

Fabricio Kassardjian

Algoritmos Genéticos

Um algoritmo evolucionário para otimização

Brasil

15 de outubro de 2019

Fabricio Kassardjian

Algoritmos Genéticos
Um algoritmo evolucionário para otimização

Monografia de final de curso, Instituto de Matemática e Estatística - USP, Matemática Aplicada

Universidade de São Paulo – USP
Instituto de Matemática e Estatística
Bacharelado em Matemática Aplicada e Computacional

Orientador: Anatoli Iambartsev

Brasil
15 de outubro de 2019

Fabricio Kassardjian

Algoritmos Genéticos

Um algoritmo evolucionário para otimização/ Fabricio Kassardjian. – Brasil, 15 de outubro de 2019-

40p. : il. (algumas color.) ; 30 cm.

Orientador: Anatoli Iambartsev

Monografia – Universidade de São Paulo – USP

Instituto de Matemática e Estatística

Bacharelado em Matemática Aplicada e Computacional, 15 de outubro de 2019.

1. Algoritmos Evolutivos. 2. Algoritmos Genéticos. 3. Otimização. I. Orientador.
II. Universidade de São Paulo - USP. III. Instituto de Matemática e Estatística - IME.
IV. Título

Fabricio Kassardjian

Algoritmos Genéticos

Um algoritmo evolucionário para otimização

Monografia de final de curso, Instituto de Matemática e Estatística - USP, Matemática Aplicada

Trabalho aprovado. Brasil, 24 de novembro de 2012:

Anatoli Iambartsev
Orientador

Professor
Convidado 1

Professor
Convidado 2

Brasil
15 de outubro de 2019

Resumo

Um dos problemas mais comuns em matemática é a busca de soluções que maximizem ou minimizem certa função e temos a disposição diversas formas de resolver que podem ser classificadas em técnicas baseadas em cálculo, aleatórias ou métodos que verificam todo o espaço solução pela melhor delas. Dentre as técnicas aleatórias temos aquelas puramente aleatória como o *random walk* e aquelas consideradas aleatórias guiadas como por exemplo o *simulated annealing* ou resfriamento simulado. No âmbito de sistemas evolutivos, os algoritmos genéticos podem ser classificado também como um método aleatório guiado, onde o estado corrente influencia a próxima escolha para a solução.

O objetivo desse trabalho é verificar e avaliar os algoritmos genéticos, explorando as técnicas e os mecanismos usados nesse método. Para tal será desenvolvido o algoritmo em uma linguagem de programação usando o paradigma de orientação a objetos e aplicá-lo em um problema de busca pela solução ótima, sendo que buscaremos a melhor solução para um modelo de Ising. Com isso poderemos avaliar o funcionamento e performance do método e como os parâmetros utilizados podem influenciar o resultado

Palavras-chave: Algoritmos Evolutivos. Algoritmos Genéticos. Otimização. Programação Orientada a Objetos.

Abstract

This is the english abstract.

Keywords: Evolutionary Algorithms. Genetic Algorithms. Optimization. Object Oriented Programming.

Lista de ilustrações

Figura 1 – Técnicas de otimização global - (COELLO; LAMONT; VELDHUIZEN, 2007)	22
Figura 2 – Exemplo de reprodução com crossover - (KLUG et al., 2011)	24
Figura 3 – Exemplo para NFL - (GOLDBERG, 1989)	27

Lista de quadros

Lista de tabelas

Lista de abreviaturas e siglas

SGA	Simple Genetic Algorithm
GA	Genetic Algorithm
NFL	No-Free-Lunch Theorem
EA	Evolutionary Algorithm
OO	Orientação a objetos

Lista de símbolos

Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
\in	Pertence

Sumário

1	INTRODUÇÃO	21
1.1	Algoritmos evolucionários	21
1.2	Biologia	23
1.3	Surgimento do algoritmo genético	25
1.4	Teorema da inexistência do almoço grátis	27
2	ALGORITMOS GENÉTICOS	29
2.1	Seleção	29
2.2	Crossover	29
2.3	Mutação	29
2.4	Esquemas	29
2.5	Fundamentos teóricos	29
3	IMPLEMENTAÇÃO DO ALGORITMO GENÉTICO	31
3.1	Modelo de Ising	31
3.2	Codificação	31
3.3	Seleção e avaliação	32
3.4	Crossover e mutação	32
3.5	Resultados	32
4	CONCLUSÃO	33
	REFERÊNCIAS	35
	APÊNDICES	37
	ANEXOS	39

1 Introdução

1.1 Algoritmos evolucionários

Os algoritmos evolucionários são um subconjunto da computação evolutiva, sendo essa um conjunto de algoritmos para busca de soluções ótimas globais. A computação evolutiva é baseada na evolução das espécies definida na biologia, e assim os algoritmos evolucionários se baseiam em processos encontrados na natureza de seleção, reprodução e mutação das espécies, com o propósito de otimizar a solução para um problema definido.

