

---

# ALGORITMOS EVOLUTIVOS EM IDENTIFICAÇÃO E CONTROLE DE PROCESSOS: UMA VISÃO INTEGRADA E PERSPECTIVAS

Leandro dos Santos Coelho

Antonio Augusto Rodrigues Coelho

Departamento de Automação e Sistemas – Universidade Federal de Santa Catarina  
C.P. 476 - CEP 88040.900 - Florianópolis - SC - Brasil  
e-mail: {lscoelho, aarc}@lcmi.ufsc.br

---

**Resumo:** Este artigo apresenta um panorama da aplicabilidade do paradigma computacional computação evolutiva em identificação e controle de processos industriais. Os algoritmos evolutivos descritos e analisados são: (i) algoritmos genéticos, (ii) programação evolutiva, (iii) estratégias evolutivas e, (iv) programação genética. O desenvolvimento de ferramentas de *software*, projeto em *hardware* e a interação com outras metodologias da inteligência computacional, tais como sistemas nebulosos e redes neurais artificiais são abordadas. As pesquisas no meio acadêmico e industrial são tratadas dentro de um espectro de avanços e perspectivas na área de automação industrial. Exemplos práticos da aplicabilidade de alguns algoritmos evolutivos são realizados na identificação experimental de um processo térmico com a estimação do atraso de transporte, e na sintonia de um controlador proporcional-integral-derivativo (*PID*) para o controle de um sistema não-linear de regulação de nível.

**Palavras-Chave:** Identificação de processos; modelagem; controle inteligente; algoritmos genéticos; programação evolutiva; estratégias evolutivas; programação genética; computação evolutiva; otimização.

**Abstract:** This paper presents an overview of evolutionary computation applications in identification and control of industrial processes. The evolutionary algorithms described and analysed here are (i) genetic algorithms, (ii) evolutionary programming, (iii) evolution strategies, and (iv) genetic programming. The development of software tools, hardware design and interaction with others computational intelligence methodologies, such as fuzzy systems and artificial neural networks are described. The researches carried out in academic and industrial environments are approached from the perspectives and evolution spectrum of the industrial automation area. Practical examples of the evolution algorithms are applied to experimental identification of a temperature process with time delay estimation, and to the

Artigo submetido em 21/11/96

1a. Revisão em 14/03/97; 2a. Revisão em 03/09/97;

3a. Revisão em 21/01/98; 4a. Revisão em 21/09/98;

Aceito sob recomendação do Ed. Cons Prof.Dr. Ricardo Tanscheit e do Ed. Cons. Prof.Dr. Fernando A.C.Gomide

tuning of a proportional-integral-derivative (*PID*) controller for regulating the liquid-level in a nonlinear system.

**Keywords:** Process identification; modeling; intelligent control; genetic algorithms; evolutionary programming; evolution strategies; genetic programming; evolutionary computation; optimization.

## 1 INTRODUÇÃO

A necessidade de projetistas e engenheiros de lidarem com sistemas cada vez mais complexos, tem viabilizado a identificação e controle de sistemas difíceis de serem modelados matematicamente (não-lineares e com atraso de transporte variante no tempo) pela utilização da inteligência computacional (*IC*). A utilização de *IC* combinada a teoria convencional de controle é motivada pela adequação ao tratamento de restrições e satisfação de requisitos de robustez e desempenho em projetos na área de automação industrial.

A progressiva implementação destas técnicas, principalmente a partir dos anos 70, vem de encontro também à crescente necessidade de melhorar a eficiência das plantas industriais pelo aumento da qualidade dos produtos, diminuição dos impactos prejudiciais ao meio ambiente, diminuição de perdas, além de outros fatores vinculados ao aprimoramento e tratamento de deficiências de métodos clássicos em sistemas de controle.

O desenvolvimento da *IC*, por pesquisadores das mais diversas áreas do conhecimento, tem levado a promissores resultados quanto à abrangente aplicabilidade das técnicas inteligentes não apenas no meio científico. As aplicações industriais, militares e produtos comerciais que empregam com sucesso técnicas, tais como: sistemas nebulosos (*fuzzy*), redes neurais artificiais, redes neurais nebulosas, computação evolutiva, *simulated annealing*, sistemas especialistas, sistemas híbridos inteligentes e teoria do caos, têm motivado crescentes investimentos em pesquisa na área, principalmente por órgãos governamentais e privados dos Estados Unidos, Japão e Europa

(EvoNews, 1996; Katayama *et alii*, 1993; Linkens & Nyongesa, 1996a).

A sabedoria da natureza manifestada através dos princípios de evolução natural, genética e seleção natural tem inspirado e provado ser um mecanismo poderoso para o aparecimento e aperfeiçoamento de seres vivos no nosso planeta por meio da execução de procuras aleatórias no espaço de possíveis seqüências de *DNA* (*deoxyribonucleic acid*). O conhecimento corrente sobre as características da evolução natural biológica levou alguns pesquisadores a tentarem utilizar os procedimentos básicos da evolução natural como embasamento à implementação de algoritmos computacionais de busca estocástica e métodos de otimização.

O paradigma computacional da computação evolutiva ou evolucionária (*CE*) imita um modelo rudimentar e simplificado do modelo da natureza como um processo adaptativo de busca e otimização que possibilite implementações computacionais. A *CE* sugere um mecanismo em que uma população de indivíduos (soluções) visa melhorar, em média, a sua adequação em relação ao ambiente, ou seja, o seu desempenho geral com respeito a um dado problema (Goldberg, 1989; Srinivas & Patnaik, 1994).

O presente artigo apresenta o estado da arte e as perspectivas de aplicação da *CE* no contexto de automação industrial. A seção 2 tem caráter introdutório e aborda os fundamentos das seguintes metodologias de *CE*: (i) algoritmos genéticos, (ii) programação evolutiva, (iii) estratégias evolutivas e, (iv) programação genética. A seção 3 abrange e descreve algumas pesquisas, aplicações em identificação e controle de processos, além de ferramentas em *software*, *hardware* e a utilização de algoritmos evolutivos em sistemas híbridos com metodologias da *IC*.

Os resultados práticos da aplicabilidade de *CE* na identificação experimental de um processo térmico com a estimação do atraso de transporte e na sintonia de um controlador *PID* em controle de um sistema não-linear de regulação de nível são descritos e analisados na seção 4. A seção 5 é dedicada a conclusão e um breve panorama de perspectivas e contribuições das metodologias de *CE* no contexto de controle de processos.

## 2 COMPUTAÇÃO EVOLUTIVA

Os métodos de busca e otimização, usualmente, são classificados em técnicas baseadas em cálculo, procura aleatória e enumerativas. Os métodos guiados por procura aleatória são baseados em técnicas enumerativas, contudo utilizam-se de informação adicional para guiar a procura, e suas maiores classes são os paradigmas de *simulated annealing* e da *CE* (Filho *et alii*, 1994).

Atualmente, a *CE* constitui-se numa alternativa às técnicas convencionais em busca e otimização. A *CE* engloba um número crescente de metodologias, das quais as mais relevantes são:

algoritmos genéticos, desenvolvidos principalmente por A.S. Fraser, H.J. Bremermann, J. Reed e J.H. Holland, entre a década de 50 e 70, com refinamentos posteriores por D. Whitley, D.E. Goldberg, K. De Jong e J. Grefenstette;

- programação evolutiva, desenvolvidas por L.J. Fogel, A.J. Owens e M.J. Walsh, nos Estados Unidos, na década de

60, refinada recentemente por D.B. Fogel, G.H. Burgin, P.J. Angeline, V.W. Porto e W. Atmar;

- estratégias evolutivas, desenvolvidas na Alemanha, por I. Rechenberg e H.P. Schwefel, na década de 60, com posteriores aprimoramentos de G. Rudolph, H.-G. Beyer, F. Kursawe, T. Bäck;
- programação genética, tratadas pelos pesquisadores J.R. Koza, J.P. Rice, K.E. Kinneer e P.J. Angeline;
- sistemas classificadores, abordados na literatura por D.E. Goldberg, J.H. Holland, S.F. Smith e S.W. Wilson.

O paradigma *CE* atualmente tende a interagir de modo a dar origem aos denominados algoritmos evolutivos ou evolucionários (*AEs*). A seguir apresenta-se uma breve base teórica dos paradigmas da *CE* e os passos necessários à sua execução representados por pseudocódigos. Mais detalhes podem ser encontrados na seguinte literatura: Bäck & Schwefel (1993), Bäck *et alii* (1997a), Davis (1991), Fogel (1995), Goldberg (1989), Holland (1975), Michalewicz (1992), Tanomaru (1995), entre outras referências.

### 2.1 Algoritmos Genéticos

Os algoritmos genéticos (*AGs*) foram desenvolvidos por J.H. Holland (1975) na *University of Michigan*. Os *AGs* são técnicas não-determinísticas de busca, otimização e aprendizagem de máquina, que manipulam um espaço de soluções potenciais utilizando mecanismos inspirados nas teorias de seleção natural de C. Darwin e na genética de G. Mendel. Os *AGs* são robustos e eficientes em espaços de procura irregulares, multidimensionais e complexos, e caracterizam-se por (Goldberg, 1994):

- operarem em uma população de pontos;
- não requererem derivadas;
- trabalharem com a codificação de seu conjunto de parâmetros, não com os próprios parâmetros (representação binária);
- realizarem transições probabilísticas, não regras determinísticas;
- necessitarem apenas de informação sobre o valor de uma função objetivo para cada integrante da população de indivíduos.

Os tradicionais *AGs*, denominados *AGs* canônicos, baseiam-se em noções do *schema theorem* e *building blocks*, e os indivíduos são representados por vetores binários (Holland, 1975), contudo esta representação não é universalmente aceita na literatura.

Alguns pesquisadores indicam que a representação real (ponto flutuante) apresenta melhor desempenho em relação a representação binária para aplicações que necessitem de tratamento de valores contínuos, pois apresenta mais compatibilidade, precisão e rapidez de execução. A representação binária é mais apropriada em aplicações que requeiram o tratamento de valores discretos (Davis, 1991; Janikow & Michalewicz, 1990; Michalewicz, 1992).

Contudo, o procedimento básico de otimização por AGs, seja este com representação canônica ou real, utiliza-se de três operadores básicos: seleção, recombinação e mutação. Os AGs são usualmente implementados conforme os seguintes passos (Bäck *et alii*, 1997a; Fogel, 1995; Michalewicz, 1992):

- (i) a população inicial de parâmetros compreende  $P$  soluções. Cada uma das soluções consiste de vetores  $x_i \in \{0,1\}$  (representação canônica) ou  $x_i \in \mathfrak{R}^n$  (representação real). Estes parâmetros são inicializados de acordo com uma distribuição uniforme;
- (ii) classificar cada solução  $x_i$ ,  $i=[1,P]$ , com relação ao cálculo da função de adequação (*fitness*), ou seja, avalia-se o grau de adaptação de cada indivíduo da população em relação ao problema;
- (iii) selecionar os indivíduos mais aptos de acordo com a estratégia de seleção;
- (iv) aplicar os operadores genéticos de recombinação e mutação;
- (v) gerar uma nova população; e
- (vi) repetir os passos (ii) a (v) até que uma condição de parada seja satisfeita.

O procedimento mencionado para os AGs, nas etapas (i) a (vi), podem ser sintetizados pelo pseudocódigo de seu ciclo evolutivo, conforme apresentado na Figura 1 (Bäck & Schwefel, 1993).

```

t := 0;
inicializar: P(0) := {a1(0), a2(0), ..., aα(0)}
avaliar: P(0) := {Φ(a1(0)), Φ(a2(0)), ..., Φ(aα(0))}
enquanto uma condição de parada não é satisfeita
{
  realizar mutação: P'(t) := mΘm(P(t))
  avaliar: P'(t) := {Φ(a'1(t)), Φ(a'2(t)), ..., Φ(a'α(t))}
  selecionar: P(t+1) := sΘs(P'(t))
  t := t + 1;
}

```

**Figura 1. Pseudocódigo básico dos AGs.**

As convenções utilizadas na Figura 1, são as seguintes:

- $a$ : indivíduo da população antiga;
- $a'$ : indivíduo da população atual;
- $\alpha$ : número de indivíduos (população);

$P(t) := \{a_1(t), a_2(t), \dots, a_\alpha(t)\}$ : população antiga no tempo (geração)  $t$ ;

$P'(t) := \{a'_1(t), a'_2(t), \dots, a'_\alpha(t)\}$ : população atual no tempo (geração)  $t$ ;

$\Phi: I \rightarrow \mathfrak{R}$ : mapeamento do função de adequação;

$m_{\Theta_m}$ : operador de mutação com parâmetros de controle  $\Theta_m$ ;

$s_{\Theta_s}$ : operador de seleção  $\ni s_{\Theta_s}: I^\alpha \rightarrow I^\alpha$ ;

O operador de seleção emprega o princípio de sobrevivência dos indivíduos mais aptos, por meio de uma metáfora aos processos de reprodução assexuada e seleção natural, de acordo com o grau de adaptação do indivíduo. Os indivíduos são selecionados de acordo com uma probabilidade dada pela sua função de adequação.

