

UMA PROPOSTA DE USO DE TECNOLOGIA ADAPTATIVA PARA SIMULAÇÃO DE REDES NEURAIIS EM UM DISPOSITIVO COMPUTACIONAL

RICARDO LUIS DE AZEVEDO DA ROCHA

Curso de Ciência da Computação
Faculdade de Informática (FCI), e Universidade São Marcos
Av. Humberto de Alencar Castelo Branco, n.3972 Bairro Assunção
ZIP CODE 09850-901, São Bernardo do Campo/SP – Brasil
e-mail: Ricardo-Rocha@acm.org

RESUMO

Este artigo descreve parte dos resultados de uma pesquisa cujo objetivo foi propor uma formulação para um método automatizado de escolha de soluções de problemas, para uso como um dispositivo computacional. Como parte do estudo buscou-se uma formulação que mantivesse características de aprendizado e paralelismo, inerentes às redes neurais. Optou-se por simular a características de uma rede neural dentro do dispositivo.

Os modelos gerados são adaptativos, isto é, o substrato formal utilizado como base na proposição do método e do dispositivo é o autômato adaptativo [1].

Palavras-chave: Modelagem, Adaptação, Autômatos, Redes Neurais.

1. INTRODUÇÃO

A proposta que originou este artigo veio da pesquisa [6], cujo objetivo é o desenvolvimento de um método de construção de modelos e resolução de problemas complexos utilizando um formalismo adaptativo [1] como base [16]. O objetivo principal é encontrar soluções para problemas computacionais, e sendo mais preciso, para aqueles problemas computacionais que possam ser transformados em problemas de linguagem (Esta escolha deve-se à necessidade de limitar a abrangência das informações recebidas e tratadas pelo dispositivo gerado). Para isso, estudam-se comparativamente alguns dos métodos e técnicas de construção e resolução de modelos correntemente adotados, e, utiliza-se características destes no intuito de propor um método alternativo, preservando as características estudadas e consideradas importantes.

Dentre os métodos e modelos estudados estão presentes as redes neurais. Considerou-se, nesta pesquisa, de grande importância a capacidade de treinamento e o grau de paralelismo inerente às redes neurais assim. Optou-se por utilizar estas características no dispositivo proposto.

Outro objetivo é propor um mecanismo de busca de soluções, que seja adequada à metodologia proposta. Este mecanismo deve ser simulado em um computador, isto é, deve ser um mecanismo de software. Um protótipo para o mecanismo é construído e testado posteriormente, de modo a validar as hipóteses da proposta. Assim, o dispositivo BSMA (Busca de Soluções por Máquina Adaptável) foi proposto, formalizado, e um protótipo experimental foi construído e exercitado, para que se completasse esta pesquisa. Maiores detalhes podem ser encontrados em [6].

Este artigo descreve sucintamente o método e o dispositivo propostos em [6], e propõe uma forma de simular redes neurais através do dispositivo BSMA. A forma proposta foi utilizada também em [6], mas acredita-se que este possa ser um ponto de partida para novas pesquisas usando os resultados encontrados.

2. MÉTODO PROPOSTO

O método de investigação de soluções é proposto a partir das características encontradas no modelo do dispositivo computacional BSMA definido, que, conforme mencionado, utiliza como base formal o autômato adaptativo, acrescido de características de outros modelos levantados, procurando-se dessa forma garantir a integridade, a integração e a uniformidade do método.

2.1. Definição do Método e Construção do Dispositivo BSMA

O método proposto é baseado em indução, da seguinte forma: Para um dado problema ou questão Q a ser solucionado, deve-se coletar o maior número possível de informações, como por exemplo, teorias, modelos, exemplos de resultados. A partir destas informações, organiza-se o material colhido segundo a sua natureza, e trabalha-se com uma ordenação deste material operando teorias juntamente com modelos, como se fossem hipóteses em um processo indutivo, e com os exemplos, como se fossem dados observados. A partir daí, tem-se um esquema adequado à utilização da teoria da previsão de Solomonoff, podendo-se aplicar a regra de Bayes, substituindo a probabilidade *a priori* por uma medida universal, como a complexidade de Kolmogorov para prefixos. Então, a resposta à questão é fornecida como uma previsão, se houver resposta alcançável pelo dispositivo utilizado [3].

