

Algoritmo Genético Híbrido Aplicado à Otimização de um Controlador com Escalonamento de Ganhos

Luis Henrique Moedinger¹, Leandro dos Santos Coelho²

¹ Siemens Ltda, IC CS II - Infrastructure and Process Support
Rua Pedro Gusso, 2635, CEP 81310-900, Curitiba, PR, Brasil
E-mail: luishenr@siemens.com.br

² Pontifícia Universidade Católica do Paraná

Grupo Produtrônica, Programa de Pós-Graduação em Engenharia de Produção e Sistemas
Laboratório de Automação e Sistemas, LAS/PPGEPS/CCET/PUCPR
Rua Imaculada Conceição, 1155, CEP 80215-901, Curitiba, PR, Brasil
E-mails: lscoelho@rla01.pucpr.br

Abstract

This paper presents the optimization of a gain scheduling controller based on Lamarckian evolution. The evolutionary theory advocated by Lamarck focuses on the inheritance of characteristics acquired for self-adaptation to environment. In the domain of the purpose of acquiring adaptive strategies, it is important to make use of the information of experiences through adaptation. The Lamarckian evolution approach presented in this paper combines a local search algorithm (simulated annealing) with genetic algorithms (global search). Simulation results deal the control of a continuous stirred tank reactor, that presents open-loop unstable dynamic and nonlinear behaviors. The simulation results show that the application of the Lamarckian evolution strategy effectively improve the parameters optimization of gain scheduling controller.

1. Introdução

Os métodos de otimização e busca estocástica são baseados nos princípios e modelos da evolução biológica natural e têm recebido crescente interesse nas últimas décadas, devido, principalmente, a sua versatilidade para a resolução de problemas complexos, nas áreas de otimização e aprendizado de máquina [1], [2]. Nos últimos anos diversas pesquisas têm sido relatadas quanto ao desenvolvimento de métodos de otimização de controladores usando algoritmos evolutivos, principalmente os algoritmos genéticos [3]-[5].

Entretanto, apesar dos algoritmos evolutivos caracterizarem-se por serem eficientes em buscas globais, apresentam dificuldades na realização de buscas locais. Uma abordagem relevante para aplicações de otimização são concepções de evolução Lamarckiana [6], [7], que apesar das controvérsias quanto ao significado da palavra *meme* também é denominada *algoritmo memético* por alguns autores [8], [9].

Para obter-se os benefícios da evolução Lamarckiana é necessário executar, inicialmente, um algoritmo evolutivo para localizar a região de ótimo global (fase de evolução), e após aplicar-se outra metodologia de otimização para a busca local (fase de aprendizado).

A vantagem da utilização de um método de busca direto (por exemplo, método simplex de Nelder-Mead, método de Hooke-Jeeves, *simulated annealing*) em relação à busca local está na possibilidade de melhoria da velocidade de convergência somente pela avaliação da função objetivo e sem a necessidade dos cálculos de derivadas. O valor final obtido pelo método de busca direto provavelmente será mais preciso, e conseqüentemente mais adequado que o obtido pelo algoritmo evolutivo, quando este é aplicado isoladamente.

O problema abordado, neste artigo, visa a aplicação de uma concepção de evolução Lamarckiana que combina um algoritmo genético com a técnica *simulated annealing* na otimização de um controlador *PID* (proporcional, integral e derivativo) com escalonamento de ganhos. O controlador foi otimizado visando um desempenho apropriado para o controle de um reator químico que apresenta características complexas, tais como não-lineares e ganho estático variável.

As seções do artigo apresentadas a seguir são organizadas da seguinte forma. Na seção 2, o algoritmo genético, o *simulated annealing* e a configuração de evolução Lamarckiana proposta para a otimização do controlador são descritos. Na seção 3, as equações que regem o controlador *PID* com escalonamento de ganhos são apresentadas. Na seção 4, a descrição e equacionamento do reator químico e os resultados de simulação são abordados. A conclusão e as perspectivas de trabalhos futuros são discutidas na seção 5.