Um método de seleção muito utilizado é o método da roleta, onde cada indivíduo ocupa, em uma roleta, uma área proporcional a sua função de adequação. Assim, aos indivíduos com maior aptidão é associada uma fatia maior da roleta e vice-versa. Outros exemplos de formas de seleção são: ranqueamento, *breeder*, *tournament*, *steady-state*, Boltzmann, entre outras. Mais detalhes das diversas formas de configuração dos operadores de seleção podem ser encontrados em Bäck *et alii* (1997a), Goldberg (1989) e Tanomaru (1995).

O operador genético de recombinação (*crossover*) é responsável pela troca de material genético entre os indivíduos com probabilidade de reproduzirem mais frequentemente “bons” indivíduos, ou seja, mais aptos ao ambiente. O operador de mutação implica na modificação do valor de cada gene de um indivíduo e visa restaurar o material genético perdido ou não explorado em uma população visando prevenir a convergência prematura do AG para soluções sub-ótimas.

A definição dos parâmetros intrínsecos aos AGs geralmente são determinados heurísticamente, tais como: tamanho da população, tamanho da estrutura dos indivíduos, probabilidade de utilização dos operadores genéticos.

Apesar de intensas pesquisas na área, principalmente com o emprego de métodos de adaptação dinâmica dos parâmetros de controle dos AGs, não existe uma regra determinística para o projetista estipular o tamanho da população e probabilidade dos operadores genéticos de modo a obter-se uma adequada relação quanto aos tópicos de diversidade na população e a capacidade de convergência dos AGs.

A literatura menciona que as configurações adotadas em AGs (representação binária) utilizam usualmente o tamanho da população entre 30 e 200, probabilidade de recombinação entre 0.5 e 1.0 e probabilidade de mutação entre 0.001 e 0.05 (Davis, 1991; Srinivas & Patnaik, 1994).

## 2.2 Programação Evolutiva

A programação evolutiva (PE) é uma classe da CE que utiliza os conceitos de evolução para gerar progressivamente soluções apropriadas em ambientes estáticos ou mesmo que mudam dinamicamente.

A PE, bem como as estratégias evolutivas, descritas a seguir no item 2.3, diferem dos AGs, pois são técnicas que simulam a evolução enfatizando a ligação comportamental (relação fenotípica) entre as populações

geradas (ancestrais e descendentes), mais que a ligação genética.

A *PE* desenvolvida por L.J. Fogel (1962, 1964) na *National Science Foundation*, teve seu enfoque inicial na evolução de máquinas de estado finito. A transformação de seqüências de símbolos de entrada em seqüências de símbolos de saída pelas máquinas de estado finito visava tratar a predição de séries temporais baseando-se em anteriormente informações disponíveis. A *PE*, posteriormente, foi estendida para problemas de otimização de parâmetros.

O procedimento de otimização pela *PE* é usualmente implementado conforme os seguintes passos (Fogel, 1994, 1995):

(i) a população inicial de parâmetros compreende  $P$  soluções. Cada uma dos indivíduos  $(x_i, \sigma_{xi})$  consiste de vetores de soluções,  $x_i \in \mathfrak{R}$ , e desvio padrão,  $\sigma_{xi} \in \mathfrak{R}_+^n$ ,  $i=[1,P]$ , com suas dimensões correspondendo a um dado número de parâmetros a serem otimizados. Os componentes de cada  $x_i$ , e os componentes de  $\sigma_{xi}$ ,  $i=[1,P]$ , são selecionados de acordo com uma distribuição uniforme em intervalo especificado *a priori*;

(ii) cada solução  $x_i$ ,  $i=[1,P]$ , é classificada com relação a função de adequação;

(iii) cada vetor de solução ancestral  $(x_i, \sigma_{xi})$ ,  $i=[1,P]$ , cria somente um vetor solução descendente  $(x_i', \sigma_{xi}')$ , de acordo com as seguintes equações:

$$\sigma_{xi}'(j) = \sigma_{xi}(j) + N\left(0, \frac{\sigma_{xi}(j)}{c}\right) \quad (1)$$

$$x_i'(j) = x_i(j) + N(0, \sigma_{xi}'(j)), \quad j=[1,n] \quad (2)$$

onde  $x_i(j)$ ,  $x_i'(j)$ ,  $\sigma_{xi}(j)$ , e  $\sigma_{xi}'(j)$  denotam o  $j$ -ésimo componente dos vetores  $x_i$ ,  $x_i'$ ,  $\sigma_{xi}$ , e  $\sigma_{xi}'$ , respectivamente. O termo  $N(\mu, \sigma_{xi})$  denota uma variável aleatória Gaussiana com média  $\mu$  e desvio padrão  $\sigma_{xi}$ , e  $c$  constitui-se de uma constante de escala;

(iv) cada vetor solução descendente  $x_i'$ ,  $i=[1,P]$ , é avaliado com relação a função de adequação;

(v) comparações são conduzidas sobre todas as  $x_i$  e  $x_i'$  soluções,  $i=[1,P]$ . Para cada solução,  $k$  oponentes (adota-se  $k=10$ ) são selecionados aleatoriamente de todos os vetores solução ancestrais e descendentes com igual probabilidade. Em cada comparação, se a solução considerada oferece pelo menos um desempenho tão adequado quanto o oponente selecionado aleatoriamente, ela recebe uma “vitória”;

(vi) das  $P$  soluções de  $x_i$  e  $x_i'$ ,  $i=[1,P]$ , os vetores soluções que apresentam mais “vitórias” são selecionados para serem ancestrais (pais) na próxima população, sendo que os vetores  $\sigma_{xi}'$  e  $\sigma_{xi}$  a elas associados são também incluídos;

(vii) repetir os passos (ii) a (vi) até que uma condição de parada seja satisfeita.

O procedimento mencionado para o paradigma *PE* podem ser sintetizados pelo pseudocódigo apresentado na Figura 2 (Bäck & Schwefel, 1993, Bäck *et alii*, 1997a).

```

t := 0;
inicializar: P(0) := {a_1(0), a_2(0), ..., a_mu(0)}
avaliar: P(0) := {Phi(a_1(0)), Phi(a_2(0)), ..., Phi(a_mu(0))}
enquanto uma condição de parada não é satisfeita
{
  realizar mutação: P'(t) := m_Om(P(t))
  avaliar: P'(t) := {Phi(a'_1(t)), Phi(a'_2(t)), ..., Phi(a'_lambda(t))}
  selecionar: P(t+1) := s_Os(P'(t) union Q)
  t := t + 1;
}

```

Figura 2. Pseudocódigo básico da *PE*

.As convenções utilizadas na Figura 2, são análogas as apresentadas na Figura 1, diferenciando-se apenas pelas seguintes convenções:

$\mu \geq 1$ : tamanho da população ancestral;

$\lambda \geq 1$ : tamanho da população descendente;

$m_{Om}$ : operador de mutação com parâmetros de controle  $Om$ ;

$s_{Os}$ : operador de seleção  $\exists s_{Os}: (I^\lambda \cup I^{\mu+\lambda}) \rightarrow I^\mu$ ;

$Q \in \{\emptyset, P(t)\}$ : conjunto de indivíduos considerados pela adição de passos de seleção, isto é, soluções ancestrais.

### 2.3 Estratégias Evolutivas

As estratégias evolutivas (*EEs*) foram desenvolvidas para a resolução de problemas técnicos de otimização em engenharia. Atualmente, constituem-se de importantes algoritmos computacionais em problemas de otimização de parâmetros.

A primeira *EE* desenvolvida foi a *EE*-(1+1), proposta por I. Rechenberg (1965) e H.-P. Schwefel (1965), no *Hermann Föttinger Institute for Hydrodynamics* da *Technical University of Berlin*, em experimentos realizados em um processo túnel de vento (Bäck *et alii*, 1997b).

A *EE*-(1+1), vem sendo progressivamente generalizada por variantes do número de ancestrais (pais),  $\mu > 1$ , e número de descendentes (filhos),  $\lambda > 1$ , por geração. As *EEs* com vários membros (*multi-membered*) têm o embasamento biológico relacionado a características de poligenia (característica do fenótipo de um indivíduo é influenciada por diversos genes) e pleiotropia (capacidade de um gene influenciar diversas características dos fenótipos).

As *EEs* com multimembros são divididas de acordo com o mecanismo de seleção, em: (i) *plus strategy* ou *EE*-( $\mu+\lambda$ ) sugerem a  $\mu$  ancestrais produzir  $\lambda$  filhos, após os  $\mu$  ancestrais e os  $\lambda$  descendentes competem pela sobrevivência e, (ii) *comma strategy* ou *EE*-( $\mu, \lambda$ ), onde os  $\lambda$  descendentes competem para sobreviver e os ancestrais são completamente substituídos a

cada geração (Coelho & Coelho, 1998a; Soucek & IRIS Group, 1992).

Os indivíduos,  $(x_i, \sigma_{xi})$ , de maneira análoga a *PE*, são diretamente representados por vetores de valores reais,  $x_i \in \mathfrak{R}^n$ , e incorporam um vetor adicional desvio padrão,  $\sigma_{xi} \in \mathfrak{R}_+^n$ .

O operador de mutação opera em cada variável,  $x_i$ , pela adição de números aleatórios normalmente distribuídos com média zero e variância  $\sigma_i^2$ , regidos pela notação  $N(0, \sigma_i^2)$ . Um novo vetor solução  $(x'_i, \sigma'_{xi})$  pode ser criado por uma regra de atualização com distribuição lognormal, tal que:

$$\sigma'_{xi} = \sigma_{xi} \cdot \exp\left(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)\right) \quad (3)$$

$$x'_i(j) = x_i(j) + N\left(0, \frac{\sigma'_{xi}(j)}{c}\right), \quad j=[1,n] \quad (4)$$

onde a mutação de  $\sigma_{xi}$  é baseada em um fator de busca global  $\tau' \cdot N(0,1)$  e um fator de busca local  $\tau \cdot N_i(0,1)$ . Estes fatores são regidos pelas seguintes equações:

$$\tau = \left(\sqrt{2\sqrt{n}}\right)^{-1} \quad (5)$$

$$\tau' = \left(\sqrt{2n}\right)^{-1} \quad (6)$$

onde  $n$  é o número de dimensões da função que está sendo otimizada.

Os operadores de recombinação são similares aos implementados em representação real nos *AGs* (Michalewicz, 1992). Entre as opções têm-se a recombinação discreta, a intermediária (local e global) ou mesmo a não realização da operação de recombinação (Bäck & Schwefel, 1993).

A implementação de *EEs*, apresenta diversas variantes, destacando-se as *EE-(μ, κ, λ, ρ)* contemporâneas e as *EEs* com mecanismos de auto-adaptação através da realização de mutações correlacionadas (Bäck & Schwefel, 1993; Beyer, 1995; Coelho & Coelho, 1998a; Rudolph, 1992). O pseudocódigo básico do ciclo evolutivo das *EEs*, não é apresentado, pois trata-se de um procedimento análogo ao pseudocódigo da *PE*, conforme tratado na Figura 2. Contudo, quanto as características das *EEs* em relação a *PE* existem algumas diferenças.

A seleção na *PE* é tipicamente probabilística realizada através de *tournament*, enquanto nas *EEs* é determinística. A operação de mutação na *PE*, usualmente, utiliza-se de distribuição Gaussiana, enquanto nas *EEs* é utilizada mutação com distribuição lognormal. Apesar que existem estudos comparativos do emprego de distribuição de Cauchy e uniforme nos operadores de mutação, em ambas *PE* e *EEs* (Bäck *et alii*, 1997a, 1997b).

A operação de recombinação na *PE* caracteriza-se por abstrair a evolução a nível de população, onde diversas espécies competem para obter parte dos recursos, enquanto, nas *EEs*, a reprodução realiza-se a nível de comportamento individual (Bäck & Schwefel, 1993; Coelho & Coelho, 1998a; Heitkoetter & Beasley, 1996).

## 2.4 Programação Genética

A programação genética (*PG*) é uma forma de *CE* que distingue-se por seu particular conjunto de escolhas para a representação, projeto de operadores genéticos e avaliação da função de adequação. A *PG* constitui-se também de uma extensão dos *AGs* no tratamento da complexidade de estruturas computacionais, visando a obtenção de soluções potenciais em um ambiente que imite o processo de Darwin (Howard & D'Angelo, 1995; Koza, 1992).

A *PG* utiliza um desenvolvimento eficiente de estrutura de dados para a geração de expressões simbólicas e executa regressões simbólicas direcionando a determinação simultânea da estrutura e complexidade requerida pelo modelo durante o processo evolutivo. A resolução de um problema através de *PG* pode ser tratado como uma busca através de possíveis combinações de expressões simbólicas definidas pelo projetista. Cada expressão é codificada em uma estrutura em árvore, também denominada de programa computacional, apresentando um comprimento variável e subdividida em nós.

Os elementos da *PG* são tipicamente conjuntos fixos de símbolos selecionados no tratamento da solução de problemas em domínio de interesse, permitindo a otimização de uma estrutura em árvore de maneira mais apropriada que apenas por parâmetros numéricos. Os elementos da *PG* são divididos em dois alfabetos, um funcional e outro terminal. O alfabeto funcional (não-terminal) é constituído, por exemplo, por um conjunto  $\{+, -, *, /, \text{sqrt}, \text{log}, \text{exp}, \text{ln}, \text{abs}, \text{and}, \text{or}\}$ , que inclui operações aritméticas, funções matemáticas, operações lógicas condicionais. O alfabeto terminal é um conjunto que inclui entradas (variáveis) apropriados para o domínio do problema, valores constantes e números.