Um problema de otimização é definido como maximizar, ou minimizar, $f(x)$ sujeito a $g_i(x) \leq 0, i = \{1, \dots, m\}$, e $h_j(x) = 0, j = \{1, \dots, n\}$ com $x \in \Omega$. A solução maximiza, ou minimiza, o escalar $f(x)$ onde x é um vetor com dimensão n , $x = \{x_1, x_2, \dots, x_n\}$ do espaço de soluções Ω . (COELLO; LAMONT; VELDHUIZEN, 2007)

Dessa forma o problema de otimização tem os seguintes componentes:

- Função objetivo $f(x)$: função de avaliação que deve se minimizada ou maximizada;
- As restrições $g_i(x)$ e $h_j(x)$: definem limites para as soluções que são permitidas;
- Espaço de soluções Ω : Conjunto com todas as possíveis soluções para o problema;

De forma mais simplificada podemos escrever o problema de otimização como, sendo Ω o espaço de soluções e a função

$$g : \Omega \rightarrow \mathbb{R}$$

e a solução o vetor $x \in \Omega$ tal que

$$\arg \min_{x \in \Omega} g(x)$$

que pode facilmente ser convertido para um problema de maximização usando $-g(x)$

Para atingir esse objetivo os algoritmos evolutivos trabalham com uma população de indivíduos que indicaremos como soluções candidatas para o problema. Baseado na avaliação de cada individuo com relação ao seu ambiente, em nosso contexto a função de avaliação, e usando operadores definidos para seleção e evolução, a população vai sendo modificada até que se atinja um resultado satisfatório para o problema. Assim como no processo de evolução das espécies definidos por Darwin, que introduziu o conceito de seleção natural através da sobrevivência do mais apto, os indivíduos da população que tem melhores avaliações, ou seja, que melhor se adaptam ao ambiente, possuem mais probabilidade de sobrevivência e assim se reproduzirem.

