o número de gerações, grande parte da população converge para o ótimo local, resultando em um baixo nível de diversidade. Nesta situação, diversos indivíduos da população têm valores altos de *fitness* se comparados com o *fitness* médio de todos os novos indivíduos possíveis. Quando o pior indivíduo da população é substituído com seus dois vizinhos próximos, a probabilidade que o pior indivíduo da população seguinte seja um destes vizinhos é alta já que eles geralmente apresentavam valores de *fitness* altos. Então, uma grande reação em cadeia (extinção) pode ser formada e um grande número de indivíduos pode ser extinto, aumentando assim o nível de diversidade da população. De fato, o fator de correlação entre o tamanho das extinções e a variância do *fitness* da população é positiva, indicando que a diversidade cresce com o tamanho das extinções. É interessante observar que este é um comportamento auto-organizável, que muda com o nível de diversidade da população.

Os melhores resultados do algoritmo proposto quando comparado com os outros dois AGs com Imigrantes Aleatórios nas simulações apresentadas podem ser explicados por dois fatores principais. Primeiro, o número de índices diferentes de indivíduos que são substituídos em um período fixo de gerações geralmente é mais alto no algoritmo proposto. No AG com Imigrantes Aleatórios em que os piores indivíduos são substituídos, é comum a substituição de indivíduos que acabaram de aparecer na população, ou seja, com os mesmos índices. Isto é explicado pelos valores baixos de *fitness* dos novos indivíduos, gerando baixos níveis de diversidade. Já no AG proposto, as reações em cadeia geram um número

maior de índices de indivíduos que são substituídos e mantidos em uma subpopulação. Este fator pode ser observado na análise do *fitness* médio da população na Tabela 1. Observe que o algoritmo proposto apresenta valores de *fitness* médio da população próximos aos dos outros AGs com Imigrantes Aleatórios, mesmo apresentando os maiores valores de *fitness* do melhor indivíduo.

O segundo fator principal que explica os melhores resultados para o AG proposto é que a probabilidade de que um novo indivíduo seja preservado geralmente é maior no algoritmo proposto. Isto ocorre porque, em um AG com Imigrantes Aleatórios, os valores de *fitness* dos indivíduos da população corrente, que geralmente estão localizados próximos do ótimo local após algumas gerações, são em geral maiores que o *fitness* médio do espaço de busca, isto é, de todos os novos indivíduos possíveis. Um imigrante geralmente sobrevive somente se seu *fitness* for próximo de ou maior que o *fitness* médio da população, o que é um evento raro quando a população está em um ótimo local muito maior que o *fitness* médio do espaço de busca. Nas simulações apresentadas, pode-se observar que a solução antiga é geralmente muito melhor que a maioria das novas soluções geradas, já que na solução antiga, o robô geralmente conseguia navegar em linha reta pelo ambiente. Por outro lado, o AG proposto preserva uma nova solução potencial em uma subpopulação e permite que esta solução evolua enquanto a extinção corrente está em progresso. Quando a extinção termina, versões evoluídas de soluções potenciais dadas por imigrantes podem ser dispersas na população atual.

A Figura 8 apresenta o *fitness* do melhor indivíduo e

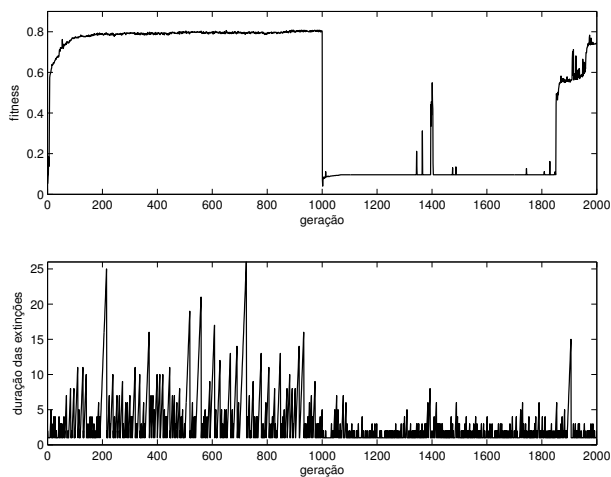


Figura 8: *Fitness* do melhor indivíduo e duração das extinções na décima execução da simulação 2 (reconfiguração após falhas).

a duração das extinções para o AG proposto na décima execução da simulação 2. Observe que, após a introdução da falha (geração 1000), o *fitness* do melhor indivíduo diminui, já que o robô não consegue navegar pelo ambiente com a estratégia previamente evoluída. No entanto, após algumas extinções, uma nova estratégia é encontrada (depois da geração 1800). Observe, ainda, que extinções com durações médias altas geralmente ocorrem quando o valor de *fitness* é alto.