2. Metodologia de otimização

2.1. Algoritmo genético

Um algoritmo genético é um procedimento de otimização estocástico, onde o valor da função de adequabilidade ou aptidão (*fitness*) obtido é oriundo dos melhores elementos das gerações anteriores, nas quais os atributos de um indivíduo mais apto tende a ser transferido para as gerações futuras.

Uma vantagem na utilização dos algoritmos genéticos é por eles não utilizarem cálculos de derivadas, isto facilita a sua utilização em problemas não-convexos e com descontinuidades [10].

2.1.1 Representação das soluções e cromossomo

Um dos principais componentes dos algoritmos genéticos são os cromossomos. O cromossomo contém os valores que a serem avaliados em relação a uma função de adequabilidade e que são submetidos às operações de seleção (acasalamento e reprodução), cruzamento e mutação.

Os cromossomos (*strings*) podem ser representados em qualquer uma das bases numéricas, para este projeto a base para representação escolhida foi a binária (ou canônica). Devido à escolha da base binária, cada *bit* da população é denominado de *gene*.

O conjunto de cromossomos forma a população a ser avaliada. A população é submetida ao seguinte ciclo evolutivo: (i) é avaliada a função de adequabilidade para verificar-se o quão “boa” é cada solução para o problema, e (ii) em seguida são realizadas as operações de seleção, cruzamento e mutação, que são descritas a seguir.

Para este problema foram determinados 30 cromossomos de 16 *bits* para compor a população de soluções para o problema. Para este projeto foi configurado como critério de parada de 30 gerações (iterações).

2.1.2 Função de adequabilidade

A função de adequabilidade é utilizada para a determinação de “quão bom” é o resultado obtido. Cada um dos cromossomos é submetido a avaliação da função de adequabilidade, quando todos os membros da população forem avaliados pode-se comparar qual deles é o que possui o melhor resultado.

Entre uma geração e outra, a informação do melhor membro da população é utilizada para aumentar a probabilidade da geração seguinte possuir cromossomos “similares” com o melhor cromossomo da população atual (estratégia de seleção elitista).

Quando a população estiver na última geração (de acordo com a adoção de um critério de parada) que se deseja avaliar, o membro da população que possuir a

melhor função de adequabilidade é escolhido como solução para o problema.

2.1.5. Operador de cruzamento (*crossover*)

Analogamente ao que ocorre na natureza, a operação de cruzamento é a responsável pela troca de genes (combinação) entre os cromossomos, através desta troca são gerados novos membros da população.

O ponto de corte do cruzamento é determinado de forma aleatória com distribuição uniforme. O cruzamento pode ocorrer ou não, a ocorrência está condicionada a p_c (probabilidade de cruzamento). Neste estudo foi utilizado um valor de $p_c = 0,80$ (80%).

2.1.4. Operador de mutação

A operação de mutação é a responsável pela troca aleatória dos genes do cromossomo. Esta alteração aleatória pode evitar que o algoritmo genético fique sujeito a mínimos locais, porém se a probabilidade de mutação for elevada, o algoritmo pode passar a se comportar de forma aleatória. Esta característica de aleatoriedade é indesejada, pois dificulta a convergência do algoritmo genético para uma solução apropriada. Para este problema foi utilizado o valor de 0,05 (5%) para a p_m (probabilidade de mutação).

2.1.5. Operador de seleção

A operação de seleção objetiva a reprodução dos indivíduos mais aptos da população, estes obtidos pelas operações de cruzamento e mutação.

2.1.6. Etapas de aplicação do algoritmo genético

Após a descrição detalhada nas seções anteriores dos operadores presentes nos algoritmos genéticos, pode-se sintetizar as etapas à aplicação desta metodologia de otimização nos seguintes passos:

- (i) gerar a população inicial de parâmetros compreendendo N_{ind} indivíduos (soluções para o problema). Cada uma das soluções consiste de vetores $x_i \in \{0,1\}$ (representação canônica). Estes parâmetros são iniciados aleatoriamente, de acordo com uma distribuição uniforme;
- (ii) classificar cada solução x_i , $i=1, \dots, N_{ind}$, com relação ao cálculo da função de aptidão, 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 uma estratégia de seleção (usada a roleta neste trabalho);
- (iv) aplicar o operador genético de cruzamento;
- (v) aplicar o operador genético de mutação;
- (vi) gerar uma nova população; e
- (vii) repetir as etapas (ii) a (vi) até que um critério de parada seja satisfeito.

