

Uso de algoritmos genéticos em problemas de roteamento de redes de computadores

Sidnei Alves de Araujo

Doutorando em Engenharia Elétrica – USP;
Professor e pesquisador do Departamento
de Ciências Exatas – Uninove.
São Paulo – SP [Brasil]
saraujo@uninove.br

Andre Felipe Henriques Librantz

Pós-Doutorado em Materiais/Modelagem computacional,
Professor e Pesquisador do Departamento
de Ciências Exatas – Uninove.
São Paulo – SP [Brasil]
librantz@uninove.br

Wonder Alexandre Luz Alves

Graduado em Ciências da Computação e
Especialização em Educação Matemática – Uninove.
São Paulo – SP [Brasil]
wonder@uninove.br

Embora as redes de computadores estejam bastante sofisticadas e dotadas de modernos algoritmos, observa-se a necessidade de novas tecnologias para melhorar-lhes o desempenho quanto à velocidade, segurança e confiabilidade. Assim, algumas soluções vêm sendo propostas e entre elas estão os algoritmos genéticos também explorados em outras áreas do conhecimento. Neste trabalho, não só é apresentado o conceito de redes de computadores e algoritmos genéticos, mas também uma alternativa para o roteamento de redes baseada nesses algoritmos, visto que se mostram eficientes na solução de problemas de otimização, podendo ser utilizados em conjunto com os algoritmos de roteamento tradicionais.

Palavras-chave: Algoritmos genéticos.
Problemas de roteamento. Redes de computador.



1 Introdução

Nos últimos anos, tem-se observado grande crescimento no tráfego das redes de computadores, decorrente das inovações tecnológicas e da difusão do acesso à internet. Com isso, cada vez mais seu gerenciamento tem ganhado importância fundamental entre os provedores de serviços de internet, responsáveis pelo roteamento desse tráfego.

O roteamento da rede é a determinação do caminho e o transporte dos pacotes da origem para o destino da rede (TANENBAUM, 1997). No transporte dos dados na rede de computadores, a informação é acondicionada em pequenos pacotes de dados que contêm cabeçalho com um endereço IP (Internet Protocol) e um conjunto de informações necessárias para sua transmissão ao longo da rede. Assim que um pacote chega a um roteador (*hardware* responsável pelo roteamento) da rede, ele consulta uma tabela de roteamento que contém o endereço IP do próximo roteador. Como cada pacote é tratado de forma individual, a ordem em que os pacotes chegam pode não ser a mesma em que são enviados. O protocolo IP apenas envia os pacotes, estando a tarefa de sua reordenação a cargo de outro protocolo chamado TCP (Transmission Control Protocol). Além disso, como o roteamento é tarefa muito complexa, a rede mundial de computadores (internet) é dividida em domínios, com regras próprias para rotear o tráfego em seu interior e outras para rotear o tráfego entre esses domínios (BURIOL, 2003). Um domínio é chamado de Sistema Autônomo (Autonomous System – AS). Ainda há protocolos válidos para atuar dentro de um AS (Interior Gateway Protocols – IGP) ou entre unidades de AS (Exterior Gateway Protocols – EGP).

Dentro dos protocolos, temos os algoritmos de roteamento, cuja função é calcular a rota dos

pacotes de dados. Para estimar o melhor caminho, baseiam-se em padrões de medidas, tais como custo, distância, atraso, confiabilidade, largura de banda e outras. Os algoritmos utilizados para computar informações métricas em um roteamento classificam-se, basicamente, em dois grupos: vetor de distância e estado de ligação (GALLO, 2003). O primeiro determina a distância entre os nós de origem e de destino, além de calcular o número de pulos, buscando gerar dados para que o roteador obtenha tabela com todos os caminhos possíveis. Já o segundo não só considera a largura de banda entre as métricas empregadas, mas também permite que os roteadores troquem informações sobre as ligações estabelecidas entre si.