O espaço de busca é um hiperespaço de todas as possíveis composições de funções que podem ser recursivamente compostas pelo alfabeto funcional e terminal. As expressões simbólicas (*S-expressions*) de uma linguagem de programação *LISP* (*List Processing*) são uma maneira especialmente conveniente de criar e manipular as composições de funções e terminais. Estes expressões simbólicas em *LISP* correspondem diretamente a uma *parse tree* que é internamente criada por muitos compiladores (Koza, 1992, 1994). A Figura 3 apresenta um diagrama de blocos típico de uma árvore de *PG*.

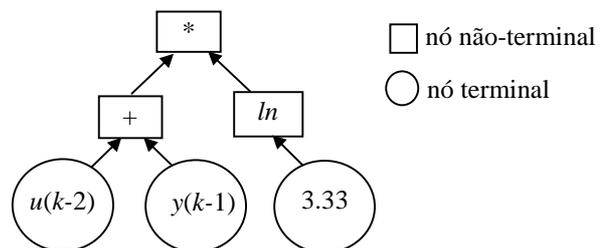


Figura 3. Diagrama de blocos típico de árvore em *PG*.

A operação de recombinação é implementada por meio de sub-árvores de indivíduos aleatoriamente selecionados e pela execução de permutações de termos das árvores. As operações de mutação consistem, geralmente, da troca de genes com imposição de restrições pelo projetista (McKay *et alii*, 1996a, 1996b; Gray *et alii*, 1996; Poli & Langdon, 1998).

O procedimento de otimização por *PG* pode ser dividido em um número seqüencial de passos (Bäck *et alii*, 1997a; Howard & D'Angelo, 1995):

- (i) criar aleatoriamente com distribuição uniforme uma população de árvores, provendo expressões simbólicas;
- (ii) avaliar cada árvore atribuindo-lhe o valor da função de adequação;
- (iii) utilizar uma técnica de reprodução pré-definida e assim copiar as árvores existentes para uma nova geração;
- (iv) aplicar o operador de recombinação em um conjunto de árvores ancestrais escolhidas aleatoriamente;
- (v) aplicar o operador de mutação;
- (vi) repetir os passos (ii) a (v) até que uma condição de parada seja satisfeita.

Durante a operação de recombinação duas árvores são escolhidas utilizando-se um dos métodos de seleção de forma similar aos métodos de seleção tratados pelos AGs. A operação de recombinação deve preservar a sintaxe das expressões simbólicas. Em outras palavras, a aplicação dos operadores genéticos deve produzir uma equação que possa ser avaliada.

Por exemplo, uma sub-árvore é selecionada aleatoriamente de uma árvore ancestral (pai) nº 1 é então trocada com uma sub-árvore aleatoriamente escolhida da árvore ancestral (pai) nº 2.

As árvores criadas são então inseridas no *mating pool* que será utilizado para formar a próxima geração. Uma operação de recombinação válida é apresentada a seguir, onde duas árvores ancestrais, conforme apresentadas na Figura 4, são tratadas.

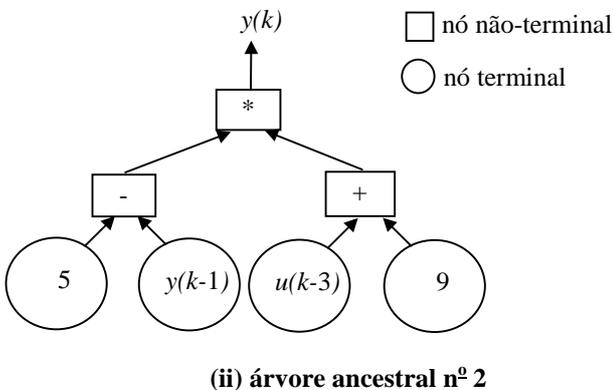
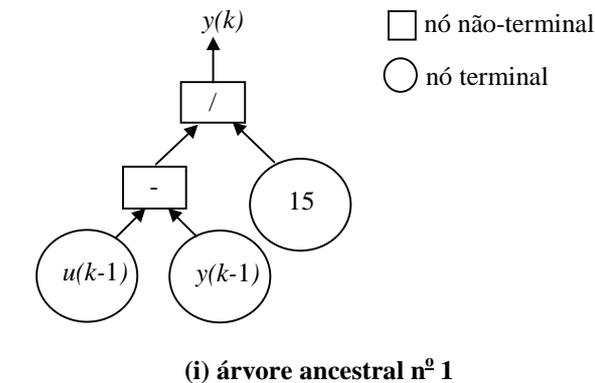


Figura 4. Representação das estruturas árvores ancestrais em PG.

Após a operação de recombinação em sub-árvores resulta na criação de duas árvores descendentes (filhos), conforme apresentadas na Figura 5.

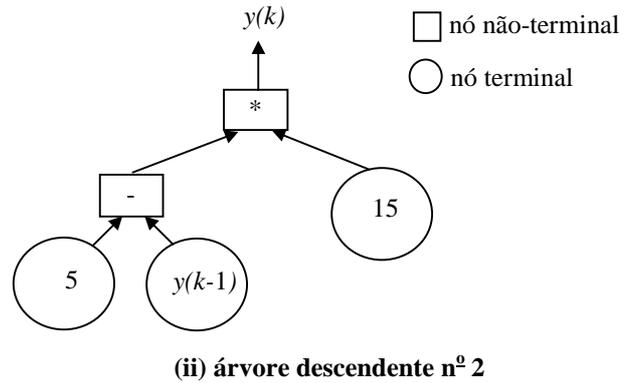
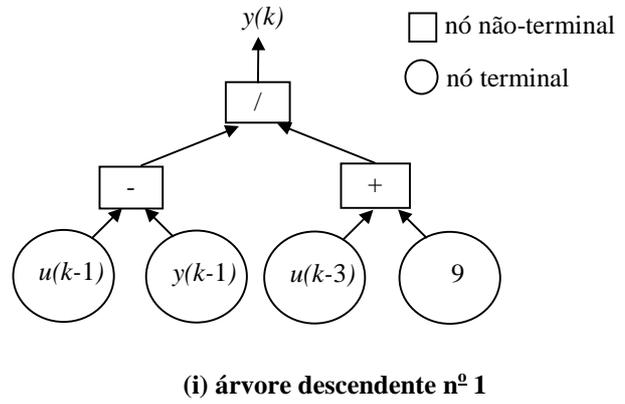


Figura 5. Representação das estruturas árvores descendentes resultantes da aplicação do operador de recombinação em PG.

O operador de mutação visa facilitar a exploração de diferentes partes do espaço de busca. A mutação cria um indivíduo modificado que é copiado para a próxima geração da população. A mutação consiste em síntese de uma mudança aleatória de uma função, uma entrada ou uma constante em uma expressão simbólica de uma população (Bäck *et alii*, 1997a).

Exemplo da aplicação do operador de mutação é a mudança do nó terminal de valor constante 15 da árvore apresentada na Figura 5(ii) para o valor 21, sendo esta operação mostrada na Figura 6.

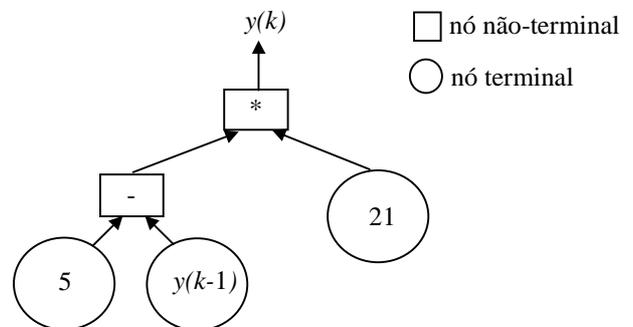


Figura 6. Exemplo da operação de mutação utilizando-se a árvore apresentada na Figura 5(ii).

A PG tem sido utilizada em diversas áreas do conhecimento, tais como: desenvolvimento de sistemas de classificação de imagens, projeto de filtros, equações apropriadas de sistemas

de equações não-lineares de sistemas caóticos, indução de árvore de decisão, detecção de características, otimização e evolução de redes neurais artificiais (Kinnear, 1994; McKay *et alii*, 1996a; 1996b). Mais informações referentes a base teórica e aplicações da *PG* podem ser encontrados em Bäck *et alii* (1997a), Kinnear (1994), Koza (1992,1994), e Soucek & The IRIS Group (1992).

## 2.5 Resumo das Características dos AEs

Os *AEs* apresentam similaridades no processo evolutivo, contudo algumas características diferem de uma técnica para outra e estão resumidas na Tabela 1. Contudo, é necessário ressaltar que existem variantes de *AEs* e na literatura tem sido propostas metodologias que apresentam características distintas ou mesmo combinação das apresentadas na Tabela 1.

**Tabela 1. Características dos AEs.**

(i) representação		
<i>AE</i>	<i>usual</i>	<i>variante</i>
<i>AG</i>	binária	inteira / real <sup>1</sup>
<i>PE</i>	real	híbrida inteira / real <sup>2</sup>
<i>EE</i>	real	híbrida inteira / real <sup>3</sup>
<i>PG</i>	comprimento variável	comprimento variável <sup>4</sup>
(ii) auto-adaptação		
<i>AG</i>	nenhuma	
<i>PE</i>	utiliza desvio-padrão e covariância <sup>3</sup>	
<i>EE</i>	utiliza desvio-padrão e covariância <sup>2,5</sup>	
<i>PG</i>	nenhuma	
(iii) função de adequação		
<i>AG</i>	não há conhecimento das propriedades	
<i>PE</i>	conhecimento da dimensão (nº de variáveis) <sup>3</sup>	
<i>EE</i>	conhecimento da dimensão (nº de variáveis) <sup>5</sup>	
<i>PG</i>	não há conhecimento das propriedades	
(iv) seleção		
<i>AG</i>	probabilística, preservativa	
<i>PE</i>	competitiva	
<i>EE</i>	determinística, extintiva	
<i>PG</i>	probabilística, preservativa	
(v) operação		
<i>AE</i>	<i>recombinação</i>	<i>mutação</i>
<i>AG</i>	operador principal	operador secundário
<i>PE</i>	operador secundário	operador principal (Gaussiano)
<i>EE</i>	operador secundário	operador principal (lognormal)
<i>PG</i>	operador principal	operador secundário
convenções (referências bibliográficas)		

<sup>1</sup> Michalewicz (1992)

<sup>2</sup> Bäck *et alii* (1997a, 1997b)

<sup>3</sup> Bäck *et alii* (1997a), Fogel (1995)

<sup>4</sup> Kinnear (1994), Koza (1992, 1994)

<sup>5</sup> Bäck & Schwefel (1993); Beyer (1995)

Os modelos apresentados na Tabela 1 têm sido aplicados em problemas nos mais diversos campos da engenharia, ciências naturais e economia, principalmente em aplicações com características não convexas, multimodais e multiobjetivo.

## 3 APLICAÇÕES EM IDENTIFICAÇÃO E CONTROLE DE PROCESSOS

O potencial de aplicabilidade da *CE* nas mais diversas áreas que interagem com automação industrial e desenvolvimento tecnológico são descritos e abordados a seguir.

### 3.1 Identificação e Linearização de Processos

A idéia básica da identificação de sistemas é permitir a construção de modelos matemáticos de um sistema dinâmico baseado em medidas, efetuado pelo ajuste de parâmetros com um dado modelo até que a saída coincida tão bem quanto possível com as saídas medidas.

As aplicações de *CE*, propostas na literatura, têm abordado a identificação de processos lineares com estimativa do atraso de transporte, linearização de processos não-lineares, incluindo-se neste contexto processos mono e multivariáveis, além de comparativos com técnicas convencionais em identificação, tais como mínimos quadrados recursivo (*MQR*), *MQR* ortogonal e filtro de Kalman estendido (Fleming & Fonseca, 1993; Tan *et alii*, 1995).

Os pesquisadores Kristinsson & Dumont (1992) tratam a identificação de processos por *AGs* e realizam comparações com a técnica da variável instrumental recursiva. Flockton & White (1993) descrevem a identificação de pólos e zeros para codificação da estrutura e coeficientes de um filtro digital por *AGs*. Tan & Li (1995) apresentam o desenvolvimento de *AGs* para sistemas de identificação do tipo norma  $L_{\infty}$ , no domínio da frequência, em aplicações de controle robusto. Li *et alii* (1995) tratam a utilização de *AG* híbrido com *simulated annealing* em problemas de identificação e linearização de processos.

Chen *et alii* (1996) aplicou *CE* em problemas de linearização no domínio da frequência de sistemas *MIMO* (*Multi-Input Multi-Output*) em um sistema não-linear de regulação de nível. Enquanto, Li & Jeon (1993) tratam a utilização de *AGs* para identificação dos regressores de sistemas *NARX* (*Non-linear Autoregressive model structure eXogenous inputs*) e efetuam uma abrangente comparação com o algoritmo dos *MQR* ortogonal.