2.2. Simulated annealing

O algoritmo *simulated annealing* (SA) ou t mpera simulada [11]   uma varia o de algoritmos de subida de encosta, onde o objetivo   a minimiza o de uma fun o objetivo (n vel de energia).

O SA baseia-se em uma analogia com a mec nica estat stica de materiais com resfriamento, onde subst ncias f sicas tais como os metais s o levados a altos n veis de temperatura e posteriormente s o gradualmente resfriadas at  alcan ar um estado m nimo de energia. Sob outras condi es, menos cuidadosas de resfriamento, o material se cristalizaria com uma energia “localmente m nima”, o que freq entemente se traduz em imperfei es estruturais. A esse processo cuidadoso de resfriamento denomina-se *annealing*.

A abordagem do SA   probabil stica. O SA n o requer informa o de derivadas e n o   afetado por descontinuidades e n o-linearidades. O SA   um procedimento de descida de encosta, mas configurado de forma modificada. O SA realiza pequenos passos de busca em uma topografia local; se o passo resulta em uma melhora na solu o, a nova solu o   aceita, caso contr rio, esta nova solu o   aceita com probabilidade que inicialmente   configurada para 1. Entretanto, com o progresso das itera es o SA   reduzida a probabilidade de aceitar uma nova solu o que n o apresenta aprimoramento em rela o  quela obtida na itera o anterior.

A id ia do SA origina-se numa combina o das observa es sobre a f sica dos materiais, com procedimento computacional tendo por finalidade a simula o do comportamento de uma cole o de  tomos (vari veis do problema) em condi es de temperatura fixa.

2.3. Evolu o Lamarckiana

Os m todos de otimiza o t m duas formas de configura o: os m todos determin sticos e os m todos estoc sticos. As t cnicas determin sticas buscam um ponto de m nimo, baseado na informa o dada pelo negativo do gradiente da fun o objetivo. A efici ncia destas t cnicas depende de diversos fatores, tais como: a solu o inicial, a precis o da avalia o da dire o de busca e o m todo utilizado para executar a busca local, al m de um crit rio de parada apropriado para o problema em quest o [12].

Nos m todos determin sticos, a solu o obtida pode ser um ponto de m nimo local, que pode ser m nimo global se a fun o apresentar apenas uma moda. Neste contexto, entre as desvantagens dos m todos determin sticos destacam-se a necessidade de avalia es do gradiente e a falta da garantia do m nimo global quando o problema tem uma superf cie de busca multimodal.

Os m todos estoc sticos, dos quais os algoritmos gen ticos fazem parte, n o necessitam do c lculo do gradiente e s o aptos a encontrar regi es promissoras quanto a obten o da solu o global. Contudo, o n mero de avalia es da fun o objetivo (complexidade computacional), necess rias para encontrar a solu o,   normalmente maior que o n mero requerido pelos m todos determin sticos.

A configura o de abordagens compostas por t cnicas determin sticas, estas hibridizadas [13], com t cnicas estoc sticas,   uma alternativa relevante. A concep o h brida de otimiza o adotada, neste artigo,   tamb m denominada de algoritmo gen tico h brido, algoritmo gen tico com busca local ou otimiza o baseada em evolu o Lamarckiana.

A sele o natural n o   uma parte da teoria de Lamarck. A id ia central da abordagem de Lamarck (quando contrastada com a abordagem Darwiniana)   que a evolu o pode ocorrer por meio do indiv duo adaptado ao seu ambiente e passando estas adapta es para seus descendentes.

A evolu o de Lamarck n o   encontrada na natureza, pois se desconhece um mecanismo em que a experi ncia de um indiv duo possa modificar seu gen tipo [14]-[16]. Entretanto, a evolu o proposta por Lamarck   totalmente poss vel para a implementa o de sistemas baseado nas id ias de Lamarck em *software* [6].