Neste trabalho, é proposto modelo de roteamento de redes de computadores por meio dos algoritmos genéticos com base no algoritmo vetor de distância, a fim de obter condições mais favoráveis às rotas de redes de computadores.

2 Algoritmos genéticos

Os algoritmos genéticos (AGs) consistem em método de busca e otimização baseado no processo de seleção natural que simula a evolução das espécies (RUSSEL; NORVIG, 1989; HAUPT; HAUPT, 1998; MICHALEWICZ, 1996). Os AGs geram os indivíduos (cadeias de bits), também chamados cromossomos, que evoluem para encontrar a solução de um dado problema. Cada um deles representa uma possível solução para o problema. Com a dinâmica do algoritmo, esses indivíduos competem entre si, e os mais aptos são selecionados para, em seguida, serem cruzados e gerar novos indivíduos mais aptos e, portanto, melhores que os anteriores. Dessa forma, a cada nova geração deve haver um indivíduo mais próximo de uma solução para o

problema. Caso o AG seja desenvolvido corretamente, deve convergir para ela. Ao contrário dos processos de busca e otimização tradicionais, os AGs trabalham em um processo de busca sobre a população de indivíduos em paralelo, avaliando as possibilidades em diferentes regiões do espaço de buscas (REF).

Os AGs constituem ferramenta poderosa para resolução de problemas complexos cujos espaços de busca das soluções ótimas são muito grandes para determiná-las com precisão por um método direto. Cabe ressaltar que essas soluções ótimas, em alguns casos, podem nem existir. Além disso, muitas vezes, o que realmente precisamos é de aproximação que nos dê resultado satisfatório no contexto do problema (CRUZ, A. J. O.; DEMASI, 2002).

Os elementos básicos de um AG são: (a) cromossomo conjunto de *bits* (também chamado indivíduo da população); (b) gene subconjunto de *bits* do cromossomo e (c) alelo cada *bit* de um gene.

A estrutura básica de um AG contempla quatro principais operações: o cálculo de aptidão (*fitness*), a seleção, o cruzamento (*crossover*) e a mutação, ilustrados no diagrama da Figura 1.

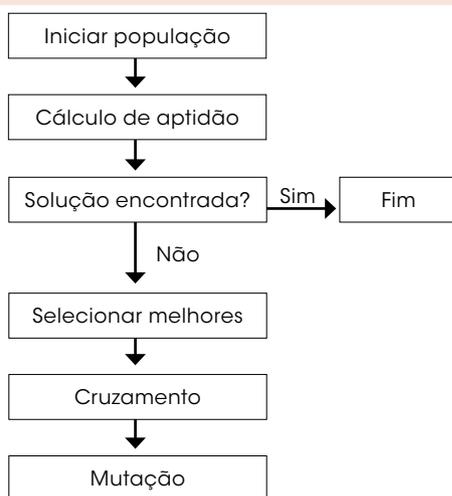


Figura 1: Estrutura básica do AG

Fonte: Os autores.

O processo de funcionamento de um AG começa com uma população de indivíduos (estados de busca) gerados aleatoriamente, chamados de população inicial. Um indivíduo normalmente é formado por uma *string* binária convertida para um valor decimal a ser avaliado de acordo com seu *fitness*. O tamanho da população é definido com base no problema a ser solucionado. Essas estruturas são, então, avaliadas para gerar oportunidades reprodutivas de forma que os cromossomos que representam as “melhores soluções” tenham mais chances de se reproduzir. A definição de solução melhor ou pior está relacionada ao problema e é quantificada de acordo com uma função de aptidão.

O cálculo da aptidão é utilizado para analisar os indivíduos selecionados a partir da população. Normalmente, por se tratar de otimização, a aptidão serve para minimizar ou maximizar uma função objetivo. Ela fornece uma medida da proximidade da solução em relação a um conjunto de parâmetros, visando encontrar o ponto ótimo. Neste trabalho, a função de aptidão é a própria função objetivo definida na equação 1.