A questão da resposta a ser gerada, quando não há qualquer informação sobre o problema, é resolvida através do uso do senso comum. Assim, busca-se uma resposta que seja aceita pela maioria dos indivíduos e, depois, verifica-se se tal resposta está correta. A forma de produzi-la através do senso comum foge ao escopo deste trabalho.

O método a ser utilizado aqui será limitado em seu aspecto de produzir solução para problemas novos, ou seja, somente se utiliza o método quando há alguma informação sobre o fenômeno ou o problema em

estudo. A utilização do dispositivo BSMA, por um pesquisador, deve ser então realizada através do envio de modelos ou teorias, e de dados de exemplo que possam servir para treinar o dispositivo na busca de uma solução para o problema proposto. A execução do treinamento pode ser feita repetidas vezes, para a mesma massa de dados, já que o dispositivo tem limite de tempo de execução, não possui senso comum e nem consciência, portanto, pode não haver resposta até mesmo durante o treinamento.

A operação do dispositivo dá-se por meio de ciclos. Assim, um ciclo corresponde a uma execução completa do modelo, ou seja, é composto pelos passos de computação executados pelos modelos utilizados. Por outro lado, é necessário impor limites de tempo de execução dentro de um ciclo, bem como um limite para o número de ciclos. Isto é garantido pela existência de um dispositivo de medida de tempo (relógio), que controla todos os tempos de execução.

2.2. Dispositivo BSMA Proposto

É fundamental, para o propósito de encontrar soluções, que o dispositivo possua um indicador que permita identificar se a trajetória computacional adotada é adequada, isto é, se ela conduz a uma solução. Para a especificação do indicador, há três alternativas viáveis: Treinamento, Aprendizagem por exemplos e Especificação de uma função de verificação.

A alternativa escolhida foi a primeira. Com isso, para cada par do conjunto (entrada, saída) fornecido, modelos de autômato são gerados e são exercitados, tendo seus comportamentos comparados com o comportamento esperado para cada par. Assim, modelos que apresentem comportamentos diferentes do esperado são descartados. Cada processo que consiga atingir a resposta esperada é utilizado para compor novos processos de solução.

Assim, o elemento de entrada do dispositivo deve especificar uma construção inicial, associado a um conjunto de triplas estipuladas pelo usuário do dispositivo: $\{(e_i, s_i, v_i) \mid e_i - \text{entrada}, s_i - \text{saída}, v_i - \text{valor}, i \geq 1\}^n, n \geq 1$.

O indicador para a função de medição opera com um conjunto de dados para o qual se deseja encontrar uma hipótese consistente e com o conjunto de modelos de autômato, que representa o conjunto de hipóteses levantadas. Esta função de medição determina, na realidade, se os modelos de solução estão trilhando um caminho que leva ao objetivo a ser alcançado, desempenhando um papel similar à manutenção de meta.

Na Figura 1, o elemento controlador aparece como o item mais importante, já que ele centraliza todas as ações do dispositivo, disparando o exercício dos modelos de solução através da entrada e identificação dos problemas realizada pelo elemento de entrada. Assim, o controlador induz alterações nas construções, de forma a tentar gerar novos modelos. Caso não se chegue a uma solução, pode ocorrer a necessidade de processamento cíclico e, mesmo assim, pode-se não alcançar uma solução. Neste caso a resposta assumida é “Solução não-Exequível”, por inexistência ou impossibilidade de aplicação prática.

