

Algoritmo genético

Origem: Wikipédia, a enciclopédia livre.

Um **algoritmo genético** (**AG**) é uma técnica de busca utilizada na ciência da computação e em investigação operacional para achar soluções aproximadas em problemas de otimização e busca, fundamentado principalmente pelo americano John Henry Holland. Algoritmos genéticos são uma classe particular de algoritmos evolutivos que usam técnicas inspiradas pela biologia evolutiva como hereditariedade, mutação, seleção natural e recombinação (ou *crossing over*)^[1]. Alguns exemplos do uso de AG incluem otimização de aprendizagem de árvore de decisão para melhor performance, resolução de algoritmo de sudoku^[2], otimização de hiperparâmetros, e etc.

Visão Geral

Algoritmos Genéticos (AG) são implementados como uma simulação de computador em que uma população de representações abstratas de solução é selecionada em busca de soluções melhores. A evolução geralmente se inicia a partir de um conjunto de soluções criado aleatoriamente e é realizada por meio de gerações. A cada geração, a adaptação de cada solução na população é avaliada, alguns indivíduos são selecionados para a próxima geração, e recombinados ou mutados para formar uma nova população. A nova população então é utilizada como entrada para a próxima iteração do algoritmo.

O que é AG

Algoritmos genéticos diferem dos algoritmos tradicionais de otimização em basicamente quatro aspectos:

- Baseiam-se em uma codificação do conjunto das soluções possíveis, e não nos parâmetros da otimização em si;
- os resultados são apresentados como uma população de soluções e não como uma solução única;
- não necessitam de nenhum conhecimento derivado do problema, apenas de uma forma de avaliação do resultado;
- usam transições probabilísticas e não regras determinísticas.^[3]

```
função AlgoritmoGenético(população, função-objetivo) saídas: indivíduo
  entradas: população→ uma lista de indivíduos
           função-objetivo→ uma função que recebe um indivíduo e retorna um número real.
  repetir
    lista de pais := seleção(população, função-objetivo)
    população := reprodução(lista de pais)
  enquanto nenhuma condição de parada for atingida
  retorna o melhor indivíduo da população de acordo com a função-objetivo
```

Componentes principais

função-objetivo

A **função-objetivo** é o objeto de nossa otimização. Pode ser um problema de otimização, um conjunto de teste para identificar os indivíduos mais aptos, ou mesmo uma "caixa preta" onde sabemos apenas o formato das entradas e nos retorna um valor que queremos otimizar. A grande vantagem dos algoritmos genéticos está no fato de não precisarmos saber como funciona esta função objetivo, apenas tê-la disponível para ser aplicada aos indivíduos e comparar os resultados.

indivíduo

O **indivíduo** é meramente um portador do seu **código genético**. O código genético é uma representação do espaço de busca do problema a ser resolvido, em geral na forma de sequências de bits. Por exemplo, para otimizações em problemas cujos valores de entrada são inteiros positivos de valor menor que 255 podemos usar 8 bits, com a representação binária normal, ou ainda uma forma de código gray. Problemas com múltiplas entradas podem combinar as entradas em uma única sequência de bits, ou trabalhar com mais de um "cromossomo", cada um representando uma das entradas. O código genético deve ser uma representação capaz de representar todo o conjunto dos valores no espaço de busca, e precisa ter tamanho finito.^[4]

seleção

A **seleção** também é outra parte chave do algoritmo. Em geral, usa-se o algoritmo de seleção por "roleta", onde os indivíduos são ordenados de acordo com a função-objetivo e lhes são atribuídas probabilidades decrescentes de serem escolhidos - probabilidades essas proporcionais à razão entre a adequação do indivíduo e a soma das adequações de todos os indivíduos da população. A escolha é feita então aleatoriamente de acordo com essas probabilidades. Dessa forma conseguimos escolher como pais os mais bem adaptados, sem deixar de lado a diversidade dos menos adaptados. Outras formas de seleção podem, ainda, ser aplicadas dependendo do problema a ser tratado. Como exemplos pode-se citar a seleção por "torneio" (onde são selecionados diversos pequenos subconjuntos da população, sendo selecionado o indivíduo de maior adequação de cada um desses grupos), a seleção por "classificação" ou "ranking" (semelhante à seleção por "roleta", com a diferença de que a probabilidade de seleção é relacionada à sua posição na ordenação dos indivíduos da população e não à sua adequação em si) e a seleção por "truncamento" (onde são selecionados os N melhores indivíduos da população, descartando-se os outros).^[5]

reprodução

A **reprodução**, tradicionalmente, é dividida em três etapas: acasalamento, recombinação e mutação. O acasalamento é a escolha de dois indivíduos para se reproduzirem (geralmente gerando dois descendentes para manter o tamanho populacional). A recombinação, ou *crossing-over* é um processo que imita o processo biológico homônimo na reprodução sexuada: os descendentes recebem em seu código genético parte do código genético do pai e parte do código da mãe. Esta recombinação garante que os melhores indivíduos sejam capazes de trocar entre si as informações que os levam a ser mais aptos a sobreviver, e assim gerar descendentes ainda mais aptos. Por último vem as mutações, que são feitas com probabilidade a mais baixa possível, e tem como objetivo permitir maior variabilidade genética na população, impedindo que a busca fique estagnada em um mínimo local.^[6]