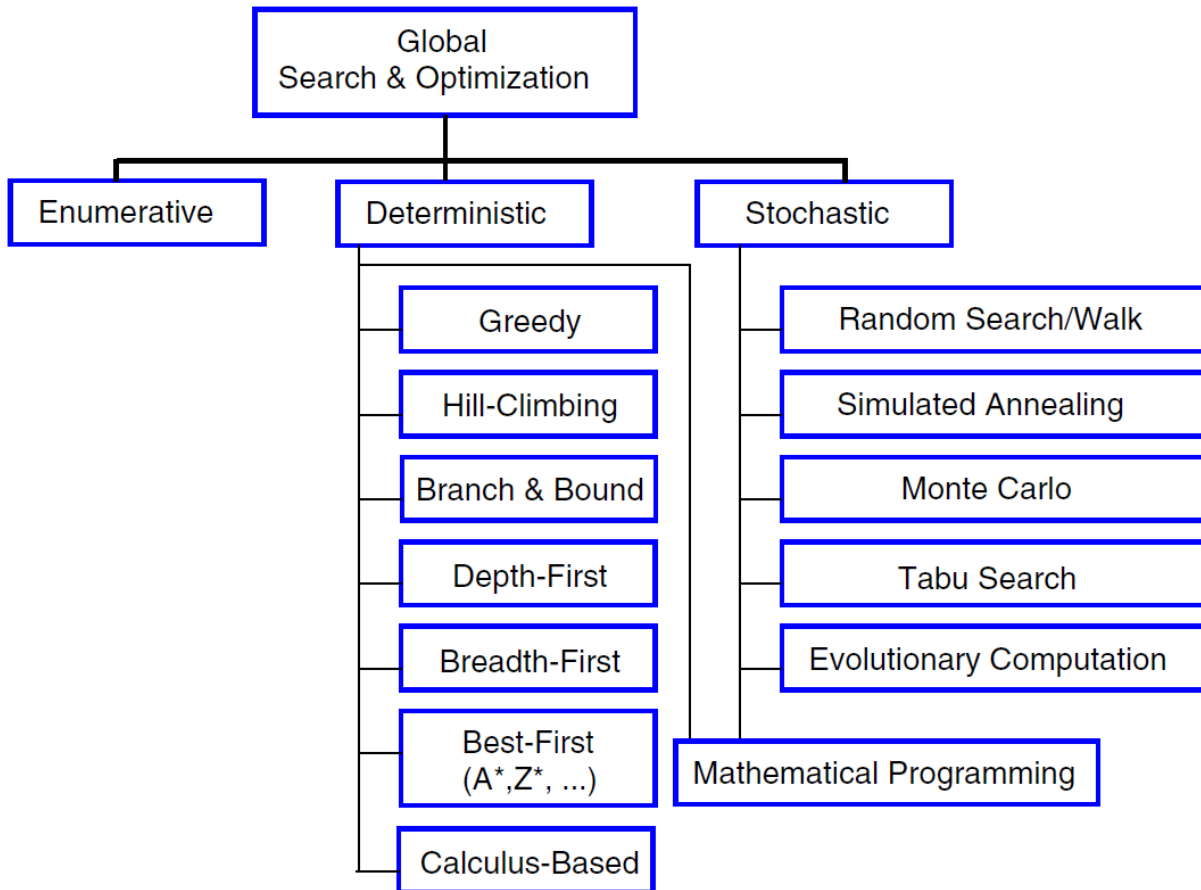


Figura 1 – Técnicas de otimização global - (COELLO; LAMONT; VELDHUIZEN, 2007)

Existem vários métodos para se otimizar um problema conforme pode ser visto na Figura 1. Existem os métodos enumerativos, que se tornam impraticáveis quando o Ω é muito grande, pois o algoritmo deveria testar todas as soluções possíveis. Os algoritmos determinísticos são os mais eficientes em determinadas condições, como por exemplo o comportamento de $f(x)$. Se a função objetivo possui múltiplos máximos (ou mínimos) alguns algoritmos determinísticos, como o *hill-climbing* por exemplo, podem ficar 'presos' em soluções locais e não encontrar a solução global. A última categoria, onde se encontram também os algoritmos evolucionários, são os estocásticos (ou randômicos). Possuem a vantagem de não ficar presos em soluções locais, porém nem sempre obterão a melhor solução.

Dentre a categoria dos métodos estocásticos existem os que são completamente aleatórios, como a busca randômica (ou passeio aleatório), e os que são de alguma forma guiado como o método de Monte Carlo por exemplo. O algoritmo evolucionário se encaixa nessa última definição, onde existem componentes aleatórios atuando na seleção e reprodução dos indivíduos mas de forma guiada pelos resultados da função de avaliação.

De acordo com LINDEN, os AE são **heurísticas**¹ que não asseguram obter o

¹ Heurísticas são algoritmos polinomiais que usualmente tendem a encontrar soluções ótimas ou próximas

melhor resultado possível, e além disso o resultado pode diferir entre as execuções do algoritmo.

Para [SIVANANDAM; DEEPA](#), um algoritmo evolucionário são processos estocásticos e iterativos que operam em um conjunto de indivíduos (população). Cada indivíduo representa uma possível solução para o problema de otimização ou busca, sendo que os parâmetros estão de alguma forma codificados nesses indivíduos. A população inicial é gerada aleatoriamente e são avaliados usando alguma função, que determina o quão bem o indivíduo responde ao problema. Esse valor determina a direção de busca do algoritmo.

1.2 Biologia

A ideia por trás dos algoritmos evolucionários e por consequência do algoritmo genético, é a teoria de evolução das espécies na natureza de **Darwin**, onde a sobrevivência de cada indivíduo é determinada por como ele se adapta ao seu meio. Assim aqueles que conseguiam vantagens sobre os demais por ter uma maior sobrevivência se reproduziam mais, e assim, passavam para as próximas gerações essas características que os diferenciavam. Darwin chamou de seleção natural esse mecanismo da sobrevivência dos mais aptos.