Nas simulações apresentadas nesta seção, assim como nos registros fósseis sobre tamanhos de extinções na natureza (Raup, 1986), existem mais extinções pequenas que grandes e o tamanho das extinções exibe uma grande variedade de escalas. Na Figura 9, o gráfico em uma escala log-log do número de extinções contra o tamanho de cada uma na décima execução da simulação 2 é apresentado. Observe que o gráfico exibe lei de potência, mesmo sem nenhuma ação de controle externo, indicando a presença de SOC. Este tipo de comportamento auto-organizável aparece em sistemas em que muitos graus de liberdade interagem e cuja dinâmica é dominada pela interação entre estes graus de liberdade, e não pela dinâmica intrínseca produzida individualmente por eles (Jensen, 1998).

## 5 CONCLUSÕES

Neste trabalho, a substituição do pior indivíduo e de seus dois vizinhos próximos em cada geração é utilizada para os Algoritmos Genéticos com Imigrantes Aleatórios aplicados a problemas não-estacionários associados a robôs móveis. Os novos imigrantes são preservados em uma subpopulação, cujo tamanho é definido não pelo programador e sim pelo

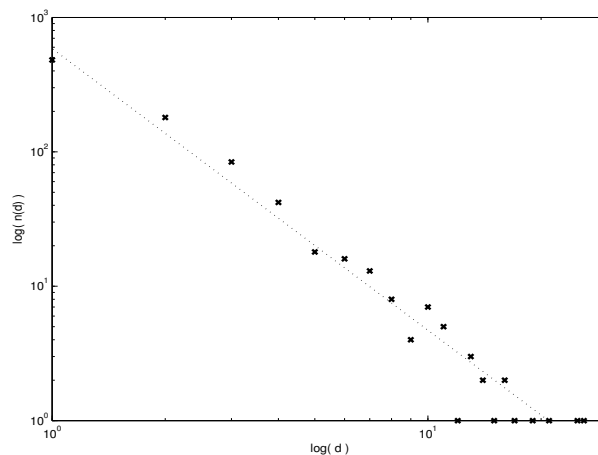


Figura 9: Número de ocorrências para cada tamanho das extinções na décima execução da simulação 2 (reconfiguração após falhas).

tamanho da extinção corrente. No algoritmo proposto, a população se auto-organiza de modo a permitir a ocorrência de extinções com uma grande variedade de escalas de tamanhos. Grandes extinções geralmente ocorrem quando a maior parte dos indivíduos da população exibe valores altos de *fitness*, e quando o nível de diversidade da população é pequeno. Desta forma, a diversidade da população é aumentada nestes casos, o que faz com que o AG possa escapar de ótimos locais induzidos por mudanças no problema. Assim, o AG proposto é interessante em problemas em que a nova solução a ser encontrada está localizada em pontos que são dificilmente alcançados, pelos operadores tradicionais, a partir do ótimo local que representa a solução antiga.

O algoritmo proposto é interessante quando ocorrem mudanças bruscas no robô ou no ambiente. Nas simulações apresentadas, dois tipos de mudanças foram introduzidos. O primeiro tipo de mudança consistiu em uma alteração no ambiente em que o robô se encontra. O segundo tipo consistiu na introdução de falhas que afetaram grande parte dos sensores de distância do robô. Para estes tipos de alterações, o algoritmo proposto apresentou resultados interessantes, permitindo que novas estratégias de navegação surgissem através do aumento da diversidade na população. Ressalta-se, no entanto, que o algoritmo proposto não é interessante quando não se necessitam de estratégias muito diferentes das que estão sendo empregadas pelo robô, por exemplo, quando as alterações no ambiente são pequenas. Isto ocorre porque os indivíduos gerados aleatoriamente são geralmente muito piores que a solução atualmente encontrada.

No futuro, novas estratégias de escolha dos vizinhos do pior indivíduo da população serão estudadas. Além disso, o algoritmo proposto deverá ser aplicado em robôs móveis reais.

## AGRADECIMENTOS

O autor agradece à FAPESP pelo apoio financeiro através do Processo JP 04/04289-6 e ao Prof. Shengxiang Yang da University of Leicester, Reino Unido, pelos comentários à respeito dos algoritmos.