Programação Genética

Por ser um algoritmo extremamente simples e eficiente, existem diversas variações em cima do algoritmo genético básico para se obter resultados melhores ou mesmo tratar novas classes de problemas. Uma dessas variações é a Programação genética. Na Programação genética os

indivíduos representam pequenos programas de computador que serão avaliados de acordo com o resultado de sua execução. Estes programas podem ser expressões simples, como fórmulas aritméticas ou programas complexos, com operações de laço e condicionais, típicas de uma linguagem de programação comum.^{[7][8]}

Bibliotecas e Frameworks para Algoritmos Genéticos

- [EvolveDotNet \(https://code.google.com/p/evolvedotnet/\)](https://code.google.com/p/evolvedotnet/) (Framework open-source para Algoritmos Genéticos - C#)
- [GALib \(http://lancet.mit.edu/ga/\)](http://lancet.mit.edu/ga/) (Framework open-source para Algoritmos Genéticos - C++)
- [GAUL \(http://gaul.sourceforge.net\)](http://gaul.sourceforge.net) (Biblioteca open-source para Algoritmos Genéticos e metaheurísticas - C)
- [GeneticSharp \(https://github.com/giacomelli/GeneticSharp\)](https://github.com/giacomelli/GeneticSharp) (Biblioteca open-source e multiplataforma para Algoritmos Genéticos - C#)
- [JAGA \(https://web.archive.org/web/20110422025937/http://www.jaga.org/\)](https://web.archive.org/web/20110422025937/http://www.jaga.org/) (Pacote open-source para Algoritmos Genéticos e Programação Genética - Java)
- [JGAP \(http://jgap.sourceforge.net/\)](http://jgap.sourceforge.net/) (Pacote open-source para Algoritmos Genéticos - Java)
- [jMetal \(http://jmetal.sourceforge.net\)](http://jmetal.sourceforge.net) (Framework open-source para otimização multiobjetivo que contém Algoritmos Genéticos - Java)
- [Pyevolve \(http://pyevolve.sourceforge.net\)](http://pyevolve.sourceforge.net) (Framework open-source para Algoritmos Genéticos e Programação Genética - Python)

Bibliografia

- KOZA, J.R. (1992). *Genetic Programming. On the Programming of Computers by Means of Natural Selection*. [S.I.]: MIT Press
- GOLDBERG, David E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. EUA: Addison-Wesley. 0-201-15767-5
- NORVIG, Peter, RUSSEL, Stuart (1995). *Artificial Intelligence. A Modern Approach*. Upper Saddle River, NJ, EUA: Prentice Hall. 0-13-103805-2
- Linden, Ricardo (2008). *Algoritmos Genéticos - uma importante ferramenta da inteligência computacional - 2ª Edição*. BR: Brasport. 9788574523736

Ver também

- [Lista de algoritmos](#)
- [Recombinação \(computação evolutiva\)](#)
- [Estratégias evolutivas](#)
- [Programação genética](#)
- [Computação natural](#)
- [Inteligência computacional](#)

Referências

1. Mitchell, Melanie (1996). *An introduction to genetic algorithms* (<https://www.worldcat.org/oclc/42854439>). Cambridge, Mass.: MIT Press. OCLC 42854439 (<https://www.worldcat.org/oclc/42854439>)
2. Gerges, Firas; Zouein, Germain; Azar, Danielle (2018). «Genetic Algorithms with Local Optima Handling to Solve Sudoku Puzzles» (<http://dl.acm.org/citation.cfm?doid=3194452.3194463>). Chengdu, China: ACM Press (em inglês): 19–22. ISBN 978-1-4503-6419-5.

- doi:10.1145/3194452.3194463 (<https://dx.doi.org/10.1145%2F3194452.3194463>). Consultado em 31 de agosto de 2022
- Goldberg, David E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. EUA: Addison-Wesley. 0-201-15767-5
 - Para uma discussão sobre as formas de representação do espaço de busca, veja: Goldberg, David E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. EUA: Addison-Wesley. 0-201-15767-5 página 80
 - Veja em Linden, Ricardo (2008). *Algoritmos Genéticos - uma importante ferramenta da inteligência computacional - 2ª Edição*. BR: Brasport. 9788574523736 Capítulo 9 - Outros Tipos de Seleção.
 - Veja em Goldberg, David E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. EUA: Addison-Wesley. 0-201-15767-5 página 147 para ver outras orperações que podem ser aplicadas nos indivíduos para a reprodução.
 - KOZA, J.R. (1992). *Genetic Programming. On the Programming of Computers by Means of Natural Selection*. [S.I.]: MIT Press
 - Poli, R., Langdon, W. B., McPhee, N. F. (2008). *A Field Guide to Genetic Programming*. [S.I.]: freely available via Lulu.com

Ligações externas

- Introdução aos Algoritmos Genéticos (<http://www.obitko.com/tutorials/genetic-algorithms/portuguese/>)
 - Visão geral sobre algoritmos genéticos (<http://www.demiurgo.com.br/2009/05/20/algoritmos-geneticos-visao-geral/>)^[*ligação inativa*]
 - Trabalho de graduação sobre Algoritmos Genéticos (<http://iaufes20092.pbworks.com/AG+-+Quadro+Cognitivo+7>)
 - Texto sobre Programação e Algoritmos Genéticos (<http://adrianoyamamoto.blogspot.com/2011/06/programacao-genetica.html>)
 - Computação Bioinspirada (<https://web.archive.org/web/20150518092208/http://computacaobioinspirada.com.br/>)
 - Criação de Algoritmos Genéticos em Java para Iniciantes (<http://www.cybtricks.com/criar-algoritmo-genetico-java-iniciantes-parte-2/>)
-

Obtida de "https://pt.wikipedia.org/w/index.php?title=Algoritmo_genético&oldid=64317173"