Neste contexto, para obter os benef cios da configura o de evolu o Lamarckiana, uma forma eficiente   executar, inicialmente, um algoritmo gen tico para localizar a regi o de “ timo” global e ap s aplicar-se outra metodologia de otimiza o para a busca local, ou seja, o algoritmo de busca local pode avaliar alguns cromossomos (ou indiv duos) na regi o pr xima a estes e substituir o cromossomo original do algoritmo gen tico. A busca local sob este ponto de vista corresponde a experi ncia do organismo.

A vantagem da utiliza o de um m todo de busca local, usualmente um m todo direto, em rela o a busca na vizinhan a de um ponto promissor de busca est  na melhoria da velocidade de converg ncia do algoritmo de otimiza o como um todo. O valor final obtido pelo m todo de busca direto em m dia tende a ser mais preciso que o obtido pelo algoritmo gen tico atuando isoladamente.

3. Controlador PID com escalonamento de ganhos

O objetivo da evolu o Lamarckiana   a otimiza o dos tr s par metros (K_p , T_i e T_d). A forma padr o do controle PID digital cl ssico   dada pela equa o a diferen as

$$u(t) = u(t-1) + q_0 e(t) + q_1 e(t-1) + q_2 e(t-2), \quad (1)$$

com valores de

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

$$q_1 = -K_p \left(1 + \frac{2T_d}{T_s} - \frac{T_s}{2T_i} \right), \quad (3)$$

$$q_2 = K_p \frac{T_d}{T_s}, \quad (4)$$

onde o sinal de controle $u(t) \in [-5, 0; 5, 0]$, T_s é o período de amostragem, $e(t)$ é o erro dado pela diferença entre a saída e a referência desejada e $y(t)$ é a saída do processo.

O projeto do controle *PID* com escalonamento de ganhos diferencia-se do *PID* convencional, pois é mais adequado ao tratamento de processos com dinâmicas diferenciadas para cada faixa de operação, ou seja, processo regidos por múltiplos modelos.

O projeto do *PID* com escalonamento de ganhos consiste na sintonia dos três parâmetros do controlador *PID*, sendo um para cada mudança de referência, y_r , totalizando 9 parâmetros, pois o procedimento de otimização prevê 3 mudanças de referência, uma em cada região de comportamento dinâmico diferente do reator químico.

O funcional adotada e a ser minimizado é dado pela equação

$$J(u, e) = \frac{\sum_{k=1}^N i^* |e(k)|}{N}, \quad (5)$$

onde N é o número de amostras do processo, w é fixado em 1, i é igual k , exceto quando ocorre alguma mudança de referência, pois neste caso, $i=1$ e após passa a ser incrementado sucessivamente a cada iteração da mesma forma que k .

Conseqüentemente, a função de adequabilidade, $F[J(e, u)]$, a ser maximizada pelo algoritmo de otimização é dada por

$$f[J(e, u)] = \frac{1}{1 + J(e, u)} \quad (6)$$

Neste caso, quanto maior o valor de $f[J(e, u)]$, melhor é o membro da população. A figura 1 apresenta a configuração utilizada para otimização do controlador *PID* com escalonamento de ganhos.

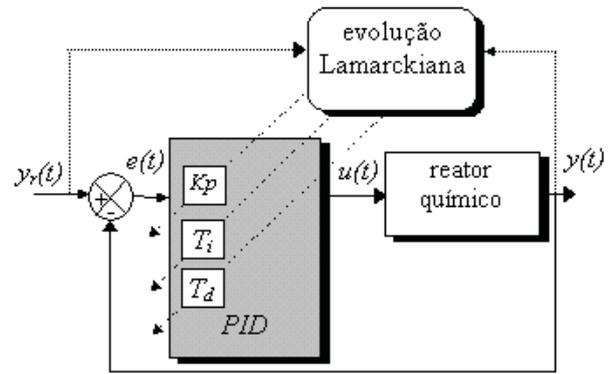


Figura 1. Controle *PID* com escalonamento de ganhos usando otimização por algoritmo de evolução Lamarckiana.