## REFERÊNCIAS

- Bak, P. (1997). *How nature works: the science of self-organized criticality*, Oxford University Press.
- Billard, A., Ijspeert, A. J. and Martinoli, A. (2000). A multi-robot system for adaptive exploration of a fast changing environment: probabilistic modeling and experimental study, *Connection Science* **11**(3/4): 357–377.
- Boettcher, S. and Percus, A. G. (2003). Optimization with extremal dynamics, *Complexity* **8**(2): 57–62.
- Branke, J. (2003). Evolutionary approaches to dynamic optimization problems - introduction and recent trends, in J. Branke (ed.), *GECCO Workshop on Evol. Alg. for Dynamic Optimization Problems*, pp. 2–4.
- Cobb, H. G. and Grefenstette, J. J. (1993). Genetic algorithms for tracking changing environments, in S. Forrest (ed.), *Proc. of the 5th Int. Conf. on Genetic Algorithms*, pp. 523–530.
- Floreano, D. and Mondada, F. (1996). Evolution of homing navigation in a real mobile robot, *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics* **26**(3): 396–407.
- Floreano, D., Nolfi, S. and Mondada, F. (2001). Co-evolution and ontogenetic change in competing robots, in M. Patel, V. Honavar and K. Balackishnan (eds), *Advances in the Evolutionary Synthesis of Intelligent Agents*, MIT Press.
- Floreano, D. and Urzelai, J. (2000). Evolutionary robots with on-line self-organization and behavioral fitness, *Neural Networks* **13**: 431–443.
- Goldberg, D. A. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc.
- Jakobi, N. (1997). Half-baked, ad-hoc and noisy: minimal simulations for evolutionary robotics, in P. Husbands and I. Harvey (eds), *Proc. of the Fourth European Conference on Artificial Life*, MIT Press, pp. 348–357.
- Jensen, H. J. (1998). *Self-organized criticality: emergent complex behavior in physical and biological systems*, Cambridge University Press.
- Kauffman, S. A. (1993). *The origins of order: self-organization and selection in evolution*, Oxford University Press.
- Krink, T. and Thomsen, R. (2001). Self-organized criticality and mass extinction in evolutionary algorithms, *Proc. of the 2001 IEEE Congress on Evolutionary Computation*, Vol. 2, pp. 1155–1161.
- Løvbjerg, M. and Krink, T. (2002). Extending particle swarm optimisers with self-organized criticality, *Proc. of the 2002 IEEE Congress on Evolutionary Computation*, Vol. 2, pp. 1588–1593.
- Marley, G. (1996). Landscapes, learning costs and genetic assimilation, *Evolutionary Computation* **4**(3): 213–234.
- Meyer, J. A. (1998). Evolutionary approaches to neural control in mobile robots, *Proc. of the IEEE Conf. on Syst., Man, and Cyb.*
- Nolfi, S. and Floreano, D. (2000). *Evolutionary robotics: the biology, intelligence, and technology of self-organizing machines*, MIT Press/Bradford Books: Cambridge, USA.
- Raup, D. M. (1986). Biological extinction in earth history, *Science* **231**: 1528–1533.
- Terra, M. H. and Tinós, R. (2001). Fault detection and isolation in robotic manipulators via neural networks - a comparison among three architectures for residual analysis, *Journal of Robotic Syst.* **18**(7): 357–374.
- Thompson, A. (1998). On the automatic design of robust electronics through artificial evolution, in M. Sipper and A. Prez-Urbe (eds), *Proc. of the 2nd Int. Conf. on Evolvable Systems: From Biology to Hardware*, Springer, pp. 13–24.
- Tinós, R. and Carvalho, A. C. P. L. F. (2006). Use of gene dependent mutation probability in evolutionary neural networks for non-stationary problems, *Neurocomputing* **70**(1-3): 44–54.
- Tinós, R., Navarro-Serment, L. E. and C. J. J. Paredis, C. J. J. (2001). Fault tolerant localization for teams of distributed robots, *Proc. of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1061–1066.
- Trojanowski, K. and Michalewicz, Z. (1999). Evolutionary algorithms for non-stationary environments, in M. A. Kłopotek and M. Michalewicz (eds), *Intelligent Inf. Systems, Proc. of the 8th Int. Workshop on Intelligent Inf. Syst.*, pp. 229–240.