O processo de seleção consiste na escolha dos melhores indivíduos que serão os reprodutores. Dessa forma, aqueles com maior aptidão são selecionados para reprodução, e os outros, descartados. A seleção de cada indivíduo é proporcional à sua aptidão. O *crossover* tem a função de recombinar o material genético dos indivíduos anteriores. A escolha de quem vai cruzar com quem pode ser feita aleatoriamente ou seguir alguma regra definida pelo desenvolvedor do algoritmo. Normalmente, os melhores indivíduos geram mais filhos que os demais. A figura 2 mostra o esquema de *crossover* simples (com apenas um ponto de corte).

Para evitar que o AG convirja muito cedo para mínimos locais, é feita operação de mutação,

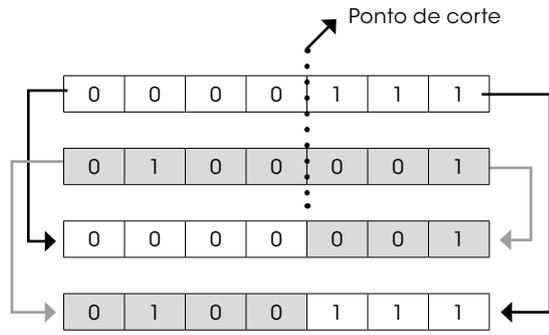


Figura 2: Crossover simples com um ponto de corte
Fonte: Os autores.

o que promove a alteração de um ou mais alelos de um cromossomo sorteado aleatoriamente com determinada probabilidade. Quando o cromossomo é representado por uma *string* binária, os valores 1 são trocados por 0, e vice-versa (figura 3).

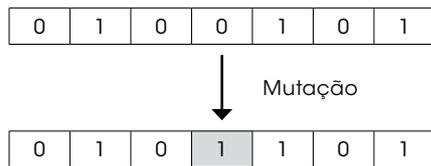


Figura 3: Exemplo de mutação
Fonte: Os autores.

A escolha adequada dos parâmetros do AG é de fundamental importância para convergência rápida. Um fator importante é a taxa de alelos a serem mutados, que deve variar de 5% a 10%, de acordo com o tamanho da população.

3 Problema de encontrar um caminho mínimo

O problema de encontrar o caminho mínimo (isto é, mais curto, mais rápido ou menos custoso) é um clássico exemplo de problema de otimização combinatória (BATALHA, 2008). Primeiro, para resolver esse tipo de problema, é preciso calcular a distância de todas as rotas e escolher a de menor distância. Esse raciocínio usa o famoso método de força bruta ou da busca exaustiva (CORMEN,

2001), ou seja, utiliza muito poder computacional para resolver o problema. Esse método nos dá um número de opções igual a $n(n-1)(n-2)...(2)(1) = n!$. Por isso, a fatorial de um número cresce muito rapidamente. Por exemplo, se houver 100 roteadores interligados, teremos 10^{158} opções de rotas. Se pudéssemos testar um bilhão de soluções por segundo, demoraríamos um tempo igual a 10^{141} anos para encontrar a melhor solução.

Pode-se perceber que esse problema é intratável¹ e, tendo em vista sua importância, precisamos de uma técnica capaz de resolvê-lo. São conhecidas muitas técnicas capazes de resolver problemas de busca, de forma exata, entre as quais a busca em profundidade e largura. Se alguma for capaz de resolver o problema de forma satisfatória (em termos de tempo de processamento), é melhor utilizá-la. Entretanto, ao encontrar problema intratável, uma boa solução será utilizar as técnicas de aproximação, tais como os algoritmos genéticos.

4 Metodologia

Na Figura 4, apresenta-se uma pequena rede, em que os nós representam os roteadores, e as arestas, uma função de custo associadas aos nós. Esse pequeno modelo será utilizado para explicar a forma de modelar os AGs para o problema do caminho mínimo.