4. Descrição do estudo de caso e resultados de simulação

4.1. Processo reator químico do tipo *CSTR* (*continuous stirred tank reactor*)

O estudo de caso consiste de um reator químico do tipo *CSTR*, onde as equações dinâmicas do reator, que representam um comportamento dinâmico complexo, dependendo da região de operação do processo, são dadas por [17]:

$$\frac{dx_1}{dt} = -x_1 + D_a (1 - x_1) e^{\frac{x_2}{1+x_2/\varphi}}, \quad e \quad (7)$$

$$\frac{dx_2}{dt} = -(1 - \beta)x_2 + BD_a (1 - x_1) e^{\frac{x_2}{1+x_2/\varphi}} + \beta u, \quad (8)$$

onde x_1 e x_2 representam a concentração dos reagentes (adimensional) e temperatura do reator, respectivamente. A entrada de controle, u , é a temperatura da camisa de esfriamento. Os parâmetros físicos nas equações do modelo do reator são D_a , φ , B e β que correspondem ao número de Damköhler, a energia de ativação, calor da reação e coeficiente de transferência de calor, respectivamente. Baseado nos parâmetros nominais do sistema de $D_a = 0,072$, $\varphi = 20$, $B = 8$ e $\beta = 0,3$, o reator exibe um comportamento instável em malha aberta.

A simulação do comportamento dinâmico do reator é realizada pela conversão das equações (7) e (8) para um sistema de equações discretas, pelo uso do método de Euler [18]. Por conseguinte as equações se transformam em

$$x_1(t+1) = x_1(t) + T_s * \left[-x_1(t) + D_a(1-x_1(t))e^{\frac{x_2(t)}{1+x_2(t)/\varphi}} \right], \quad (9)$$

$$x_2(t+1) = x_2(t) + T_s * \left[-(1-\beta)x_2(t) + BD_a(1-x_1(t))e^{\frac{x_2(t)}{1+x_2(t)/\varphi}} + \beta u(t) \right], \quad (10)$$

$$y(t) = x_2(t), \quad (11)$$

onde T_s é o período de amostragem (adotado $T_s = 0,2$ s). A relação entre a saída do reator, y , e a entrada do sinal de controle, u , pode ser obtida pela substituição da equação (10) na equação (11) com $t = t + 1$:

$$y(t+1) = x_2(t) + T_s * \left[-(1-\beta)x_2(t) + BD_a(1-x_1(t))e^{\frac{x_2(t)}{1+x_2(t)/\varphi}} + \beta u(t) \right] \quad (12)$$

A figura 1 mostra a curva estática $u(t)$ versus $y(t)$ (saída em regime permanente) do CSTR. Devido à curva estática o reator químico, que é a relação entre a entrada e a saída em regime permanente, possui três regiões bem definidas, optou-se pela utilização de um controlador PID com escalonamento de ganhos [19].

A configuração adotada para o controle PID possui as 3 regiões de operação definidas pelo sinal da entrada. Como pode ser visto na figura 2, a primeira região está definida entre 0,0 V e 1,5 V, a segunda entre 1,5 V e 5,5 V e a última entre 5,5 V e 7,0 V.

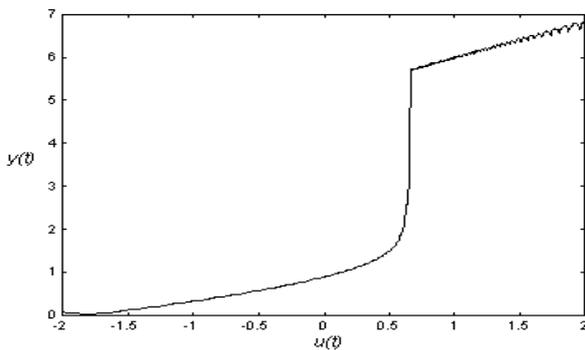


Figura 2. Curva estática do CSTR.