As construções são representadas na figura através dos modelos gerados ($M_1 .. M_n$), e juntas compõem um agregado, isto é, um vetor de soluções possíveis, porém, não necessariamente, preenchido por completo. Os agregados, compostos pelas construções ($M_1 .. M_n$), são independentes dos demais, podendo com isso gerar diferentes soluções para os problemas. A quantidade de agregados existentes no dispositivo depende basicamente de seu parâmetro de limite da quantidade de espaço alocado, e também da quantidade de diferentes modelos de solução gerados. A combinação de ambos indica a quantidade total de agregados, embora o limite máximo seja estabelecido pelo parâmetro de limite da quantidade de espaço.

Pode-se observar que cada construção dentro de um agregado gera saída, que não passam diretamente ao elemento controlador. A Figura 1 mostra as saídas sendo tratadas por um elemento “CP” (comparador). Este elemento simboliza, na realidade, que o controlador realiza uma comparação entre as construções exercitadas que permaneceram, e que permitem gerar os valores das medidas de complexidade utilizadas e

não que exista um elemento específico denominado comparador. Maiores detalhes em [6].

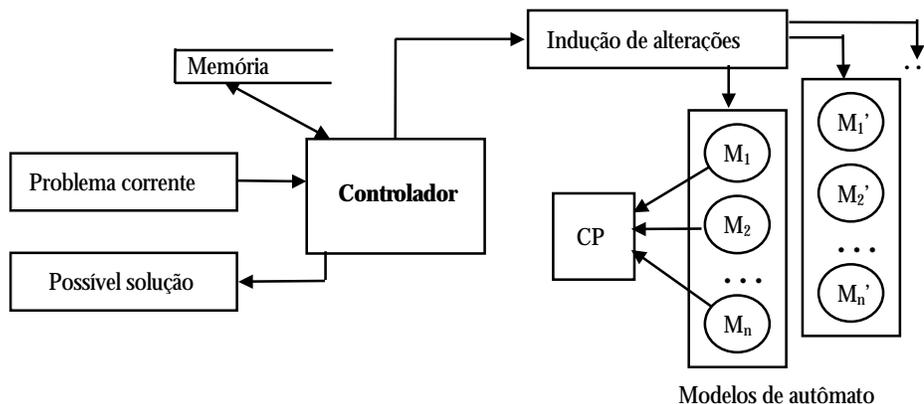


Figura 1 - Construção do dispositivo BSMA proposto, utilizando um modelo adaptável

3. Simulação de Redes Neurais no dispositivo BSMA

Uma das possibilidades é utilizar-se um modelo de aprendizado baseado em redes de perceptrons simples, em que cada neurônio possa ser representado por um modelo de autômato adaptativo. Essas redes devem possuir um mecanismo de controle sobre os tipos de operações e as capacidades de cada neurônio.

Outra possibilidade é utilizar um modelo de aprendizado baseado em rede neural multicamadas. Essas redes devem possuir um mecanismo de controle sobre os tipos de operações, bem como uma forma de construí-las e controlá-las adequadamente, de maneira a atingir um determinado objetivo previamente especificado.

Optou-se por utilizar algumas características de um dos modelos de redes. O modelo de redes de Hopfield é então utilizado como motivação para a proposta do método e do dispositivo BSMA. A idéia é utilizar modelos de autômato diferentes operando ao mesmo tempo.

3.1. Modelo de Autômatos como uma Rede Neural

Pode-se utilizar um formalismo adaptável como base para a construção de um modelo de neurônio. Conforme observado anteriormente, um formalismo adaptável pode explorar as alternativas de execução existentes dentro de seu modelo, realizando as tarefas de maneira similar à execução de vários modelos diferentes. Com isso, pode-se explorar o potencial do autômato adaptativo (automodificação) para modelar os nós de uma rede neural de Hopfield.