Ma & Wu (1995) utilizam *PE* em identificação de parâmetros de um gerador. Kim *et alii* (1996) aplicam um mecanismo de aceleração da *PE* e o empregam na identificação e controle de uma mesa X-Y. Coelho & Coelho (1998a) apresentam um estudo comparativo de *AGs*, *EEs* e *AG* híbrido com *simulated annealing* aplicado a problemas de identificação e controle de processos experimentais. Fardin *et alii* (1997) aplicam os *AGs* em identificação de sistemas discretos, este um procedimento relevante à aplicação de técnicas de controle preditivo baseado em modelo. Lee *et alii* (1997) tratam o problema de modelagem em sistemas de potência por *PG*.

Koza (1992) apresenta um procedimento de identificação pela utilização de *PG* visando encontrar uma função da resposta ao impulso de um sistema linear invariante no tempo. A modelagem matemática de processos químicos complexos, tais como uma coluna de destilação e um processo de manufatura, tem sido realizada pela utilização da metodologia *PG* aliada a procedimentos de análise estatística, como alternativa a modelagem por técnicas de regressão e redes neurais artificiais são propostos por McKay *et alii* (1996a; 1996b).

## 3.2 Controle de Processos

O projeto e a configuração em sistemas de controle utilizando-se do paradigma *CE* têm abrangido uma variada gama de aplicações. As metodologias de *CE* têm sido empregadas devido as seguintes características:

- tratamento adequado de sistemas sujeitos a restrições;
- não requerimento de informações relativas a derivadas, estas usualmente necessárias nos métodos convencionais de otimização;
- adequação à implementação em paralelo;
- possibilidade de utilização do conhecimento *a priori* pelo projetista; e
- adequação para lidarem com sistemas complexos e espaços de busca multimodais e multiobjetivo.

A *CE* é uma ferramenta robusta em configuração e projeto de sistemas de controle, todavia geralmente é empregada através de um procedimento *off-line*. A sintonia do controlador de três parâmetros *PID* e controlador por alocação de pólos via *CE* têm sido descritas e analisadas por vários pesquisadores e aplicada em diversos processos do meio industrial.

Porter & Jones (1992) apresentam o projeto e sintonia de controle *PID* via *AGs* para processos monovariáveis. Kröhling (1997) apresenta o projeto de controle *PID* para rejeição de distúrbios pela utilização de funções de penalização e dois *AGs* aplicados a otimização e cooperando entre si.

O problema de regulação de sistemas hidráulicos via *AGs* é tratada por Nordvik & Renders (1991). O controle de um pêndulo invertido pela utilização de controle *bang-bang* é apresentado por Kawaji *et alii* (1994). Dimeo & Lee (1994) apresenta o projeto e sintonia de controle *PID MIMO* aplicado a uma caldeira. Varsek *et alii* (1993) tratam o projeto e a sintonia de controladores por *AGs* aplicados ao controle de um pêndulo invertido. Fogel (1995) apresenta um procedimento para identificação e controle visando o equilíbrio de um pêndulo invertido utilizando *PE*. Chipperfield & Fleming (1996) apresentam um *overview* da aplicação de *AGs* em controle de processos.

Aplicações militares e espaciais têm sido realizadas, mesmo que simuladas. Rock & Hirsh (1993) apresentam as vantagens e desvantagens da aplicação de *AEs* em projeto de aviões. Schultz & Grefenstette (1990) utilizam técnicas evolutivas ao problema do aprendizado de planos táticos em um simulador de vôo. Hull & Johnson (1994) aplicam *AGs* no projeto de controle *LQR* (*Linear Quadratic Regulator*) em controle de trajetória de mísseis. Zuo (1995) trata o projeto de controlador adaptativo digital por *AGs* aplicado a um modelo de simulação de uma estação espacial.

Os aspectos relativos a otimização multiobjetivo e esquemas baseados na definição de otimalidade de Pareto têm sido alvo de pesquisas por Chipperfield & Fleming (1996) e Fonseca & Fleming (1995). Liu & Patton (1994) apresentam o desenvolvimento de algoritmos de otimização multiobjetivo para controle robusto multivariável por *AGs*. Takahashi *et alii* (1997) apresentam um metodologia de otimização multiobjetivo  $H_2/H_\infty$  em projeto de compensadores *PID* para sistemas sujeitos a perturbações e incertezas paramétricas. Este estudo apresenta uma comparação de desempenho e complexidade computacional dos *AGs* de relação a algoritmos baseados em gradiente.

Marrison & Stengel (1997) tratam o projeto de compensadores em sistemas controle robusto aplicando métodos de busca aleatória e *AGs*. Patton & Liu (1994) propõem o projeto de controladores robustos *MIMO* combinado *AGs* à otimização baseada em gradiente. Saravanan (1995) apresenta a utilização de *PE* na síntese de controladores ótimos em uma variedade de problemas que dependem da otimização de um índice de desempenho. O projeto de controle de estrutura variável por *AGs* é proposto por Lin & Chen (1995). Projeto de controladores que satisfaçam o critério de  $H_2/H_\infty$  em processos *SISO* (*Single-Input Single-Output*) (Chen *et alii*, 1995) e *MIMO* (Parag & Lewin, 1996) também têm sido tratados na literatura.

A *PG* tem sido tratada em problemas de controle ótimo que envolvem a descrição de sistemas dinâmicos por variáveis de estado, como o tratamento diferenciado do clássico problema de controle do pêndulo invertido (Koza, 1992, 1994; Soucek & The IRIS Group, 1992).

Apesar das limitações oriundas da alta demanda computacional inerente as metodologias de *CE*, em algumas circunstâncias é possível abordar aplicações em tempo real, usualmente quando o modelo do processo é avaliado e o período de tempo é longo o suficiente para permitir o processamento e convergência dos *AEs*.

Outra situação é quando a *CE* é utilizada direcionando a adaptação *on-line* em um controlador projetado adequadamente *a priori*. Porém, para aplicação das otimizações propostas, existe a necessidade de processamento rápido da informação, pois a convergência das metodologias de *CE* é lenta, o que exige a aplicação de sistemas paralelos e/ou distribuídos de alta velocidade de processamento (Linkens & Nyongesa, 1995).

Outra área em constante avanço é a de sistemas autônomos, controle de trajetória de veículos, mecatrônica e robótica. Cavalcanti & Sales Jr. (1996) aplicam um *AE* híbrido, baseado em *AG* e *PE*, no controle e posicionamento de pêndulos de único e duplo braços. Ramos & Barone (1994) empregam os *AGs* em controle de trajetória de um veículo.

Variante & Bennaton (1996) utilizam *AGs* na sintonia de controladores convencionais aplicados ao controle de um manipulador robótico. Koza (1992, 1994) apresenta a utilização de *PG* em controle de robôs móveis, desvio de obstáculos, além do tratamento do problema clássico na área de controle inteligente denominado *truck backer-upper*. Fukuda & Shibata (1994) apresentam um *overview* da aplicação de controle neural, nebuloso e *AEs* em robótica.

As metodologias de *CE* combinadas com outras técnicas de *IC* podem ser configuradas nos três níveis do sistema, por exemplo, em controle adaptativo nebuloso para adaptação no nível de execução e, *AEs* no gerenciamento eficiente da escolha de uma estratégia ótima de coordenação e apropriado sequenciamento de operações abrangendo os níveis de coordenação e planejamento (Lee *et alii*, 1997; Passino, 1995; Sanz *et alii*, 1994).

## 3.3 Sistemas Híbridos

Diversas metodologias de *IC* combinadas a *CE* são propostas na literatura, destacam-se a hibridização com a teoria de controle convencional, sistemas especialistas, redes neurais artificiais, sistemas nebulosos e *simulated annealing*

(Dracopoulos & Jones, 1995; Schaffer, 1994; Linkens & Nyongesa, 1996b).

A configuração, projeto e sintonia de controladores lógicos nebulosos (*CLNs*) são confiados ao conhecimento heurístico do projetista, o que abre a perspectiva da utilização de *CE* com diversas variantes possíveis, principalmente em processos monovariáveis. Além disso, o projeto pode ser estendido para tratamento de processos multivariáveis.

Os enfoques usuais são em otimização de fatores de escala, funções de pertinência e base de regras. Os *CLNs* híbridos com *CE* servem à configuração de controladores *PD*, *PI* ou *PID*, controle adaptativo nebuloso (direto e indireto) e projetos de sistemas auto-ajustáveis.

Na literatura são encontrados diversos exemplos da combinação de sistemas nebulosos e *CE*. Karr & Sharma (1994) propõem a otimização de um sistema de controle de *pH* baseado em lógica nebulosa e *AGs*. Bäck & Kursawe (1994) apresentam um *overview* da utilização de *AGs* e *EEs* na sintonia das funções de pertinência de *CLNs*.

Gomide & Gudwin (1994) apresentam uma introdução aos princípios e idéias que fundamentam os *CLNs*, em geral, tratando também a aplicação de *AGs* no projeto destes controladores. Dumitrache & Buiu (1995) apresentam o projeto e análise de *CLNs* por *AGs* aplicado em controle de um sistema *truck backer-upper*. Inoue *et alii* (1997), Hoffmann & Pfister (1994), Linkens & Nyongesa (1995, 1996a) apresentam diversos métodos de projeto de *CLNs* por *AGs*.

Algoritmos de *simulated annealing*, princípios evolutivos de Lamarck e efeito Baldwin têm sido combinados à *CE*, visando aumentar a velocidade de convergência dos métodos frente a buscas locais (Ghoshray *et alii*, 1995; Kennedy, 1993). As técnicas de *CE* caracterizam-se por serem robustas e eficientes em busca globais, contudo deficientes quanto a ajustes finos nos valores obtidos (Coelho & Coelho, 1998a; Li *et alii*, 1995).

O paradigma computacional *CE* pode ser combinado com as redes neurais artificiais (*RNAs*) de três maneiras: (i) otimização das conexões e ajuste dos pesos, (ii) otimização da topologia da *RNA* e, (iii) ajuste das regras de aprendizado (coeficiente de aprendizado, fator de momento e funções de ativação).

As *RNAs* são aplicadas em identificação paramétrica linear e principalmente *black-box* de modelos dinâmicos não-lineares, com estrutura *NFIR* (*Non-linear Finite Impulse Response model*), *NARX*, *NOE* (*Non-linear Output Error model*), *NARMAX* (*Non-linear AutoRegressive Moving Average model structure with exogenous inputs*), *NBJ* (*Non-linear Box-Jenkin model structure*) ou por representação em espaço de estados são tratadas na literatura.

As arquiteturas de *RNAs* geralmente tratadas são as *feedforward* multicamadas e *feedback* (recorrentes), abrangendo, entre outras, as *RNAs* de Hopfield, *LVQ* (*Learning Vector Quantization*), *ART* (*Adaptive Resonance Theory*), *CMAC* (*Cerebellar Model Articulation Controller*).

A área denominada *neurocontrol* estuda a aplicação de *CE* em estratégias híbridas com as *RNAs* em controle supervisionado, controle inverso direto e controle adaptativo neural. Saravanan & Fogel (1995) tratam o controle de um pêndulo invertido duplo por otimização evolutiva de um controlador neural. Zitar & Hassoun (1995) propõem o projeto de controladores neurais por *AG* assistido por um sistema baseado em *reinforcement*

*learning* e aplicam esta configuração em diversos problemas de controle. Yao & Liu (1997) apresentam um novo sistema evolutivo para projeto de *RNAs* por *PE*.

Ichikawa & Sawa (1992) aplicam *AGs* para o aprendizado de controladores neurais diretos em processos simulados. Dracopoulos & Jones (1995) propõem o projeto de controladores neural evolutivos aplicados em controle de um modelo de satélite. Coelho & Coelho (1998b) apresentam o projeto de sistemas híbridos inteligentes pela sinergia de *CLNs*, *RNAs* e *CE* aplicados a controle de processos. Schaffer (1994) apresenta um *overview* das possibilidades de combinações de *AGs* com *RNAs* e sistemas nebulosos.

Um índice bibliográfico de *CE* aplicado à área de controle de processos é detalhado e classificado por Alander (1995), abrangendo identificação, controle, otimização, robótica, sistemas híbridos de *RNAs* e *CLNs* combinados a *CE*, entre outras áreas correlacionadas.

Outras áreas correlacionadas à automação industrial, que vêm abrigando intensas pesquisas, são projeto de circuitos *VLSI* (Bright & Arslan, 1996), escalonamento, identificação e tratamento de problemas do tipo *unit commitment* em sistemas de potência (Orero & Irving, 1996; Sheblé *et alii*, 1996), projeto de sistemas e pacotes de *CACSD* (*Computer-Aided Control System Design*) e *CAE* (*Computer-Aided Engineering*) (Drechsler *et alii*, 1996; Galante *et alii*, 1993) e integração de sistemas (Magid & Dawoud, 1995).

Outras técnicas de *CE* que não foram avaliados neste artigo, como os sistemas classificadores, são tratadas em aplicações para controle de processos por pesquisadores, a citar Huang & Fogarty (1991), Linkens & Nyongesa (1995), Odetayo & McGregor (1988), e Hahnert III & Ralston (1995).

### 3.4 Ferramentas de Software e Hardware

Entre as diversas ferramentas e ambientes computacionais disponíveis e aplicadas em identificação e controle de processos têm-se os *AGs* no ambiente do *MATLAB* (*FlexTool*), e nos softwares *GENOCOP*, *GENESIS*, *GENEsYs* e *GENITOR*.