4.2. Resultados de simulação e análise de desempenho

Após a escolha do PID a ser utilizado, foi implementado um método de otimização baseado em evolução Lamarckiana. Para esta implementação foi

utilizada uma política de seleção elitista, ou seja, o melhor membro de uma geração foi re-inserido na geração seguinte.

A figura 3 apresenta a resposta para o sistema utilizando os valores otimizados utilizando um algoritmo de evolução Lamarckiana. Os valores obtidos por otimização para o controlador PID são apresentados na tabela 1.

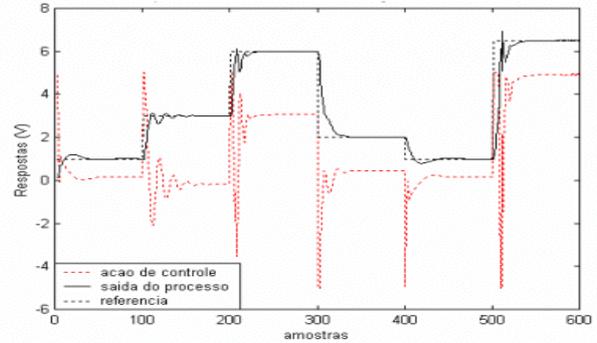


Figura 3. Simulação do PID para controle do CSTR usando otimização por evolução Lamarckiana.

Tabela 1: Parâmetros do PID otimizados pelo algoritmo de evolução Lamarckiana.

ganho proporcional	tempo integral	tempo derivativo
K_{p1}	T_{i1}	T_{d1}
1,6550	1,2384	0,4375
K_{p2}	T_{i2}	T_{d2}
5,1774	1,7504	0,0029
K_{p3}	T_{i3}	T_{d3}
4,8735	1,0779	0,0256

A figura 4 mostra os valores do fitness para cada geração da população. Pode-se observar que o melhor membro da população é reinserido na geração seguinte. Nota-se também que a população de uma forma geral vai convergindo, ou seja, tem o seu valor médio de fitness melhorado com o passar das gerações.

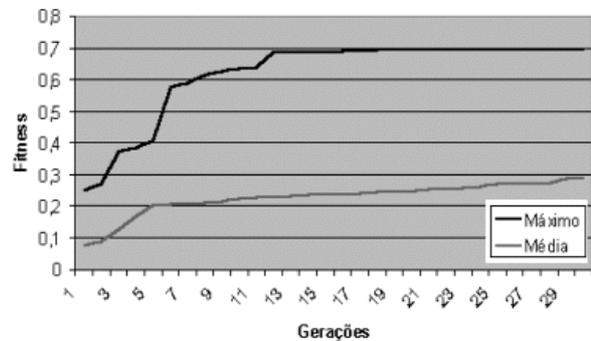


Figura 4: Convergência da população do algoritmo de evolução Lamarckiana.

5. Conclusão

Os algoritmos genéticos apresentam vantagens e desvantagens em relação aos métodos tradicionais e este aspecto serve para enfatizar a necessidade de não abandonar os métodos convencionais de otimização. Porém, o projetista pode utilizar algoritmos genéticos como uma alternativa no auxílio a resolução de problemas onde os métodos convencionais podem falhar ou não terem desempenho adequado.

Os algoritmos genéticos convencionais geralmente são eficientes para a busca global, mas são relativamente lentos na realização de buscas locais. Esta característica conduz à possíveis formas de integração com métodos numéricos de descida de encosta. Neste artigo foi utilizado o algoritmo *simulated annealing*.

A vantagem da abordagem de evolução Lamarckiana adotada é a possibilidade de configurar-se algoritmos de otimização que permitam tanto a busca global quanto local. A utilização do algoritmo *simulated annealing* associado ao algoritmo genético foi realizada com sucesso e os resultados foram promissores.