Contudo Darwin não sabia explicar como essa informação era passada dos ancestrais para os descendente, onde então entram as descobertas de **Mendel**, que através dos conceitos de genética determinava como características eram compartilhadas entre pais e filhos.

A unidade básica de informação é o gene, que é um bloco de sequências de DNA e o conjunto de genes formam o cromossomo. Cada gene tem um *locus*, que define a região dentro do cromossomo onde está localizado, e possui um conjunto de valores possíveis chamados de alelos. Os genes controlam as características do indivíduo, sendo que a expressão dessas no indivíduo é denominada de fenótipo. Assim um conjunto específicos de genes define o genótipo do indivíduo que está associado a um fenótipo, que apresenta as características codificadas no genótipo e que podem ser modificadas pelo ambiente.

Organismos com cromossomos combinados em pares são chamados de diplóides em contraste com os que não possuem pares, chamados de haplóides. Na natureza a maioria dos seres mais complexos que se reproduzem sexualmente são normalmente diplóides e possuem um ou mais pares de cromossomos sendo que sua quantidade e tamanho dependem de cada ser vivo.

Durante a reprodução ocorre a transmissão da informação que pode ser de dois tipos: assexuada e sexuada. Na reprodução assexuada, o organismo replica a si mesmo, presente mais em seres simples, não representa tanta diversidade pois como não existe

delas, mas sem garantias

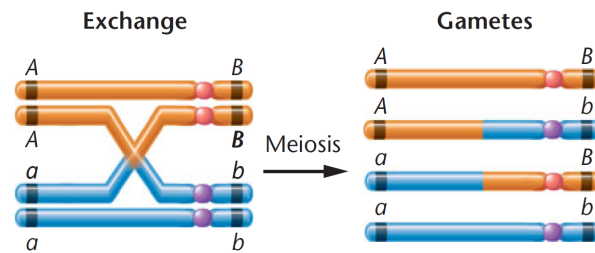


Figura 2 – Exemplo de reprodução com crossover - (KLUG et al., 2011)

combinação de material genético entre dois seres, ocorre apenas a retransmissão de material genético, somente sujeito a alterações por mutação na cópia.

Na reprodução sexuada de seres diplóides, há presença de dois indivíduos que compartilham seu material genético para formar um novo organismo. No início da reprodução, existe a cópia do material genético e recombinação, também chamado de *crossover* (Figura 2). Esse processo é feito com os cromossomos se cruzando, por isso o termo *crossover*, um sobre o outro em um ou mais pontos havendo assim a troca nas sequências de genes. Após feita a recombinação, o material genético é dividido em gametas que então são combinados com os gametas do outro pai para formar novamente um cromossomo diplóide completo, gerando assim os novos indivíduos. Para organismos haplóides é feita apenas a combinação das sequências de cada pai.

Dentro da etapa de replicação do DNA podem ocorrer erros ou alterações influenciadas por fatores externos, gerando assim as mutações. Isso pode ser positivo, negativo ou não influenciar o resultado final, mas é um outro mecanismo que pode determinar a evolução das espécies.

Os genes definem as características dos indivíduos, mas também existe interação entre os genes, chamada de epistasia (*epistasis*), podendo um par de genes mascarar ou modificar a característica final de outro par de genes. (KLUG et al., 2011). Isso é importante do ponto de vista do algoritmo genético pois nem sempre a melhor avaliação estará associada a um único parâmetro, podendo estar associado a uma combinação de parâmetros.

Mendel ainda definiu o conceito de dominância-recessividade em organismos diplóides, assumindo que cada característica é controlada por pares de genes, sendo recebidos um de cada pai. Assim um dos alelos do gene é dominante sobre o outro apresentando a característica final no indivíduo, e outro alelo será considerado recessivo.

Combinando indivíduos que melhor se adaptaram as condições de sobrevivência, provavelmente deve gerar novos indivíduos que tenha características ainda melhores que seus antecessores. O outro caso pode acontecer também e o novo indivíduo ter piores condições de viver, e assim o processo de seleção natural levará esse espécime a extinção

favorecendo outros que se sobressaíram na próxima geração.

A evolução então é um processo adaptativo onde através de mutações e recombinação entre os indivíduos, vão surgindo novas gerações que devem apresentar cada vez mais seres que se adaptam cada vez melhor ao ambiente.