Dentre as ferramentas que utilizam *EEs* e *PE* existem, por exemplo, o *ESCaPaDE* e o *LICE*. Enquanto que, para execução de *PG*, há implementações usualmente em *LISP* ou *C++*, exemplos são *GPEIST*, *GPQUICK*, *GEPETTO* e *GPC++* (Chipperfield & Fleming, 1996; Man *et alii*, 1996).

Quanto aos sistemas híbridos inteligentes, têm-se o *Gensym* e o *NeuroShell 2*, que utilizam *CE* para otimização de *RNAs*, e o *EnGENEous*, que alia a *CE* com sistemas especialistas. Uma lista detalhada e abrangente de ferramentas de desenvolvimento e softwares sem finalidade comercial está disponível na Internet via *ftp/www*. Lista de pacotes comerciais disponíveis no mercado, conceitos básicos, referências e eventos relevantes ligados a *CE* podem ser encontradas em: EvoNews (1996, 1998), Filho *et alii* (1994), Heitkoetter & Beasley (1996), Man *et alii* (1996), e Vemuri & Cedeño (1997).

As aplicações em *hardware* têm sido abordadas principalmente a nível acadêmico e de pesquisa, em projeto de *chips* e programação de microcontroladores com implementação de sistemas híbridos por meio de controladores nebulosos, neurais, neuro-nebulosos e teoria de autômatos finitos, onde os *AGs*, *PE*, *EEs* e *simulated annealing* aliam-se na otimização

off-line dos controladores inteligentes (Chiaberge & Reyneri, 1995).

O projeto de circuitos VLSI (*Very-Large Scale Integration*) também tem sido alvo de intensas pesquisas e publicações científicas, e uma área emergente é a de *hardware* evolutivo, onde a evolução artificial pode operar no sentido de reconfiguração de circuitos eletrônicos para produção de poderosos e eficientes sistemas de controle, principalmente em aplicações de robôs móveis e autônomos, por meio de um tipo de circuito VLSI denominado *FPGA (Field Programmable Gate Array)* (Bright & Arslan, 1996; Drechsler *et alii*, 1996; EvoNews, 1996; Thompson, 1995).

A implementação em processamento paralelo e de agentes distribuídos é facilitada por aspectos de simplicidade de adaptação dos AEs à configuração de importantes mecanismos, tais como presa-predador, migração e modelos de difusão. A utilização de diversos processadores possibilita um aumento de ganho em tempo computacional, em relação a máquinas com processamento seqüencial (Fogarty & Bull, 1995; Kawachi *et alii*, 1995).

## 4 EXEMPLOS PRÁTICOS

As pesquisas e os trabalhos desenvolvidos pelos autores na área de AEs objetivam tópicos relativos a (Coelho & Coelho, 1998a, 1998b):

- análise das vantagens e desvantagens da utilização de AEs em problemas de sintonia de parâmetros em identificação de processos lineares e não-lineares;
- projeto automático e avaliação de controladores inteligentes com configuração via AEs;
- estudo comparativo de características de complexidade, convergência e mecanismos auto-adaptação em AEs;
- desenvolvimento de ferramentas de controle inteligente através da hibridização com outros paradigmas da IC.

Nesta seção, exemplos da aplicabilidade em identificação de alguns AEs são apresentados em identificação experimental e estimação do atraso de transporte de um processo térmico descritos em Coelho (1995).

Este artigo trata também da sintonia de um controlador *PID*, visando o controle de um sistema não-linear de regulação de nível, composto de dois tanques acoplados, sujeito a mudanças de referência e perturbações de carga. As diferentes metodologias empregadas são: *AG*, *AG* híbrido com *simulated annealing (AGSA)*, *PE* e *Ees*. A função de adequação,  $F(J)$ , das metodologias da *CE*, tanto para identificação quanto controle adotadas, tem por meta a maximização de

$$F(J) = \frac{\alpha}{1+J} \quad (7)$$

onde  $\alpha$  é um fator de escala. Nas implementações adota-se  $\alpha=1$  em identificação do processo térmico,  $\alpha=15 \times 10^4$  em controle de nível, e  $J$  é a função custo ou critério de erro adotado pelo projetista.

### 4.1.1 Identificação e Linearização de Processo Térmico

O modelo matemático paramétrico utilizado na estimação dos parâmetros do processo tem a seguinte expressão:

$$A(z^{-1})y(k) = B(z^{-1})u(k-d) + v(k) \quad (8)$$

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-na} \quad (9)$$

$$B(z^{-1}) = b_1z^{-1} + b_2z^{-2} + \dots + b_nz^{-nb} \quad (10)$$

onde  $y(k)$ ,  $u(k)$ ,  $v(k)$  e  $d$  são a saída, entrada, ruído e atraso de transporte, respectivamente. O ruído é uma seqüência normal de números aleatórios com média zero e variância unitária. O objetivo das metodologias aplicadas é encontrar os parâmetros adequados para os polinômios  $A(z^{-1})$ ,  $B(z^{-1})$  e  $d$ .

O critério,  $J$ , considerado para estimação dos modelos discretos estimados é a minimização da integral do erro de previsão quadrático, *ISE (Integral Square-Error)*, isto é:

$$J = ISE = \int e^2(t) dt = \sum_{k=1}^{NA} [y(k) - \hat{y}(k)]^2 \quad (11)$$

A avaliação do erro de estimação também é realizada pelo critério de erro de previsão relativo, *RPE (Relative Prediction-Error Criterion)*, regido por

$$RPE = \frac{\sum_{k=1}^{NA} [y(k) - \hat{y}(k)]^2}{\sum_{k=1}^{NA} [y(k)]^2} \quad (12)$$

onde  $y(k)$  é a saída desejada,  $\hat{y}(k)$  é a saída estimada e,  $NA$  é o número de amostras coletadas do experimento.

### 4.1.2 Descrição do Processo Térmico

O processo térmico é um tanque retangular metálico de dimensões 15cm x 13cm x 23cm, com capacidade para 4,5 litros. Os diversos componentes do sistema de temperatura são: um sensor de temperatura do tipo *PT-100* conectado a um circuito de transmissão para escalonamento (0 a 5 V) e amplificação do sinal do sensor (faixa de 25°C até 90°C) e uma resistência elétrica industrial de 750 W (Coelho, 1995).

O processo é submetido a uma entrada e são coletadas 400 amostras do processo para um período de amostragem de 5 s. A Figura 7 apresenta o diagrama da configuração adotada para o processo térmico.

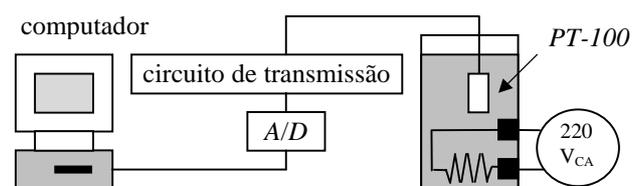


Figura 7. Configuração do processo térmico.

### 4.1.3 Resultados Experimentais

No projeto dos algoritmos implementados adotam-se os parâmetros apresentados na Tabela 2. Nos experimentos em identificação de processos configura-se o intervalo de estimação do atraso de transporte entre [0;10], o parâmetro tamanho de população de 30 indivíduos e o número máximo de 500 gerações de ciclo evolutivo.

Uma observação relevante é que nos experimentos, tanto em identificação quanto controle, utiliza-se representação real nos *Ags* e *AGSA* e os operadores de recombinação e mutação propostos por Michalewicz (1992).

**Tabela 2. Parâmetros usados nas técnicas de CE.**

Téc.nº	técnica	$\sigma$	seleção	recombinação	mutação
1	AG	-	roleta	* $p_r = 0.8$	# $p_m = 0.2$
2	AGSA	-	roleta	* $p_r = 0.8$	# $p_m = 0.2$
3	AGSA	-	breeder	* $p_r = 0.8$	# $p_m = 0.2$
4	EE-(5,25)	1	-	**	eq.(3)-(4)
5	EE-(5+25)	1	-	**	eq.(3)-(4)
6	EE-(1,29)	1	-	-	correlatas
7	PE	1	-	-	eq.(1)-(2)

convenções	
$p_r$	probabilidade de recombinação aritmética
$p_m$	probabilidade de mutação não-uniforme
**	recombinação discreta (x) e aritmética intermediária ( $\sigma$ ).

A Tabela 3 apresenta os melhores resultados obtidos por CE, nos 10 experimentos realizados com cada técnica, em identificação experimental do processo térmico. As respostas real e estimada da modelagem mais adequada do processo térmico em malha aberta são apresentadas na Figura 8.

**Tabela 3. Resultados da identificação do processo térmico.**

Intervalo de busca dos polinômios $A(z^{-1})$ e $B(z^{-1})$ de [-2;2]									
nº	na	nb	$a_1$	$a_2$	$b_1$	$b_2$	d	ISE	RPE
1	2	2	-.173	-.823	-1,86	1,87	9	114,2	$2,3 \times 10^{-2}$
2	1	1	-.995	-	,006	-	3	3,97	$7,9 \times 10^{-4}$
3	1	1	-.995	-	,006	-	7	3,95	$7,8 \times 10^{-4}$
4	1	1	-.991	-	,009	-	3	21,56	$4,3 \times 10^{-3}$
5	1	2	-.991	-	-1,18	1,19	3	16,02	$3,2 \times 10^{-3}$
6	1	1	-.995	-	,006	-	2	3,92	$7,8 \times 10^{-4}$
7	1	1	-.995	-	,006	-	2	3,92	$7,8 \times 10^{-4}$

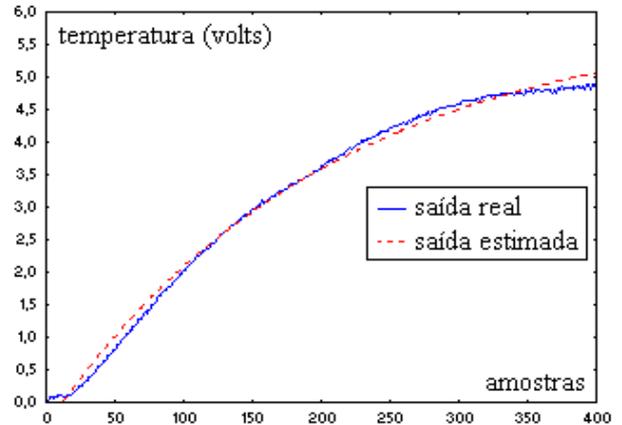
  

intervalo de busca dos polinômios $A(z^{-1})$ e $B(z^{-1})$ de [-1;1]									
nº	na	nb	$a_1$	$a_2$	$b_1$	$b_2$	d	ISE	RPE
1	1	1	-.995	-	,006	-	4	4,08	$8,1 \times 10^{-4}$
2	1	1	-.995	-	,006	-	7	3,93	$7,8 \times 10^{-4}$
3	1	1	-.995	-	,006	-	6	3,95	$7,8 \times 10^{-4}$
4	1	2	-.996	-	,472	-.466	3	12,82	$2,6 \times 10^{-3}$
5	1	2	-.991	-	,712	-.702	4	24,70	$4,9 \times 10^{-3}$
6	1	1	-.995	-	,006	-	2	3,92	$7,8 \times 10^{-4}$
7	1	1	-.995	-	,006	-	2	3,92	$7,8010^{-4}$

As convenções estão definidas de acordo com as equações (8)-(12) apresentadas na seção 4.1.

Nos experimentos realizados foram testadas a identificação do processo de baixa ordem, combinando as possíveis soluções no intervalo  $a \in \mathfrak{R} \forall na \in [1;2]$  e  $b \in \mathfrak{R} \forall nb \in [1;2]$ .

Na análise das Tabelas 2 e 3, observa-se que a estimativa de parâmetros mais adequada são as realizadas pelas metodologias PE e EE-(1,29) com mutações correlatas em estimação dos parâmetros do processo térmico, com  $a_1 = -0,9950$ ;  $b_1 = 0,0058$ ; e atraso de transporte,  $d=2$ . O resultado é utilizado para plotar os gráficos da resposta real e estimada do processo térmico em malha aberta, conforme apresentados na Figura 8.



**Figura 8. Gráficos de resposta real e estimada do processo térmico em malha aberta.**

### 4.2 Controle de um Processo de Nível

O controlador PID é o controlador mais popular no meio industrial devido à sua simplicidade de implementação e baixo custo. A sintonia adequada dos parâmetros  $K_p$  (ganho proporcional),  $T_i$  (tempo integral) e  $T_d$  (tempo derivativo) de acordo com um período de amostragem ( $T_s$ ), o PID proporciona um apropriado comportamento dinâmico ao processo controlado.