A construção de um modelo de rede neural através do uso do autômato adaptativo exige uma função de saída, para que cada nó possa produzir uma resposta às entradas recebidas. Ao se construir um modelo de rede neural, que é essencialmente um modelo de computação paralela, pode-se distribuir os nós (modelos de autômato) por processadores de autômato, que irão realizar as tarefas de cada nó. Um processador de autômato é uma máquina seqüencial capaz de executar um modelo de autômato adaptativo. Em caso de haver uma máquina paralela, distribui-se entre os processadores reais existentes os processadores de autômato necessários. A quantidade de processadores de modelos de autômato necessários à realização da tarefa global deve ser determinada através da quantidade de nós da rede, e depois distribuída pelos processadores reais existentes. As entradas necessárias ao processamento de cada nó são enviadas por um processador central, que também recebe todas as saídas geradas pelos processadores de autômato. Desta maneira pode-se controlar o processamento da rede e permitir o seu funcionamento cíclico, até a estabilização.

A cada entrada enviada à rede, uma seqüência de ações é realizada pelo processador central, para garantir que cada nó receba sua entrada adequada. Após o processamento dos nós, as saídas dos processadores de autômato devem ser enviadas ao processador central, que as combina de acordo com a topologia da rede, enviando novas entradas para os nós. Quando a rede se estabiliza, o processamento cessa.

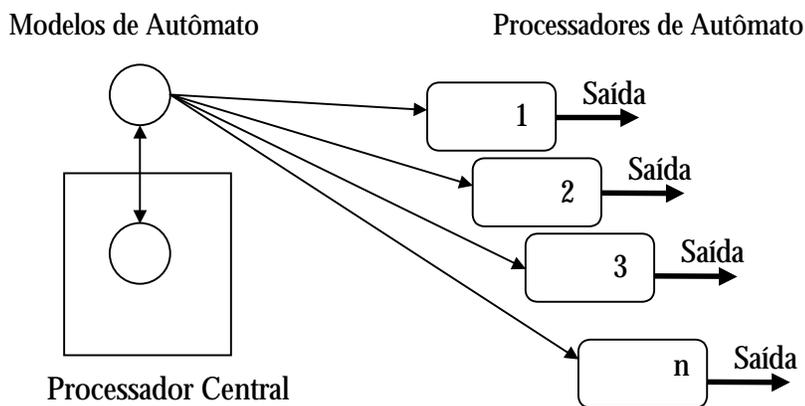


Figura 2 - Utilização de um formalismo adaptável para modelar os nós de uma rede neural

Ao utilizar uma estratégia composta, de forma a que se possa construir modelos em ambientes seriais ou paralelos, procura-se obter bom desempenho e baixo custo em qualquer ambiente.

A simulação de uma rede neural em um dispositivo BSMA pode ser vislumbrada através dos seguintes teoremas, que não estão demonstrados, sendo apenas observada a estratégia de demonstração:

Teorema 1: *Um dispositivo BSMA pode simular uma neural de Hopfield qualquer*

Estratégia de demonstração: Usando a definição do dispositivo BSMA e a definição de rede de Hopfield baseadas em [15]; usando-se a definição de estado da rede de Hopfield, pode-se utilizar o autômato adaptativo como base para construção de uma rede de Hopfield qualquer, na qual os pesos são elementos inteiros. Em cada transição do autômato, o peso do ramo da rede de Hopfield é representado por um símbolo, a ser consumido na transição. Assim, basta construir uma função adaptativa C cuja função é computar e armazenar os valores recebidos até o momento. Como a máquina satisfaz às definições da rede de Hopfield, conclui-se que este dispositivo pode simular uma rede de Hopfield [15].

■

Teorema 2: *Um dispositivo BSMA pode solucionar qualquer problema que esteja apto a representar através de sua estrutura interna, desde que não haja limites, em tempo e espaço ocupado pelas construções, para se encontrar uma solução.*

Estratégia de demonstração: Através do algoritmo de controle do dispositivo, pode ser verificado que a condição de existência de solução é fruto de duas condições anteriores: possibilidade de construção de um modelo inicial e a convergência dos modelos gerados.

A primeira condição é consequência direta da proposição do Teorema, já que só é possível construir um modelo se for possível representá-lo internamente através de um modelo de autômato. Já a segunda condição prega o direcionamento a uma solução, através da convergência da complexidade de Kolmogorov, aproximada pelo dispositivo.