1.3 Surgimento do algoritmo genético

Os algoritmos evolucionários vem sendo estudados a um longo tempo, desde da década de 40 que os cientistas se inspiram na natureza para criar os primeiros passos para a inteligência artificial, passando pela década de 50 onde começam os estudos sobre sistemas adaptativos para gerar soluções candidatas para problemas de difícil solução. Na década de 60 Rechenberg desenvolveu as estratégias evolucionárias (*evolutionary strategies*) usando cromossomos compostos de números reais para estudos com aerofólios. Também apareceu a programação evolucionária, usando estruturas de pequenas máquinas de estados para resolver determinadas tarefas, que evolui para programação genética onde pequenos programas passam a ser as soluções candidatas. Existiram outros trabalhos sobre algoritmos evolucionários, programação evolucionária e algoritmos genéticos nas áreas de otimização e aprendizado de máquina, porém foram os trabalhos de Holland nas décadas de 60 e 70 que consolidou os algoritmos genéticos. (MITCHELL, 1996; LINDEN, 2008)

Alguns autores como MITCHELL, JACOBSON, KWONG; MAN; TANG entre outros citam Holland como criador do algoritmo genético (*Genethic Algorithm* ou GA) em 1975 no livro "*Adaptation in Natural and Artificial Systems*". Nele Holland formaliza uma estrutura para sistemas adaptativos, e encaixa o GA como uma abstração para a evolução biológica.

O objetivo de Holland não era fazer algo específico mas sim analisar os sistemas adaptativos e criar uma solução computacional que simulasse os processos encontrados nos sistemas de adaptação natural. Para isso cria uma abstração da evolução na biologia usando uma estrutura formal teórica que poderia atender diversos sistemas evolutivos e não somente o GA. Na formulação do GA por ele, os cromossomos seria representações binárias e utilizou os operadores de crossover, inversão e mutação inspirados na genética. O fato de ter usado cromossomos binários serviu de influência para vários outros estudos que se seguiram, mas será exposto no [Capítulo 2](#) que existem outras formas de representar os cromossomos que dependem exclusivamente do problema a ser estudado.

O algoritmo genético geral proposto por Holland se baseia em uma população de cromossomos com alelos '0' e '1' e com os operadores de crossover, mutação e inversão. As evoluções seguem os seguintes passos:

1. Seleção de um cromossomo na população de forma estocástica baseada nas avaliações

de todos os cromossomos

2. Aplicações dos operadores genéticos sobre uma cópia do indivíduo selecionado em 1.
3. Seleção de outro cromossomo de forma aleatória com probabilidade igual para todos a ser substituído pelo novo cromossomo gerado em 2
4. Avaliar o novo cromossomo
5. Retonar ao 1

Essa é uma descrição bem resumida sobre o algoritmo e serve de base para as derivações. Por exemplo no caso do uso de um operador de crossover no Item 2, há a necessidade de seleção de outro cromossomo para formar um par e assim gerar uma nova estrutura.

Avaliando o algoritmo, temos alguns itens básicos que estarão presentes. A codificação do cromossomo para representar os parâmetros das soluções para os problemas. Uma população de cromossomos que serão avaliados simultaneamente durante as iterações do processo. A função de avaliação, que baseada nos parâmetros de cada estrutura irá fornecer uma forma de comparar os diversos elementos presentes na população. Os operadores genéticos que serão usados, sendo que podem ser combinados de diversas formas.

De acordo com MITCHELL existem 3 operadores para o algoritmo mais simples:

Seleção seleciona cromossomos favorecendo os que tem melhor função de avaliação

Crossover também chamado de recombinação, combina dois cromossomos na geração de um novo

Mutação Altera de forma aleatória alguns alelos dos cromossomos por exemplo em uma codificação binária seria alterar um dos bits do parâmetro.

No Capítulo 2 será explorado com mais detalhes cada um dos operadores. Holland também descreve um operador de inversão, que apesar de garantir uma melhor diversidade genética impõe uma carga computacional grande para os ganhos efetivos, e nos trabalhos mais recentes se tornou um operador quase nunca usado.

Outro teoria que Holland formula sobre os GAs é em relação ao processo de paralelismo intrínseco, sobre a forma do algoritmo trabalhar com esquemas em vez de testar indivíduos específicos da população. GOLDBERG aborda esse tema com a hipótese dos blocos de construção (*Building block hypothesis*).