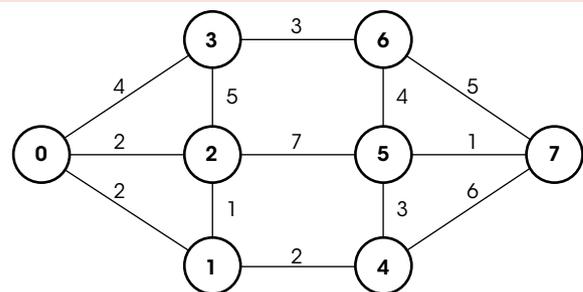
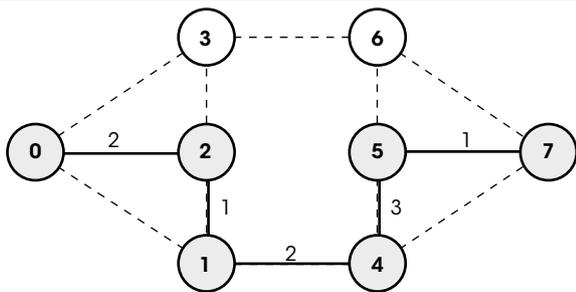


Figura 4: Exemplo de uma rede
Fonte: Os autores.

A idéia de usar os AGs para determinar o caminho mínimo entre dois roteadores consiste em determinar uma população de indivíduos (ou cromossomos), cada um deles representando um possível caminho para trafegar os dados entre os roteadores (genes) na rede. Veja na Figura 5 que os nós coloridos representam o indivíduo mais apto (resposta) no processo de evolução dos AGs. Este exemplo mostra a rota entre os roteadores de 0 até 7:



Caminho	0	2	1	4	5	7
Cromossomo	000	010	001	100	101	111

Figura 5: Exemplo de uma rede, na qual os nós coloridos representam um possível caminho

Fonte: Os autores.

Para conhecer o tamanho dos genes e do cromossomo, é necessário saber o número de nós da rede. No caso do exemplo em destaque, são observados oito nós; logo, o total em números binários está compreendido entre 000 e 111 (três dígitos), considerando-se a contagem do nó zero ao sétimo. Cada um dos oito nós representa um gene. Isso significa que, se convertermos os valores binários do cromossomo para decimal, obteremos a representação de uma rota possível, como mostrado no esquema anterior.

Já para determinar o tamanho máximo do cromossomo, basta calcular o caminho, excluindo o nó inicial e o final, visto que sabemos, *a priori*, qual o nó de origem e o de destino e desejamos descobrir apenas o melhor caminho entre eles. Assim, se considerássemos os nós restantes, nosso cromos-

somo teria “nº nós” – 2, o que seria suficiente para ir do nó inicial ao final, sem revisitação de nós. Assim, sabemos que, para isso, seria necessário “nº nós” – 1 pulos, ou seja, 7 pulos. A Figura 6 ilustra melhor a posição do cromossomo.

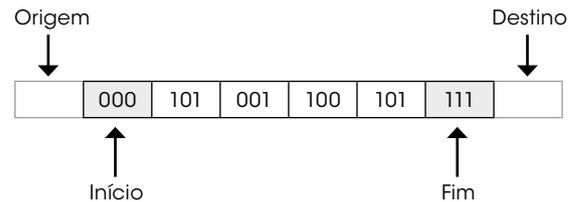


Figura 6: Genes de início e fim

Fonte: Os autores.

O cromossomo em questão (Figura 6) contém seis genes, em que três alelos formam um gene, totalizando 18 dígitos binários (alelos). Cada gene, ao ser convertido para decimal, representa um nó da rede. Assim, partimos do nó de origem, tomando cada par e acumulando seu respectivo custo. Seguimos a ordem do caminho gerado, percorrendo nó a nó, até o destino. Verificamos, a cada nó, sua validade e se o destino foi encontrado. No caso de um par de nós não representar ligação válida, a análise é interrompida nesse ponto do cromossomo. Por exemplo, suponha o caminho da Figura 7.

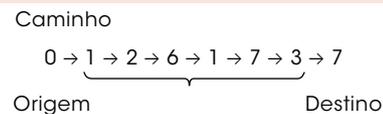


Figura 7: Representação de um caminho inválido

Fonte: Os autores.