Entre as desvantagens da abordagem de evolução Lamarckiana adotada é o maior custo computacional em termos de memória e velocidade de processamento do algoritmo híbrido em relação ao algoritmo genético. Além disso, no projeto do algoritmo de evolução Lamarckiana deve-se ter cuidado para que a abordagem escolhida não reduza a diversidade da população (abuso da adoção de estratégias elitistas) do algoritmo genético rapidamente e apresente desta forma uma estagnação quanto a realização de buscas globais.

Referências

- [1] T. Bäck, D. B. Fogel and Z. Michalewicz. *Handbook of evolutionary computation*. Bristol: Institute of Physics Pub., 1997.
- [2] K. F. Man, K. S. Tang and S. Kwong. Genetic algorithms: concepts and applications. *IEEE Transactions on Industrial Electronics*, 43(5): 519-534, 1996.
- [3] M. Jamshidi, R. A. Krohling, L. S., Coelho and P. Fleming. *Robust control with genetic algorithms*, Boca Raton, CRC Press, FL, 2002.
- [4] P. J. Fleming and C. M. Fonseca. Genetic Algorithms in control systems engineering. *Proceedings of 12th IFAC World Congress*, Sydney, Australia. vol. 2, pages 383-390, 1993.
- [5] J. T. Alander. An indexed bibliography of genetic algorithms in control. *Report 94-1*, Department of Information Technology and Production Economics, University of Vaasa, Finland, 1995.
- [6] C. Wellock and B. J. Ross. An examination of Lamarckian genetic algorithms, *Genetic and Evolutionary Computation Conference Late Breaking Papers*, San Francisco, CA, USA, pages 474-481, 2001.
- [7] D. Whitley, V. S. Gordon and K. Mathias. Lamarckian evolution, the Baldwin effect and function optimization, *Parallel Problem Solving from Nature*, Eds. Y. Davidor, H. P. Schwefel and R. Manner, Springer-Verlag, pages 6-15, 1994.
- [8] P. Moscato and M. G. Norman. A 'memetic' approach for the travelling salesman problem — implementation of a computational ecology for combinatorial optimisation on message-passing systems, *Proceedings of the International Conference on Parallel Computing and Transputer Applications*, IOS Press, Amsterdam, 1992.
- [9] P. Merz and B. Freisleben. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem, *IEEE Transactions on Evolutionary Computation*, 4(4): 337-352, 2000.
- [10] M. Srinivas and L. M. Patnaik. Genetic algorithms: a survey. *IEEE Computer*, June, pp. 17-26, 1994.
- [11] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220: 45-54, 1983.
- [12] J. J. Grefenstette. Lamarckian learning in multi-agent environments. *Proceedings of the 4th International Conference on Genetic Algorithms*, San Diego, Morgan Kaufmann, July 1991.
- [13] J.-M. Renders and S. P. Flasse. Hybrid methods using genetic algorithms for global optimization. *IEEE Transactions on Systems, Man, and Cybernetics — Part B: Cybernetics*, 26(2): 243-258, 1996.
- [14] B. A. Julstrom. Comparing Darwinian, Baldwinian, and Lamarckian search in a genetic algorithm for the 4-cycle problem. *Genetic and Evolutionary Computation Conference, Late Breaking Papers*, Orlando, FL, USA, pages 134-138.
- [15] D. H. Ackley and M. L. Littman. A case for Lamarckian evolution. *Artificial Life III*, C. G. Langton, SFI Studies in the Sciences of Complexity, Proc. Vol. XVII, Addison-Wesley, pages 3-10.
- [16] K. W. C. Ku, M. W. Mak and W. -C. Siu. A study of the Lamarckian evolution of recurrent neural networks. *IEEE Transactions on Evolutionary Computation*, 4(1): 31-41, 2000.
- [17] C. -T. Chen and S. -T. Peng. A nonlinear control strategy based on using a shape tunable neural controller. *Journal of Chemical Engineering of Japan*, 30(4): 637-646, 1997.
- [18] K. J. Åström and B. Wittenmark. *Computer controlled systems: theory and design*, New Jersey: Prentice Hall, 1994.
- [19] K. J. Åström and T. Hägglund. Practical experiences of adaptive techniques. *Proceedings of American Control Conference*, San Diego, CA, 1990.