O Teorema 5.1.21 de [6] mostra que há convergência do valor esperado, o que garante que a segunda condição também está preservada. Entretanto, o processo de convergência pode dar-se para valores muito grandes de tempo. Portanto, nem todos os problemas que podem ser modelados terão solução exequível, conforme estipulado anteriormente. Somente terão solução exequível os problemas que possam ser solucionados dentro dos limites de tempo e espaço impostos ao dispositivo.

Logo, qualquer problema para o qual é possível encontrar-se uma representação adequada no dispositivo e não haja limite de tempo e nem de espaço, pode ser solucionado, e a busca será guiada pela medida da complexidade das possíveis soluções geradas.



4. Uso do Dispositivo BSMA em uma ambiente paralelo

A formulação de um esquema de paralelismo pode ser interessante, já que pode agilizar a busca de soluções através do uso de computações efetuadas em paralelo. Mesmo com paralelismo, o funcionamento do dispositivo proposto no capítulo anterior não é alterado em sua forma, porque a busca de soluções continua seguindo o processo original, sendo apenas agilizado ao se executar mais de um modelo em paralelo.

Quanto à troca de mensagens, pode-se utilizar uma estratégia semelhante à utilizada em “PVM” - utiliza em cada processador distribuído pelo sistema de computação um pequeno programa que faz o papel de agente de comunicação, provendo as primitivas de comunicação definidas - [13], ou utilizar um substrato para troca de mensagens, que as próprias linguagens a serem escolhidas devem possuir, desde que esse se mostre bastante adequado.

Um substrato que pode ser utilizado é a Internet, por meio de ambientes de programação baseados na linguagem Java (C++). Pode-se utilizar todo o substrato de troca de mensagens da rede, e também o paralelismo subjacente.

4.1. Estratégias de Paralelismo

Uma estratégia de paralelismo que pode ser adotada é a mesma utilizada em uma rede neural, em que cada processador trata seus dados independentemente dos demais, e envia os resultados ao processo que o iniciou. Tem-se aqui um modelo de paralelismo que utiliza dados (em forma de modelos) diferentes em processadores diversos e independentes e que não necessita de memória compartilhada para seu funcionamento. Basta, para isto, existir um mecanismo de troca de mensagens qualquer, como por exemplo, um sistema operacional de rede, já que as tarefas são tratadas de forma independente.

Para adequar esta proposta às formas correntes de programação em um sistema paralelo, pode-se utilizar as definições de Ben-Ari sobre a divisão de tarefas neste tipo de sistema [13], segundo as seguintes analogias concebidas para lidar com o problema em questão:

- Micro-tarefa definida em [13], por analogia: porção do código de controle do dispositivo que deve ser executada nos diversos processadores, e também a porção do código do elemento controlador que deve ser executada em único processador;
- Macro-tarefa definida em [13], por analogia: a execução de um modelo de solução baseado em autômatos adaptativos propriamente dito.

Com relação aos processadores, pode-se tratar dos mesmos como um conceito, sem referência direta a uma máquina física, mas a uma máquina virtual. Logo, pode-se trabalhar com um modelo paralelo mesmo em uma máquina com um único processador, tratando os modelos como processos, de forma concorrente, em

um mesmo processador físico. Esta observação está sendo colocada de forma a viabilizar o uso do dispositivo BSMA proposto, mesmo em um computador com um único processador.

Portanto, a gerência do processamento dos modelos de autômato adaptativo pode ser realizada através do elemento controlador, que indica o momento de executar mais um passo de computação, o momento de parar a computação, etc. Para que o elemento controlador realize adequadamente suas tarefas há duas possibilidades:

- Utilizar uma variável de controle de passo, compartilhada entre o elemento controlador e os demais modelos de autômato;
- Utilizar a troca de mensagens entre o elemento controlador e os demais modelos em execução.

Para o controle de passos de processamento, a estratégia a ser adotada é o compartilhamento de variável de controle; para os demais eventos a estratégia a ser adotada é a troca de mensagens.