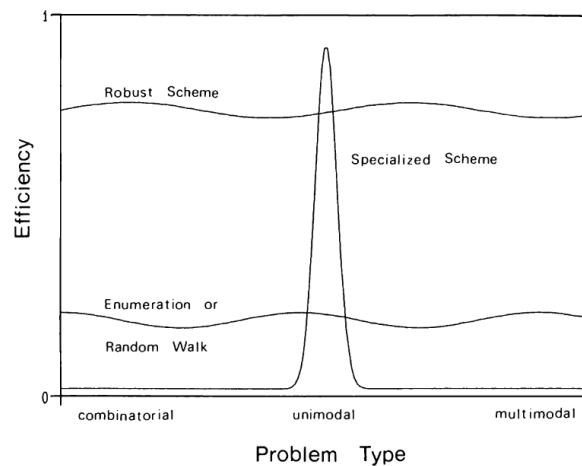


Figura 3 – Exemplo para NFL - (GOLDBERG, 1989)

1.4 Teorema da inexistência do almoço grátis

Existe um compromisso entre a eficiência de um algoritmo e sua robustez, isso é, ele pode ser muito eficiente para determinados problemas mas serem ineficientes em outros. Isso implica que não existe um algoritmo que seja eficiente para todos os problemas, e então é razoável supor que algoritmo genético tenha melhor eficiência em determinadas situações. Essa falta de um algoritmo universal é comparado por SPALL a busca de uma agulha no palheiro sem nenhum dado sobre sua posição. Para essa situação a busca aleatória seria um dos melhores métodos de busca e em média nenhum outro seria mais eficiente.

Essa falta de um algoritmo universal para solução dos problemas de otimização parte do teorema da inexistência do almoço grátis (NFL - *No-Free-Lunch Theorem*) proposto por Wolpert. O NFL ainda afirma que todos os algoritmos de busca tem em média o mesmo desempenho (LINDEN, 2008). O que de fato deve ser chamado a atenção é que apesar do apelo do algoritmo comparado com a evolução da espécies, é que se houver um método específico para determinado problema, definitivamente ele será mais eficiente que o GA.

Na Figura 3 GOLDBERG mostra que no espectro de problemas de busca e otimização, um método especializado em determinado problema sempre terá a maior eficiência e que os métodos aleatórios e enumerativos apesar de serem considerados menos eficientes podem ter desempenho melhor que o método específico em um espectro mais amplo. O método robusto representa um algoritmo idealizado para solução dos problemas em todas as variedades de problemas. Seria interessante ter um algoritmo robusto, mesmo abandonando o pico de eficiência pois assim atenderia uma variedade de problemas. Como tal algoritmo não é conhecido o que pode ser feito é usar métodos híbridos combinando mais de uma técnica.

2 Algoritmos Genéticos

O algoritmo genético (GA) parte de uma população de *cromossomos* que representam as soluções candidatas para o problema de otimização. Cada uma das soluções é avaliada de acordo com critérios inerentes do problema, para posteriormente serem selecionados e combinados de forma a criar novas soluções candidatas.

Já é possível perceber uma das vantagens do algoritmo, que ele irá fazer uma avaliação direta das soluções e de forma paralela. Por exemplo em problemas NP-Completo onde são difíceis de se obter soluções numéricas eficientes, mas é possível testar soluções em tempo polinomial, o algoritmo se torna atrativo com essas características.

2.1 Seleção

2.2 Crossover

2.3 Mutação

2.4 Esquemas

2.5 Fundamentos teóricos

3 Implementação do algoritmo genético

Para implementação do algoritmo foi utilizado a linguagem de programação JAVA de forma a aproveitar o paradigma de orientação a objetos e reaproveitamento de código. A estrutura do programa permite com classes básicas formar a lógica do algoritmo e com classes derivadas especializar o algoritmo para especificações de cada problema. Dessa forma com pouco código extra sendo feito é possível reaproveitar as classes já definidas, usando o polimorfismo inerente a linguagens OO, é possível reescrever e acrescentar as partes necessárias para embutir conhecimento específico no problema.

3.1 Modelo de Ising

—Explicação simples sobre o modelo de Ising

O algoritmo genético simples será usado para apresentar uma solução para modelo de Ising em uma matriz 10×10 , com cada nó (spin) representado pelos possíveis valores -1 e 1 .