As técnicas tradicionais de sintonia baseiam-se em métodos como domínio da frequência, Ziegler-Nichols e heurísticos de tentativa e erro, como refinamentos para satisfazer as condições de projeto. Nesta seção, adota-se a sintonia por CE como metodologia alternativa à sintonia do PID (PID-CE). A forma padrão do controle PID digital é dada pela equação a diferenças:

$$u(k) = u(k-1) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) \quad (13)$$

onde as constantes  $q_0$ ,  $q_1$ , e  $q_2$  satisfazem

$$q_0 = K_p \left( 1 + \frac{T_s}{2T_i} + \frac{T_d}{T_s} \right) \quad (14)$$

$$q_1 = -K_p \left( 1 + \frac{2T_d}{T_s} - \frac{T_s}{2T_i} \right) \quad (15)$$

$$q_2 = K_p \frac{T_d}{T_s} \quad (16)$$

O critério de erro adotado para sintonia do controle PID é a minimização no tempo de uma função custo,  $J(u,e)$ , dada pela soma do critério de erro ITSE (Integral Time Squared Error) e a variação do controle ponderada ( $w=0,1$ ), tal que,

$$J(u,e) = \sum_{k=0}^{NA} k \left[ e(k)^2 + w \Delta u(k)^2 \right] \quad (17)$$

onde  $NA$  é o número de amostras,  $e(k)$  é o erro dado pela diferença entre a saída  $y(k)$  do processo e a referência ( $sp$ ) desejada e,  $\Delta u(k)$  é a variação do sinal de controle. A configuração para o projeto e controle  $PID$  obedece a estrutura apresentada na Figura 9.

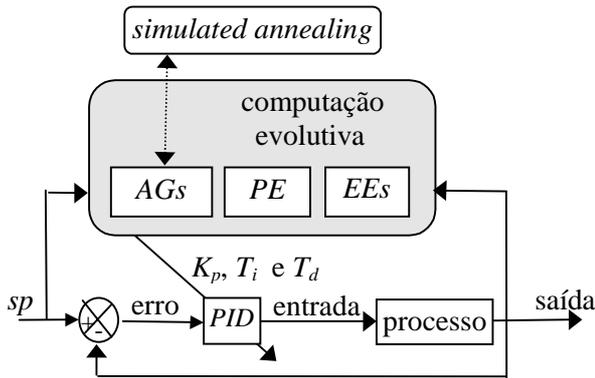


Figura 9. Controlador  $PID$  com sintonia por  $CE$ .

Nota-se pela Figura 9, que a sintonia dos três parâmetros do controlador  $PID$  por  $CE$  necessita apenas da informação do erro do processo, contudo, uma desvantagem desta metodologia de sintonia é que necessita-se várias avaliações da função custo para evoluir-se para um conjunto adequado de parâmetros para o controlador  $PID$ .

#### 4.2.1 Descrição do Processo

O sistema de regulação de nível não-linear é formado por dois tanques acoplados conforme proposto por Li *et alii* (1995) e está ilustrado na Figura 10.

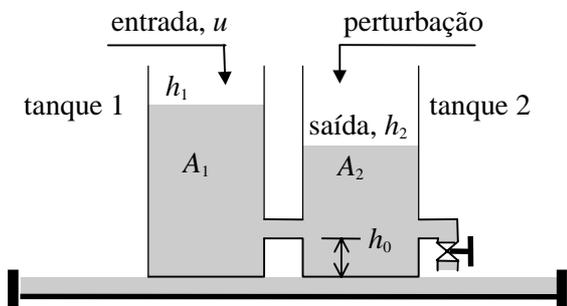


Figura 10. Sistema de regulação de nível não-linear.

As equações que caracterizam a dinâmica em malha aberta do processo são

$$A_1 \dot{h}_1 = u - a_1 c_1 \sqrt{2g(h_1 - h_2)} \quad (18)$$

$$A_2 \dot{h}_2 = a_1 c_1 \sqrt{2g(h_1 - h_2)} - a_2 c_2 \sqrt{2g(h_2 - h_0)} \quad (19)$$

$$y = h_2(t - \tau) \quad (20)$$

onde a área dos tanques é  $A_1=A_2=97 \text{ cm}^2$ ; a área dos orifícios é  $a_1=0,396 \text{ cm}^2$  e  $a_2=0,395 \text{ cm}^2$ ; as constantes de descarga são  $c_1=0,53$  e  $c_2=0,63$ ; a altura dos orifícios é  $h_0=3 \text{ cm}$ , a aceleração da gravidade é  $g=981 \text{ cm/s}^2$ , e o atraso de transporte inserido no sistema é de  $\tau=6$  unidades de tempo.

#### 4.2.2 Resultados Experimentais

Os ensaios realizados no processo não-linear de nível visam manter o nível de líquido do tanque 2 durante 600 iterações, em cada uma das três referências propostas de 11 cm, 6 cm e 9 cm, respectivamente, com um mínimo de sobre-elevação e erro nulo em regime.

Quanto ao projeto dos controladores  $PID-CE$  foram otimizados apenas para as mudanças de referência, não prevendo perturbações de carga. As perturbações de carga em cerca de 25% do controle máximo ( $8,325 \text{ cm}^3/\text{s}$ ) foram aplicadas ao tanque 2, nos instantes 300 a 600, 900 a 1200 e 1500 a 1800, após o projeto dos controladores, para análise da robustez, sensibilidade e atuação do controle  $PID$  frente a situações imprevistas no projeto do controlador. O sinal do controle  $u$  de entrada do líquido no tanque 1 é limitado à faixa de  $[0; 33,3 \text{ cm}^3/\text{s}]$  e obtém-se a saída,  $y$ , da planta calculada, utilizando-se o método de Runge-Kutta de 4ª ordem (Li & Häubler, 1996).

O intervalo de busca dos três parâmetros do  $PID$  estabelecido no projeto é  $[0;200]$ , máximo de 30 gerações de ciclos de evolução e população de 30 indivíduos. As técnicas utilizadas são as mesmas expostas na Tabela 2, em identificação do processo térmico. A Tabela 4 apresenta os melhores resultados obtidos na sintonia do  $PID$  com a realização de 10 experimentos para cada metodologia de  $CE$ , enquanto na Figura 11 é apresentada uma comparação de convergência da função custo,  $J(u,e)$ , nos 10 experimentos (média) em sintonia do  $PID$ .

Tabela 4. Resultados em sintonia do  $PID$  via  $CE$ .

$n^e$	$K_n$	$T_i$	$T_d$	$ts_1$	$tp_1$	$o_1\%$	$\Delta u_1$	$ts_2$
1	149,63	14,94	12,12	276	293	0,45	332,7	77
2	30,80	26,46	0,00	280	291	0,15	111,4	145
3	30,23	24,07	0,04	278	291	0,29	111,1	144
4	199,51	19,07	4,76	282	295	0,13	332,8	84
5	81,87	26,70	0,04	322	329	0,06	111,8	199
6	17,96	30,26	0,03	278	298	0,47	110,9	85
7	18,41	22,12	0,02	280	292	0,50	112,0	97

$n^e$	$TP_2$	$o_2\%$	$\Delta u_2$	$ts_3$	$tp_3$	$o_3\%$	$\Delta u_3$	$J(u,e)$
1	94	5,75	316,7	110	127	1,30	243,8	209355
2	159	0,60	106,2	144	123	0,99	23,0	201965
3	160	0,95	94,9	111	123	0,69	22,2	201886
4	98	1,72	316,3	116	129	0,37	243,7	209071
5	204	0,02	98,0	157	158	0,01	22,7	203418
6	100	1,13	95,4	121	120	1,24	22,3	201875
7	97	1,04	94,4	111	130	1,21	21,7	201863

Convenções

$ts_i$	tempo de subida
$tp_i$	tempo de pico
$o_i\%$	porcentagem de sobre-elevação ( <i>overshoot</i> )
$\Delta u_i$	variação do sinal de controle
$i$	referência, onde: $i = 1$ (amostras 1 a 600), $i = 2$ (amostras 601 a 1200), e $i = 3$ (amostras 1201 a 1800)

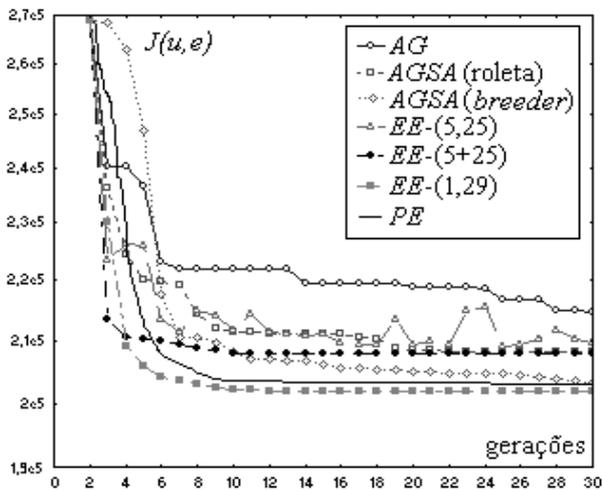


Figura 11. Comparação da função custo nos 10 experimentos (média) para sintonia do PID.

Pela análise dos resultados obtidos conforme mostrados na Tabela 3 constata-se, de maneira análoga aos expressados em identificação do processo térmico, que a técnica *EE*-(1,29) é a que proporciona o resultado mais adequado na sintonia dos parâmetros do controlador *PID* em  $K_p=17,96$ ;  $T_i=30,96$  e  $T_d=0,03$ , com valor obtido de  $J(u,e)=201863$ , e função de adequação  $F(J)=0,74$  além de apresentar uma rápida convergência em relação aos outros paradigmas utilizados.

A Figura 12 apresenta os resultados, da fase de sintonia, do *PID* pela técnica *EE*-(1,29) na resposta em malha fechada sujeito a mudanças de referência do *PID-CE*, enquanto na Figura 13 é mostrada a resposta, da fase de testes, do controle *PID* em malha fechada a mudanças de referência e perturbações de carga do *PID-CE* otimizado por *EE*-(1,29). Os testes mostram que a sintonia do *PID-CE* é adequada, apesar de pequenas flutuações na saída do tanque 2, apresenta um desempenho satisfatório quanto a variações no sinal de controle aplicado a planta, conforme ilustra a Figura 13.

Nota-se, pelos resultados nos casos estudados, que a *EE* com utilização de mutações correlatas, devido a suas características de aprendizado de segundo nível, via auto-adaptação, comporta-se melhor comparando-se aos *AGs* e *PE* analisados. Contudo, os estudos estão sendo realizados em vinculação de aprendizado com adaptação dos parâmetros de configuração em *AGs*, *AGSA* e *PE*. Como desvantagem, a implementação de *EEs* com mutações correlatas requer uma maior complexidade algorítmica que a demandada em implementação de *AGs*, *AGSA* ou *PE*.

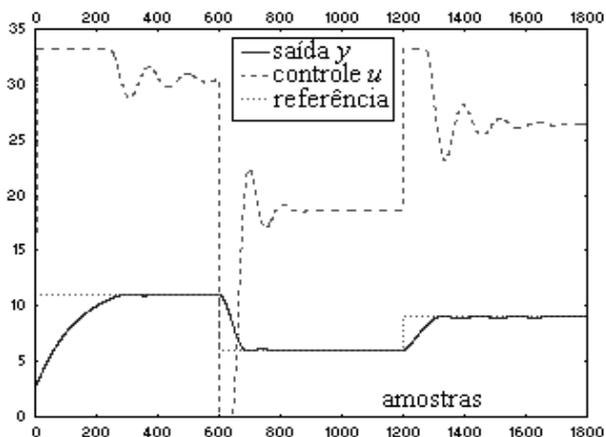


Figura 12. Resposta em malha fechada a mudanças de referência do *PID-CE* (fase de sintonia).

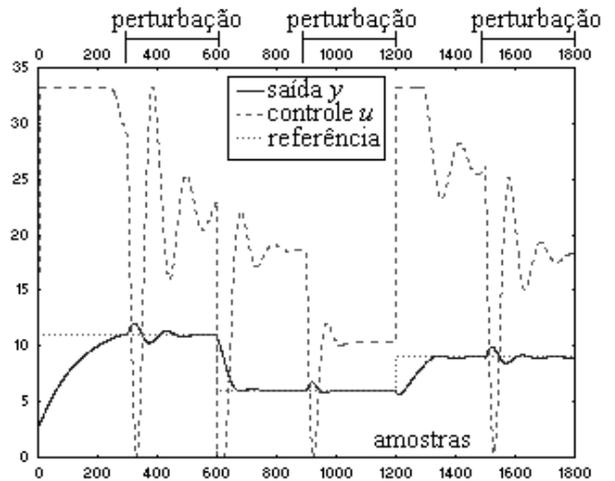


Figura 13. Resposta em malha fechada a mudanças de referência e perturbações de carga do *PID-CE* (fase de testes).

## 5 CONCLUSÃO E PERSPECTIVAS

Este artigo apresentou uma visão geral da aplicabilidade do paradigma *CE* em identificação e controle de processos industriais. Os algoritmos evolutivos descritos e analisados foram os algoritmos genéticos, programação evolutiva, estratégias evolutivas e programação genética.

Os paradigmas da *CE* apresentaram comportamento adequado na tarefa de identificação do processo térmico. Entretanto, nota-se que cada *AE* apresenta peculiaridades quanto a sintonia de seus parâmetros de configuração que podem afetar e muito o seu desempenho. Em *AGs* e *AGSA*, este fato está vinculado principalmente a escolha da estratégia de seleção, probabilidades de recombinação e mutação, e tamanho da população. Na *PE* e em *EEs*, o desempenho da destas técnicas está ligado a escolha adequada do número de ancestrais e descendentes, qual a distribuição mais adequada para utilizar-se na no operador de mutação, implementação ou não de mecanismos de auto-adaptação, entre outros fatores que influenciam as propriedades de convergência de cada metodologia.

Estudos posteriores em identificação visam a implementação de métodos à validação da estimativa dos parâmetros por meio da utilização de sinais Gaussianos (ruído branco), *PRBS* (*Pseudo-Random Binary Sequence*), e modelos matemáticos diferentes para cada referência (múltiplos modelos).