Se analisarmos novamente a Figura 4, veremos que, a partir do segundo nó gerado (nó 2), o caminho não é legítimo, ou seja, o nó 2 não apresenta ligação com o nó 6. Logo, os custos só seriam acumulados até o último nó válido. Consideremos, agora (Figura 8), que nossa origem seja o nó 3, e nosso destino, o 4.

the necessity of new technologies is observed to improve the performance regarding speed, security and reliability. In this context, some solutions have been proposed and, between them, it is the Genetic Algorithms technique that is also explored in other areas of knowledge. In this article, we introduced the concept of computers networks, the Genetic Algorithm technique and its application in routing problems solving. We also presented an alternative to find the routing based on Genetic Algorithms, once it has been showed efficient as a solution in optimization problems and it can be used together with the traditional routing algorithms.

Key words: Genetic algorithms. Routing problems. Computer networks.

Notas

- 1 Problema intratável é aquele cujo espaço de solução é tão amplo que, para percorrê-lo, precisamos de um tempo muito longo; em decorrência de tal problema, pode-se considerá-lo um problema sem solução.

Referências

- BARBOSA, H.J.C. *Introdução aos Algoritmos Genéticos: minicurso*. Rio de Janeiro: SBMAC, 1997.
- GALLO, M.A.; HANCOOK, W.M. *Comunicação entre Computadores e tecnologias de rede*. 1. ed. São Paulo: Thomson, 2003.
- GOLDBERG, D.E. *Genetic algorithms in search, optimization, and machine learning*. 1st ed. Boston: Addison-Wesley, 1989.

HAUPT, R.L.; HAUPT, S.E. The Binary Genetic Algorithm. In: HAUPT, R.L.; HAUPT, S.E. *Practical Genetic Algorithms*. 1st ed. New York: Wiley-Interscience, 1998. p. 25-48.

TANENBAUM, A.S. A Camada de Rede. In: TANENBAUM, A.S. *Redes de computadores*. Rio de Janeiro: Campus, 1997. p. 387-546.

BURIOL, L.S.; FRANÇA, P.M.; RESENDE, M.G.C.; RIBEIRO, C.C. Otimizando o roteamento do tráfego na Internet, In: Anais do XXXV Simpósio Brasileiro de Pesquisa Operacional, Natal, 2003. p. 1722-1732.

RUSSEL, S.; NORVIG, P. *Artificial intelligence: a modern approach*. New Jersey: Prentice Hall, 1995.

GOLDBERG, D. E. Genetic algorithms in search, optimization, and machine learning. Massachusetts: Addison-Wesley, 1989.

HAUPT, R. L.; HAUPT, S. E. The Binary Genetic Algorithm. In: HAUPT, R. L.; HAUPT, S. E. *Practical Genetic Algorithms*. 1. ed., New York: Wiley-Interscience, 1998. Cap. 2, p. 25-48.

MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. New York: Springer Berlin: Heidelberg, 1996.

CRUZ, A. J. O.; DEMASI, P. Algoritmos coevolucionários cooperativos em jogos. In: Proceedings of the 1st Brazilian Workshop in Games and Digital Entertainment, v. 1, p. 1-6, 2002.

CORMEN, Th.H., LEISERSON Ch.E., RIVEST R.L., STEIN C. *Introduction to Algorithms*, 2nd edition. MIT: Press & McGraw-Hill, 2001.

BATALHA, M. O. *Introdução à Engenharia de Produção*. Rio de Janeiro: Elsevier, 2008.

Recebido em 10 abr. 2007 / aprovado em 23 jul. 2007

Para referenciar este texto

ARAÚJO, S. A. de; LIBRANTZ, A. F. H; ALVES, W. A. L. Uso de algoritmos genéticos em problemas de roteamento de redes de computadores. *Exacta*, São Paulo, v. 5, n. 2, p. 321-327, jul./dez. 2007.