4.2. Implementação de um esquema de Paralelismo

O esquema de paralelismo que poderia ser adotado pressupõe que a máquina física paralela, a ser usada na implementação, tenha a possibilidade de lidar com variáveis compartilhadas. Como não é pretensão desta Tese discutir arquiteturas de implementação de paralelismo, pode-se então adotar um modelo de arquitetura que seja bastante difundido. A literatura indica uma tendência de arquiteturas com área de memória compartilhada diminuta [14], porém suficiente para uso em pequena escala (como é o caso desta pesquisa). Portanto, a estratégia de paralelismo proposta na seção anterior pode ser utilizada, já que é possível adotar um esquema que contemple o uso de variáveis compartilhadas para o presente caso.

Com este esquema de paralelismo, torna-se possível implementar um modelo de solução para ser exercitado no dispositivo BSMA proposto. Um aspecto importante neste caso é que os dados enviados como mensagens entre os processadores serão construções (ou mudanças a serem efetuadas nas construções) baseadas em autômatos e, na prática, cada processo funciona como um interpretador de modelos de autômatos adaptativos, operando de forma concorrente com os demais processos. Para isso, os modelos de autômatos devem estar em um formato que agilize sua execução dentro do interpretador. Em outras palavras, os modelos devem estar em uma codificação conveniente (“compilados”).

Para sincronizar uma operação neste modelo, o processo principal que inicia os demais, necessita de algumas mensagens de controle, como:

- Envio de construção: ao receber esta mensagem, elimina-se a construção anterior, baseada em autômato adaptativo, do agregado que a recebeu;
- Início de processamento: cada construção dentro de um agregado é iniciada;
- Parada: ao receber esta mensagem, o processo em questão interrompe sua execução;
- Envio compulsório de resposta: esta mensagem obriga o processo que a recebe a enviar o estado final de sua operação (informações sobre a execução), e dos resultados obtidos, mesmo parciais, se houver.

Uma observação importante é que este esquema de paralelismo não necessita de uma máquina paralela específica, homogênea, com processamento simétrico. Pode trabalhar em uma rede de computadores heterogênea, já que a estratégia e o esquema adotados não prevêm o uso excessivo de memória compartilhada, e cada processo opera independentemente dos demais.

Desta maneira, não é necessário nem mesmo implementar uma estratégia baseada, por exemplo, em PVM [13], já que a própria linguagem escolhida – no caso a implementação paralela poderá ser feita através de

um processador “LISP” escrito em linguagem “C” – possui um substrato que permite utilizar as funções do sistema operacional para a troca de mensagens, que é bastante adequado.

Como os trechos a serem executados em paralelo são as construções de autômatos, que representam na realidade dados de processamento, encontrar uma linguagem com esta característica não é complicado, já que todas as linguagens de programação paralela possuem primitivas para explicitar a execução em paralelo de trechos de programas.

5. Considerações Finais

Na pesquisa base para este trabalho, propõe-se um método para estruturar modelos de computação, utilizando os autômatos adaptativos como substrato. Outros modelos computacionais importantes, com características diferentes dos autômatos adaptativos, também foram estudados para compor o método e permitir a construção de um dispositivo que pudesse aprender e chegar a soluções de problemas complexos.

Dentre as características diferentes encontradas em outros modelos computacionais acham-se: a possibilidade de introdução de modificações aleatórias, semelhante a uma mutação genética (encontrada em algoritmos genéticos); a combinação de partes estruturais, transições, nos modelos (também encontrada em algoritmos genéticos); a possibilidade de treinamento e o exercício de elementos em paralelo (encontradas nas redes neurais), etc.

Para outros problemas que envolvam otimização, busca da melhor alternativa, o dispositivo BSMA proposto também se mostra propício, já que realiza a busca sempre para o melhor modelo de solução, baseado na redução de complexidade. Basta formular o problema com este direcionamento, ou seja, como um problema de minimização de complexidade expresso em forma de linguagem, e utilizar o dispositivo [12].