Quanto a sintonia de controle *PID* aplicado ao controle de um sistemas de regulação de nível nota-se pelas Figuras 11, 12 e 13, que os *AEs* apresentaram resultados adequados na sintonia de controle *PID*, podendo estes serem estendidos para projeto de outros controladores inteligentes e mesmo visar o tratamento de aplicações em processos *MIMO*.

Os experimentos em identificação e controle apresentados levam a concluir-se que apesar da literatura apresentar um espectro abrangente e multidisciplinar em aplicações do paradigma computacional *CE*, algumas questões de projeto de algoritmos evolutivos permanecem em aberto e tem sido alvo de intensas pesquisas na comunidade de *CE*.

As questões relevantes englobam fatores que devem ser analisados mais profundamente, tais como: (i) regras determinísticas na seleção dos parâmetros de configuração dos algoritmos; (ii) estudo das propriedades de convergência, no sentido de melhorar a eficiência e robustez da *CE* para

aplicações em tempo real; (iii) sedimentação da teoria matemática; (iv) mecanismos de auto-adaptação e auto-organização; e (v) eficiente hibridização de mecanismos evolutivos entre paradigmas evolutivos.

A CE apresenta vantagens e desvantagens em relação aos métodos tradicionais e este aspecto serve para enfatizar a necessidade de não abandonar-se os métodos convencionais de otimização (por exemplo, na sintonia do controlador PID). Porém, o projetista pode utilizar-se de métodos evolutivos como uma alternativa no auxílio a resolução de problemas onde os métodos convencionais podem falhar ou não terem desempenho adequado. Em síntese, os AEs convencionais são eficientes na busca global, mas são relativamente lentos em sintonia local, o que conduz à natural integração com métodos numéricos eficientes de *hill climbing* em buscas locais.

De acordo com o teorema *no free lunch* (NFL) (Wolpert & Macready, 1997) não existe algum algoritmo para resolução de todos problemas de otimização que é genericamente (em média) superior que algum outro algoritmo competidor. A questão se os AEs são inferiores ou superiores a algum método alternativo é insensata. O que pode ser afirmado somente é que AEs comportam-se melhor que outros métodos com respeito a resolução de uma classe específica de problemas, e como consequência que eles comportam-se pior para outras classes de problemas.

O teorema NFL pode ser confirmado pela análise dos AEs em relação a muitos métodos clássicos de otimização. Os métodos clássicos são mais eficientes para resolução de problemas lineares, quadráticos, fortemente convexos, unimodais, separáveis, e em muitos outros problemas em especial. Por outro lado, os AEs tem sido utilizados nos mais diversos problemas quando estes são descontínuos, não diferenciáveis, multimodais, ruidosos, e de outro modo superfícies de resposta não convencionais são envolvidas (Bäck *et alii*, 1997a; Wolpert & Macready, 1997).

Em síntese, as perspectivas da utilização e os avanços na teoria e na prática dos AEs estão intrinsecamente ligados ao caminho a ser seguido pelos avanços tecnológicos e à crescente integração de máquinas ditas “inteligentes” no meio industrial e de bens de consumo.

## AGRADECIMENTOS

Os autores agradecem aos revisores anônimos por seus comentários úteis e construtivos, que serviram à melhoria da redação e abrangência deste artigo.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Alander, J.T. (1995). An Indexed Bibliography of Genetic Algorithms in Control. Report 94-1, University of Vaasa, Dep. of Information Technology and Production Economics, Finland.
- Bäck, T. & H.-P. Schwefel (1993). An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation*, vol. 1, no. 1, pp. 1-23. The MIT Press.
- Bäck, T. & F. Kursawe (1994). Evolutionary Algorithms for Fuzzy Logic: A Brief Overview. *Proc. of 5th Int. Conf. on Information Processing and Management of*

*Uncertainty in Knowledge-based Systems*, Paris, France, pp. 659-664.

- Bäck, T., D.B. Fogel & Z. Michalewicz (eds.) (1997a). *Handbook of Evolutionary Computation*. Institute of Physics Publishing, Bristol, Philadelphia and Oxford University Press, New York, Oxford.
- Bäck, T., U. Hammel & H.-P. Schwefel (1997b). Evolutionary Computation: Comments on the History and Current State. *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, pp. 3-17.
- Beyer, H.-G (1995). Toward a Theory of Evolution Strategies: Self-adaptation. *Evolutionary Computation*, vol. 3, no. 3, pp. 311-348. The MIT Press.
- Bright, M.S. & T. Arslan (1996). Genetic Framework for the High Level Optimisation of Low Power VLSI DSP Systems. *Electronics Letters*, vol. 32, no. 13, pp. 1150-1151.
- Cavalcanti, J.H.F. & E.F. Sales Jr. (1996). Movimento do Robô-Trapezista usando Computação Evolucionária. *XI Congresso Brasileiro de Automática*, São Paulo, SP, Brasil, p. 567-572.
- Chen, B.-S., Y.-M. Cheng & C.-H. Lee (1995). A Genetic Approach to Mixed  $H_2/H_\infty$  Optimal PID Control. *IEEE Control Systems*, vol. 15, no. 5, pp. 51-60.
- Chen, K.C., M.R. Gong & Y. Li (1996). Evolutionary Linearisation in the Frequency Domain. *Electronic Letters*, vol. 32, no. 1, pp. 74-76.
- Chiaberge, M. & L.M. Reyneri (1995). Cintia: A Neuro-Fuzzy Real-Time Controller for Low-Power Embedded Systems. *IEEE Micro*, June, pp. 40-47.
- Chipperfield, A. & P. Fleming (1996). Evolutionary Algorithms for Control Engineering. *13th Triennial World Congress IFAC*, San Francisco, CA, USA, pp. 181-186.
- Coelho, A.A.R. (1995). Laboratory Experiments for Education in Process Control. *Workshop on Control Education and Technology Transfer Issues*, Curitiba, Brazil, pp.133-138.
- Coelho, L.S. & A.A.R. Coelho (1998a). Genetic Algorithms and Evolution Strategies Applied in Identification and Control: Case Study. *Soft Computing in Engineering Design and Manufacturing*, Chawdhry, P. K., R. Roy & R. K. Pant (eds.), Springer-Verlag, Berlin, Germany, pp. 430-438.
- Coelho, L.S. & A.A.R. Coelho (1998b). Computational Intelligence in Process Control: Fuzzy, Evolutionary, Neural, and Hybrid Approaches. *Int. Journal of Knowledge-Based Intelligent Engineering Systems*, vol. 2, no. 2, pp. 80-94.
- Davis, L. (ed.) (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, USA.
- Dimeo, R.M. & K.Y. Lee (1994). Genetic-Based Control of MIMO Boiler-Turbine Plant. *Proc. of the 33rd Conf. on Decision and Control*, Lake Buena Vista, FL, USA, pp. 3512-3517.

- Dracopoulos, D.C. & A.J. Jones (1995). Neural Networks and Genetic Algorithms for the Attitude Control Problem. *Int. Workshop on Artificial Neural Networks*, Malaga, Spain.
- Drechsler, R., H. Esbensen & B. Becker (1996). Genetic Algorithm in Computer Aided Design of Integrated Circuits. *1st Online Workshop on Soft Computing*.
- Dumitrache, I. & C. Buiu (1995). Hybrid Geno-Fuzzy Controllers. *IEEE Int. Conf. on Systems, Man and Cybernetics*, Vancouver, British Columbia, Canada, pp. 2034-2039.
- EvoNews (1996). *Newsletter of the EvoNet Network of Excellence in Evolutionary Computation*, Special issue: Industrial Applications of Evolutionary Computing. European Commission's ESPRIT IV Programme.
- EvoNews (1998). *Newsletter of the EvoNet Network of Excellence in Evolutionary Computation*, issue 6. European Commission's ESPRIT IV Programme.
- Fardin, J.F., L.V.R. Arruda & W.C. do Amaral (1997). Identificação Relevante para Controle Preditivo utilizando Algoritmo Genético. *3ª Simpósio Brasileiro de Automação Inteligente*, Vitória, ES, Brasil, pp. 295-300.
- Filho, J.L.R., P.C. Treleaven & C. Alippi (1994). Genetic Algorithm Programming Environments. *IEEE Computer*, vol. 27, no. 6, pp. 28-43.
- Fleming, P.J. & C.M. Fonseca (1993). Genetic Algorithms in Control Systems Engineering. *12th Triennial World Congress IFAC*, Sydney, Australia, vol. 2, pp. 383-390.
- Flockton, S.J. & M.S. White (1993). Pole-Zero System Identification using Genetic Algorithms. *Proc. of 5th Int. Conf. on Genetic Algorithms*, S. Forrest (ed), Morgan-Kaufman, San Mateo, CA, USA, pp. 531-535.
- Fogarty, T.C. & L. Bull (1995). Optimising Individual Control Rules and Multiple Communicating Rule-Based Control Systems with Parallel Distributed Genetic Algorithms. *IEE Proc.-Control Theory Appl.*, vol. 142, no. 3, pp. 211-215.
- Fogel, D.B. (1994). An Introduction to Simulated Evolutionary Optimization. *IEEE Trans. on Neural Networks*, vol. 5, no. 1, pp. 3-14.
- Fogel, D.B. (1995). *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, USA.
- Fogel, L.J. (1962). Autonomous Automata. *Industrial Research*, vol. 4, pp. 14-19.
- Fogel, L.J. (1964). *On the Organization of Intellect*, PhD dissertation, University of California, CA, USA.
- Fonseca, C.M. & P.J. Fleming (1995). An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, vol. 3, no. 1, pp. 1-16. The MIT Press.
- Fukuda, T. & T. Shibata (1994). Fuzzy-Neuro-GA Based Intelligent Robotics. In: J.M. Zurada, R.J. Marks II, C.J. Robinson (Ed.), *Computational Intelligence: Imitating Life*, pp. 352-362. IEEE Press, Piscataway, USA.
- Galante, M., M. Cerrolaza & W. Annicchiarico (1993). Optimization of Structural and Finite Element Models via Genetic Algorithms. *World Congress on Optimal Design of Structural Systems*, vol. II, In: J. Herskovits (eds.), Rio de Janeiro, RJ, Brazil, pp. 127-136.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, MA.
- Goldberg, D.E. (1994). Genetic and Evolutionary Algorithms Come of Age. *Communications of the ACM*, vol. 37, no. 3, pp. 113-119.
- Gomide, F.A.C. & R.R. Gudwin (1994). Modelagem, Controle, Sistemas e Lógica Fuzzy. *Controle & Automação*, vol. 4, no. 3, p. 97-115.
- Ghoshray, S., K.K. Yen & J. Andrian (1995). Modified Genetic Algorithms by Efficient Unification with Simulated Annealing. *Int. Conf. Artificial Neural Networks and Genetic Algorithms*, Alés, France, pp. 487-490.
- Gray, G.J., Y. Li, D.J. Murray-Smith & K.C. Sharman (1996). Structural System Identification using Genetic Programming and a Block Diagram Oriented Simulation Tool. *Electronic Letters*, vol. 32, no. 15, pp. 1423-1424.
- Hahnert III, W.F. & P.A.S. Ralston (1995). Analysis of Population Size in the Accuracy and Performance of Genetic Training for Rule-Based Control Systems. *Computers Ops. Research*, vol. 22, no. 1, pp. 65-71.
- Heitkoetter, J. & D. Beasley (1996). The Hitch-Hitker's Guide to Evolutionary Computation: A list of Frequently Asked Questions (FAQ). USENET: comp.ai.genetic. Available via anonymous FTP form rtfm.mit.edu: /pub/usetnet /news.answers/ai-faq/genetic.
- Hoffmann, F. & G. Pfister (1994). Automatic Design of Hierarchical Fuzzy Controllers using Genetic Algorithms. *Proc. of 2nd European Conf. on Intelligent Techniques and Soft Computing*, Aachen, Germany, pp. 1516-1522.
- Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Reprinted in 1992 by MIT Press.
- Howard, L.M. & D.J. D'Angelo (1995). The GA-P: A Genetic Algorithm and Genetic Programming Hybrid, *IEEE Expert*, vol. 10, no. 3, pp. 11-15.
- Huang, R. & T.C. Fogarty (1991). Adaptive Classification and Control-Rule Optimisation via a Learning Algorithm for Controlling Dynamic System. *Proc. of the 30th Conf. on Decision and Control*, Brighton, England, pp. 867-868.
- Hull, A.R. & R.W. Johnson (1994). Performance Enhancement of a Missile Autopilot via Genetic Algorithm Optimization Techniques. *Proc. of the American Control Conf.*, Baltimore, Maryland, USA, pp. 1680-1684.