Sendo $\Lambda = [-5, 5]^2$ as posições da matriz, e $\sigma_t \in \{-1, 1\}$ representando os valores de cada spin em determinada posição da matriz, e seja $(\sigma_t)_{t \in \Lambda} = \sigma$ um conjunto de valores da matriz.

A função de avaliação que deve ser minimizada é dada por $H(\sigma) = -\sum_{\langle t, t' \rangle} \sigma_t \sigma_{t'}$ onde t e t' representam dois spins vizinhos.

3.2 Codificação

A classe cromossomo será usada para representar cada solução para o problema e conterá um vetor de valores inteiros de tamanho 100, representando assim a matriz 10×10 , alinhando uma linha da matriz na sequência da outra no vetor. Assim o nosso genótipo é uma cadeia de inteiros, onde o fenótipo será a matriz correspondente quebrando o vetor em linhas de tamanho 10.

Exemplo da relação genótipo x fenótipo para uma matriz 3×3 :

$$\text{cromossomo} = [-1, 1, 1, 1, -1, 1, 1, 1, -1]$$

$$\text{fenótipo} = \begin{bmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix}$$

3.3 Seleção e avaliação

3.4 Crossover e mutação

3.5 Resultados

4 Conclusão

Referências

COELLO, C. C.; LAMONT, G. B.; VELDHUIZEN, D. A. van. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. New York, NY: Springer, 2007. ISBN 978-0-387-36797-2. Disponível em: <<https://www.amazon.com/Evolutionary-Algorithms-Multi-Objective-Problems-Computation-ebook/dp/B00DZ75DNE?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=B00DZ75DNE>>. Citado 3 vezes nas páginas 9, 21 e 22.

GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. 1. ed. Reading, MA: Addison-Wesley Professional, 1989. ISBN 0201157675. Disponível em: <<https://www.amazon.com/Genetic-Algorithms-Optimization-Machine-Learning/dp/0201157675?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0201157675>>. Citado 3 vezes nas páginas 9, 26 e 27.

JACOBSON, B. K. L. *Genetic Algorithms in Java Basics*. New York, NY: Springer-Verlag GmbH, 2015. Disponível em: <https://www.ebook.de/de/product/25481443/lee_jacobson_burak_kanber_genetic_algorithms_in_java_basics.html>. Citado na página 25.

KLUG, W. S. et al. *Concepts of Genetics*. São Francisco, CA: Benjamin Cummings, 2011. ISBN 978-0-321-72412-0. Disponível em: <<https://www.amazon.com/Concepts-Genetics-William-S-Klug/dp/0321724127?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0321724127>>. Citado 2 vezes nas páginas 9 e 24.

KWONG, S.; MAN, K.-F.; TANG, K.-S. *Genetic Algorithms*. London: Springer London, 2001. ISBN 1852330724. Disponível em: <https://www.ebook.de/de/product/3753808/sam_kwong_kim_fung_man_kit_sang_tang_genetic_algorithms.html>. Citado na página 25.

LINDEN, R. *Algoritmos Genéticos*. 2. ed. Rio de Janeiro: Brasport, 2008. ISBN 978-8574523736. Citado 3 vezes nas páginas 22, 25 e 27.

MITCHELL, M. *An Introduction to Genetic Algorithms (Complex Adaptive Systems)*. 1. ed. Cambridge, MA: The MIT Press, 1996. ISBN 0262133164. Disponível em: <<https://www.amazon.com/Introduction-Genetic-Algorithms-Complex-Adaptive/dp/0262133164?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0262133164>>. Citado 2 vezes nas páginas 25 e 26.

SIVANANDAM, S.; DEEPA, S. N. *Introduction to Genetic Algorithms*. New York, NY: Springer, 2007. ISBN 978-3-540-73189-4. Disponível em: <<https://www.amazon.com/Introduction-Genetic-Algorithms-S-N-Sivanandam/dp/354073189X?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=354073189X>>. Citado na página 23.

SPALL, J. C. *Introduction to Stochastic Search and Optimization*. Hoboken, NJ: John Wiley & Sons, 2003. ISBN 0471330523. Disponível em: <https://www.ebook.de/de/product/2876128/spall_stochastic_optimization.html>. Citado na página 27.

Apêndices

Anexos