Finalmente, o método e o dispositivo BSMA propostos permitem o uso dos autômatos adaptativos para resolver problemas de aprendizado e resolução de problemas, isto porque o método prevê treinamento (devido à simulação de rede neural). Assim um modelo, como em um exemplo de árvore de decisão [6], pode “aprender” a resolver um problema e, a partir de então, inferir respostas a questões ou situações para as quais não foi treinado. Com isso, abre-se mais uma alternativa de estudo na área de inteligência artificial.

A simulação de rede neural proposta parte do pressuposto de que os problemas a serem enfrentados são problemas de linguagem, isto é, podem ser expressos através de uma formulação por meio de gramáticas. Apesar de limitar um pouco a forma de apresentação de problemas, pode-se buscar alguma forma de ampliação através do uso de linguagem natural.

Comentários

Desde já, observa-se que a utilização de linguagem natural para formulação das solicitações do usuário exigirá a elaboração de uma sofisticada interface homem-máquina.

6. Referências Bibliográficas

- [1] JOSÉ NETO, J. Adaptive automata for context-dependent languages. *ACM SIGPLAN Notices*, v. 29, n. 9, p. 115-124, Sep. 1994.
- [2] LEWIS, H. R.; PAPADIMITRIOU, C. H. *Elements of the theory of computation*. New Jersey, Prentice-Hall, Inc, 1998.
- [3] LI, M; VITÁNYI, P. *An introduction to Kolmogorov complexity and its applications*. 2nd. ed., New York, Springer-Verlag, 1997.
- [4] MINSKY, M; SEYMOUR P. *Perceptrons*. Cambridge (MA), MIT Press, 1988.
- [5] POPPER, K.; ECCLES, J. C *O eu e seu cérebro*. Brasília, Editora Universidade de Brasília, 1995.
- [6] ROCHA, R. L. A. *Um método de escolha automática de soluções usando tecnologia adaptativa*. São Paulo, 2000. 211p. Tese (Doutorado) – Escola Politécnica, Universidade de São Paulo.

- [7] RUSSELL, S. J.; NORVIG, P. *Artificial intelligence a modern approach*. New Jersey, Prentice-Hall, Inc., 1995.
- [8] SEARLE, J. *O Mistério da Consciência*. São Paulo, Martins Fontes, 1998.
- [9] SOLOMONOFF, R. J. A formal theory of inductive inference - Part I e Part II. *Information Control*, v.7, p.1-22; p.224-254, 1964.
- [10] SPEARS, W. M.; DE JONG, K. A.; BÄCK, T.; FOGEL, D. B.; DE GARIS, H. An overview of evolutionary computation. In: European Conference on Machine Learning, 1993. *Proceedings*. p. 442-459.
- [11] WOOLDRIDGE, M.; JENNINGS, N. Formalizing the cooperative problem solving process. In: Thirteenth International Workshop on Distributed Artificial Intelligence (IWDAI-94), Lake Quinalt, WA, 1994. *Proceedings*. p. 403-417.
- [12] ROCHA, R. L. A.; JOSÉ NETO, J. Construction of models based on adaptive automata through grammar descriptions. In: IASTED International Conference – Applied Simulation and Modelling (ASM-2000), Banff, Alberta, Canada, 2000. *Proceedings*. [CD-ROM].
- [13] BEN-ARI, M. *Principles of concurrent and distributed programming*. Cambridge (UK), Prentice-Hall International, 1990.
- [14] DUNCAN, Ralph. A survey of parallel computer architectures. *Computer*, Feb., p. 5-16, 1990.
- [15] SIU, K.; ROYCHOWDHURY, V.; KAILATH, T. *Discrete neural computation - a theoretical foundation*. New Jersey, Prentice-Hall Information and System Sciences Series, 1995.
- [16] ROCHA, R. L. A.; NETO J. J., *Um estudo sobre o método indutivo e sua relação com a computação*. LAPTEC 2000, ISBN: 85 8579 529 8, São Paulo, Brasil, 2000, pp. 95-109.