- Ichikawa, Y. & T. Sawa (1992). Neural Network Application for Direct Feedback Controllers, *IEEE Trans. on Neural Networks*, vol. 3, no. 2, pp. 224-231.
- Inoue, H., K. Kamei & K. Inoue (1997). Automatic Generation of Fuzzy Rules using Hyper Elliptic Cone Membership Function by Genetic Algorithms. *Proc. of 7th Int. Fuzzy Systems Association World Congress*, Prague, Czech Republic, vol. II, pp. 383-388.
- Janikow, C.J. & Z. Michalewicz (1990). A Specialized Genetic Algorithm for Numerical Optimization Problems. *Proc. 2nd Int. IEEE Conf. on Tools for AI*, Herndon, VA, USA, pp. 798-804.
- Karr, C.L. & K. Sharma (1994). An Adaptive Process Control System Based on Fuzzy Logic and Genetic Algorithms, *Proc. of the American Control Conf.*, Baltimore, Maryland, USA, pp. 2470-2474.
- Katayama, R., Y. Kajitani, K. Kuwata & Y. Nishida (1993). Developing Tools and Methods for Applications Incorporating Neuro, Fuzzy and Chaos Technology. *Computer Ind. Engineering*, vol. 24, no. 4, pp. 579-592.
- Kawaji, S., K. Ogasawara & H. Honda (1994). Swing up Control of a Pendulum using Genetic Algorithms. *Proc. of the 33rd Conf. on Decision and Control*, Lake Buena Vista, FL, USA, pp. 3530-3532.
- Kawauchi, Y., M. Inaba & T. Fukuda (1995). Genetic Evolution and Self-Organization of Cellular Robotic System. *JSME Int. Journal - Dynamics, Control, Robotics, Design and Manufacturing*, Series C, vol. 38, no. 3, pp. 501-509.
- Kennedy, S.A. (1993). Five Ways to a Smarter Genetic Algorithm. *AI Expert*, December, pp. 35-38.
- Kim, J.-H., H.-K. Chae, J.-Y. Jeon & S.-W. Lee (1996). Identification and Control of Systems with Friction using Accelerated Evolutionary Programming. *IEEE Control Systems*, vol. 16, no. 4, pp. 38-47.
- Kinrear, K.E. Jr (ed.) (1994). *Advances in Genetic Programming*, MIT Press, Cambridge, MA.
- Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA.
- Koza, J.R. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press, Cambridge, MA.
- Kristinsson, K. & G.A. Dumont (1992). System Identification and Control using Genetic Algorithms. *IEEE Trans. on System, Man, and Cybernetics*, vol. 22, no. 5, pp. 1033-1046.
- Kröhling, R. (1997). Design of PID Controller for Disturbance Rejection: A Genetic Optimization Approach. *Proc. of 2nd Int. Conf. on GA in Eng. Systems: Innovations and Applications*, Glasgow, U.K., pp. 498-503.
- Lee, G.D., B.W. Lee & S.H. Chang (1997). Genetic Programming Model for Long-Term Forecasting of Electric Power Demand, *Electric Power Systems Research*, vol. 40, pp. 17-22.
- Li, C.J. & Y.C. Jeon (1993). Genetic Algorithm in Identifying Non Linear Auto Regressive with Exogenous Input Models for Non Linear Systems. *Proc. of the American Control Conf.*, San Francisco, CA, USA, pp. 2305-2309.
- Li, Y., K.C. Tan & K.C. Ng (1995). Performance Based Linear Control System Design by Genetic Evolution with Simulated Annealing. Technical Report CSC-95017, Department of Electronics and Electrical Engineering, University of Glasgow, U.K.
- Li, Y. & A. Häubler (1996). Artificial evolution of Neural Networks and its Application to Feedback Control. *Artificial Intelligence in Engineering*, vol. 10, no. 20, pp. 143-152.
- Lin, S.-C. & Y.-Y. Chen (1995). A GA-Based Fuzzy Controller with Sliding Mode. *Proc. of the Int. Joint Conf. of the 4th Int. Conf. on Fuzzy Systems and the 2nd Int. Fuzzy Eng. Symposium*, Yokohama, Japan, vol. III, pp. 1103-1110.
- Linkens, D.A. & H.O. Nyongesa (1995). Genetic Algorithms for Fuzzy Control, Part 2: Online System Development and Application. *IEE Proc.-Control Theory Appl.*, vol. 142, no. 3, pp. 177-185.
- Linkens, D.A. & H.O. Nyongesa (1996a). Learning Systems in Intelligent Control: An Appraisal of Fuzzy, Neural and Genetic Algorithm Control Applications. *IEE Proc.-Control Theory*, vol. 143, no. 4, pp. 367-386.
- Linkens, D.A. & H.O. Nyongesa (1996b). A Hierarchical Multivariable Fuzzy Controller for Learning with Genetic Algorithms. *Int. J. Control*, vol. 63, no. 5, pp. 865-883.
- Liu, G.P. & R.J. Patton (1994). Multiobjective Robust Control Design Using Eigenstructure Assignment and Genetic Algorithms. *Proc. of IFAC Symposium on Robust Control Design*, Rio de Janeiro, RJ, Brazil, pp. 162-167.
- Ma, J.T. & Q.H. Wu (1995). Generator Parameter Identification using Evolutionary Programming. *Electrical Power & Energy Systems*, vol. 17, no. 6, pp. 417-423.
- Man, K.F., K.S. Tang & S. Kwong (1996). Genetic Algorithms: Concepts and Applications. *IEEE Trans. on Industrial Electronics*, vol. 43, no. 5, pp. 519-534.
- Magid, Y.L.A. & M.M. Dawoud (1995). Tuning of AGC of Internconnected Reheat Thermal Systems with Genetic Algorithms", *IEEE Conf. on Systems, Man and Cybernetics*, Vancouver, British Columbia, Canada, pp. 2622-2627.
- Marrison, C.I. & R.F. Stengel (1997). Robust Control System Design using Random Search and Genetic Algorithms. *IEEE Trans. on Automatic Control*, vol. 42, no. 6, pp. 835-839.
- McKay, B., M.J. Willis, H.G. Hiden, G.A. Montague & G.W. Barton (1996a). Identification of Industrial Processes Using Genetic Programming. *Int. Conference on Identification in Eng. Systems*, Swansea, U.K.
- McKay, B., M.J. Willis, G.A. Montague & G.W. Barton (1996b). Using Genetic Programming to Develop

- Inferential Estimation Algorithms. *Genetic Programming Conference*, San Francisco, CA, USA.
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin, Germany.
- Nordvik, J.-P. & J.-M. Renders (1991). Genetic Algorithms and their Potential for Use in Process Control: A Case Study. *Proc. of the 4th Int. Conf. on Genetic Algorithms*, R.K. Belew (ed), Morgan-Kaufman, San Mateo, CA, USA, pp. 480-486.
- Odetayo, M.O. & D.R. McGregor (1988). Genetic Algorithm for Inducing Control Rules for a Dynamic System. *Proc. of the 3rd Int. Conf. on Genetic Algorithms*, J.D. Schaffer (ed), Morgan-Kaufman, San Mateo, CA, USA, pp. 177-182.
- Orero, S.O. & M.R. Irving (1996). A Genetic Algorithm For Generator Scheduling in Power Systems. *Electrical Power & Energy Systems*, vol. 18, no. 1, pp. 19-26.
- Patton, R.J. & G.P. Liu (1994). Robust Control Design via Eigenstructure Assignment, Genetic Algorithms and Gradient-based Optimisation. *IEE Proc.-Control Theory Appl.*, vol. 141, no. 3, pp. 202-207.
- Parag, A. & D.R. Lewin (1996). Decentralized Control System Synthesis using a Genetic Algorithm. *13th Triennial World Congress IFAC*, San Francisco, CA, USA, pp.193-198.
- Passino, K.M. (1995). Intelligent Control for Autonomous Systems. *IEEE Spectrum*, June, pp. 55-62.
- Poli, R. & W.B. Langdon (1998). Genetic Programming with One-Point Crossover. *Soft Computing in Engineering Design and Manufacturing*, Chawdhry, P. K., R. Roy & R.K. Pant (eds.), Springer-Verlag, Berlin, Germany, pp. 180-189.
- Porter, B. & A.H. Jones (1992). Genetic Tuning of Digital PID Controllers. *Electronics Letters*, vol.28, no. 9, pp. 843-844.
- Ramos, A.R.M. & D.A.C. Barone (1994). Aplicação de Algoritmo Genético no Controle de Trajetória de um Veículo. *I Simpósio Brasileiro de Redes Neurais*, Caxambu, MG, Brasil, p.147-152.
- Rechenberg, I. (1965). Cybernetic Solution Path of an Experimental Problem, *Library Translation 1122*, Royal Aircraft Establishment, Farnborough, U.K.
- Rock, D. & J. Hirsh (1993). Will GAs Breed with Aerospace?. *AI Expert*, December, pp. 29-34.
- Rudolph, G. (1992). On Correlated Mutations in Evolution Strategies. *Parallel Problem Solving From Nature 2*, R. Männer & B. Manderick (eds), Elsevier Science Publishers, Amsterdam, pp. 105-114.
- Sanz, R., R. Galán, A. Jiménez, F. Matía, J.R. Velasco & G. Martínez (1994). Computational Intelligence in Process Control. *IEEE Int. Conference on Neural Networks*, Orlando, FL, USA, vol. IV, pp. 2203-2208.
- Saravanan, N. (1995). Evolutionary Programming for Synthesis of Optimal Controllers, *Proc. of the 4th Annual Conf. on Evolutionary Programming*, The MIT Press, pp. 645-656.
- Saravanan, N. & D.B. Fogel (1995). Evolving Neural Control Systems. *IEEE Expert*, vol. 10, no. 3, pp. 23-27.
- Schaffer, J.D. (1994). Combinations of Genetic Algorithms with Neural Networks or Fuzzy Systems. In: J.M. Zurada, R.J. Marks II, C.J. Robinson (Ed.), *Computational Intelligence: Imitating Life*, pp. 371-382. IEEE Press, Piscataway, USA.
- Schwefel, H.-P. (1965). *Kybernetische Evolution als Strategie der Experimentellen Forschung in der Strömungstechnik*, Dipl.-Ing. Thesis, Technical University of Berlin, Hermann Föttinger Institute for Hydrodynamics, Germany.
- Schultz, A.C. & J.J. Grefenstette (1990). Improving Tactical Plans with Genetic Algorithms. *Proc. of the 2nd Int. IEEE Conf. on Tools for AI*, Herndon, VA, USA, pp. 328-334.
- Sheblé, G.B., T.T. Maifeld, K. Brittig, G. Fahd & S.F. Coppinger (1996). Unit Commitment by Genetic Algorithm with Penalty Methods and a Comparison of Lagrangian Search and Genetic Algorithm - Economic Dispatch Example. *Electrical Power & Energy Systems*, vol. 18, no. 6, pp. 339-346.
- Soucek, B. & The IRIS Group (eds.) (1992). *Dynamic, Genetic, and Chaotic Programming: the Sixth Generation*. John Wiley & Sons, New York, USA.
- Srinivas, M. & L.M. Patnaik (1994). Genetic Algorithms: A Survey. *IEEE Computer*, vol. 27, no. 6, pp. 17-26.
- Takahashi, R.H.C., P.L.D. Peres & P.A.V. Ferreira (1997). Multiobjective  $H_2/H_\infty$  Guaranteed Cost PID Design. *IEEE Control Systems*, vol. 17, no. 6, pp. 37-47.
- Tan, K.C. & Y. Li (1995).  $L_\infty$  using an Enhanced Genetic Algorithm. Tech. Report CSC-95029, Centre for Systems and Control, Department of Electronics and Electrical Engineering, University of Glasgow, U.K.
- Tan, K.C., Y. Li, D.J. Murray-Smith & K.C. Sharman (1995). System Identification and Linearization using Genetic Algorithms with Simulated Annealing. *Proc. IEE/IEEE Int. Conf. on GA in Eng. Syst.: Innovations and Appl.*, Sheffield, U.K., pp. 164-169.
- Tanomaru, J. (1995). Motivação, Fundamentos e Aplicações de Algoritmos Genéticos. *II Congresso Brasileiro de Redes Neurais*, Curitiba, PR, Brasil, p. 373-403.
- Thompson, A. (1995). Evolving Electronic Robot Controllers that Exploit Hardware Resources. *Proc. of the 3rd European Conf. on Artificial Life*, Springer Verlag.
- Varsek, A., T. Urbancici & B. Filipic (1993). Genetic Algorithms in Controller Design and Tuning. *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 23, no. 5, pp. 1330-1339.
- Variante, S.F. & J.F. Bennaton (1996). Uma Aplicação de Algoritmos Genéticos em Controle de Manipuladores.

*XI Congresso Brasileiro de Automática*, São Paulo, SP, Brasil, p. 531-535.

- Vemuri, V.R. & W. Cedeño (1997). Industrial Applications of Genetic Algorithms. *Int. Journal of Knowledge-Based Intelligent Engineering Systems*, vol. 1, no. 1, pp. 1-12.
- Wolpert, D.H. & W.G. Macready (1997). No Free Lunch Theorems for Optimization, *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, pp. 67-82.
- Yao, X. & Y. Liu (1997). A New Evolutionary System for Evolving Artificial Neural Networks, *IEEE Trans. on Neural Networks*, vol. 8, no. 3, pp. 694-713.
- Zitar, R.A. & M.H. Hassoun (1995). Neurocontrollers Trained with Rules Extracted by a Genetic Assisted Reinforcement Learning System, *IEEE Trans. on Neural Networks*, vol. 6, no. 4, pp. 859-879.
- Zuo, W. (1995). Multivariable Adaptive Control for a Space Station Using Genetic Algorithms. *IEE Proc.-Control Theory Appl.*, vol. 142, no. 2, pp. 81-87.