

Sudoku, Aplicando Algoritmos para Tratamento de uma Pipeline e Solução do Tabuleiro

Douglas L. P. Olinto¹, Lucas Cruvinel²

¹Inteligencia Artificial Aplicada – Instituto Federal de Goiás (IFG)
Caixa Postal 74.055-110 – Goiânia – GO – Brazil

dleonard@estudantes.edu.ifg.br

lucaseacruvinel@gmail.com

Abstract. *Sudoku, a widely recognized numerical puzzle, has historical roots dating back to the 18th century, but its modern form and global popularity emerged in the late 20th century, notably in Japan. Sudoku fundamentally presents a challenge of number combination and permutation within a 9x9 grid, subject to strict rules of numerical arrangement. This scientific article explores its historical origins, modern development, underlying mathematical principles, worldwide proliferation, complexity, and resolution methods, including brute force and the Hill Climbing algorithm. Despite its seemingly straightforward nature, Sudoku provides a captivating research field for puzzle enthusiasts and scholars, shedding light on the intricate interplay between mathematics and entertainment.*

Resumo. *O Sudoku, um quebra-cabeça numérico de renome mundial, tem raízes históricas que remontam ao século XVIII, mas sua forma moderna e popularização ocorreram nos últimos anos do século XX, notadamente no Japão. O Sudoku é essencialmente um desafio de combinação e permutação de números em uma grade 9x9, regido por regras estritas de organização numérica. Este artigo científico explora suas origens históricas, desenvolvimento moderno, fundamentos matemáticos, disseminação global, complexidade e métodos de resolução, incluindo força bruta e o algoritmo de Hill Climbing. O Sudoku, apesar de sua aparente simplicidade, oferece um campo de pesquisa cativante para entusiastas de quebra-cabeças e acadêmicos, destacando a intrincada relação entre a matemática e o entretenimento.*

1. Introdução

O Sudoku, um desafio numérico amplamente reconhecido e adorado, possui raízes históricas que mergulham profundamente na história, remontando ao século XVIII. No entanto, sua forma contemporânea e ascensão à fama ocorreram nas últimas décadas do século XX, com destaque para sua proliferação no Japão. O Sudoku é, essencialmente, um intrigante enigma de combinação e permutação de números dispostos em uma grade de 9x9, sob rígidas regras que ditam a organização desses números. Este artigo científico tem como objetivo lançar luz sobre as origens históricas do Sudoku, seu desenvolvimento moderno, os fundamentos matemáticos subjacentes, a disseminação global desse quebra-cabeça, sua resolução e complexidade, além de abordar em detalhes métodos de resolução, incluindo a abordagem de força bruta e outras.

2. Origens Históricas

O Sudoku, em sua forma moderna, está intimamente relacionado com o trabalho do arquiteto suíço Leonhard Euler, que publicou em 1783 um trabalho intitulado "Lugar

das Tabuletas Mágicas" (ou "Lieux à Tabulées Magiques" em francês). Embora esse trabalho não apresente o Sudoku da maneira como o conhecemos atualmente, é uma peça fundamental na história do quebra-cabeça. Euler explorou problemas relacionados à quadrados mágicos, que são arranjos de números em uma grade onde a soma dos números em cada linha, coluna e diagonal é a mesma. Este conceito se tornou um precursor do Sudoku e evidencia como a matemática desempenhou um papel significativo na criação do quebra-cabeça.

3. Desenvolvimento Moderno

A forma moderna do Sudoku ganhou destaque no Japão no final do século XX. Nos anos 1980, o jornal japonês Nikoli reintroduziu o quebra-cabeça, nomeando-o "Su Doku", que é uma abreviação de sua descrição japonesa "Sūji wa dokushin ni kagiru", que significa "os números devem ocorrer apenas uma vez". A Nikoli desempenha um papel fundamental na popularização do Sudoku e é creditada por seu renascimento. A partir desse ponto, o Sudoku se espalhou globalmente e se tornou uma paixão para entusiastas de quebra-cabeças em todo o mundo.

4. Matemática por Trás do Sudoku

O Sudoku é um quebra-cabeça intrincado que envolve conceitos matemáticos complexos. Ele é essencialmente um problema de combinação e permutação de números em uma grade 9x9, com regras rigorosas sobre como esses números devem ser organizados. A matemática subjacente inclui conceitos de teoria dos grafos, combinatória e álgebra. Para uma análise matemática mais aprofundada do Sudoku, é recomendável o livro "The Mathematics of Sudoku" de Jason Rosenhouse e Laura Taalman.

5. Popularização Global

O Sudoku se tornou um fenômeno global nas primeiras décadas do século XXI, com a publicação em massa de livros de quebra-cabeças, aplicativos de resolução e competições. Uma ampla variedade de variações e tamanhos de Sudoku surgiram ao longo do tempo, adicionando complexidade ao desafio e expandindo ainda mais o campo de pesquisa matemática relacionado.

6. Resolução e Complexidade

O Sudoku é um problema NP-completo, o que implica que encontrar a solução para um tabuleiro pode ser computacionalmente desafiador. No entanto, existem algoritmos eficazes para resolvê-lo. Pesquisadores dedicaram esforços consideráveis para explorar a complexidade do Sudoku, como evidenciado no estudo intitulado "The Complexity of Sudoku" de Frazer e Logie. Este estudo analisa a complexidade computacional do Sudoku em detalhes, destacando as estratégias e algoritmos que permitem resolver os quebra-cabeças.

7. Métodos de Resolução Aplicados por Algoritmos

Um algoritmo nada mais é que uma sequência de instruções ou comandos realizados de forma sistemática com a finalidade de resolver um problema ou executar uma determinada tarefa. Ou seja, é criado para resolver “problemas”, com instruções bastante simples e exatas.

7.1 Resolução por Força Bruta

Uma das abordagens mais diretas para resolver um Sudoku é a força bruta. Nesse método, todos os espaços vazios do tabuleiro são preenchidos testando todas as combinações possíveis de números até encontrar uma que satisfaça todas as regras do Sudoku. Embora eficaz, esse método pode ser demorado quando o tabuleiro é complexo.

No nosso caso, implementamos uma abordagem semelhante à força bruta, pois, em vez de testar todas as combinações possíveis de uma só vez, primeiro são avaliados os espaços vazios individualmente, seguido da avaliação de todas as combinações possíveis para preencher cada uma dessas posições. Por meio dessa abordagem, ao introduzir apenas essa simples condição, conseguimos reduzir mais de 50% dos casos de teste, logo, reduzindo o custo computacional e aumentando a velocidade de resposta.

7.2 Resolução por Recozimento Simulado

O algoritmo de recozimento simulado é uma estratégia de otimização que encontra inspiração no processo de recozimento de materiais, onde um material é aquecido e depois resfriado gradualmente, visando alcançar um estado de menor energia. No nosso cenário, aplicamos o recozimento simulado para otimizar uma função de custo, na qual essa função representa a contagem de elementos únicos nas linhas ou colunas. Em outras palavras, se todas as linhas, colunas e quadrantes de um jogo de Sudoku forem distintos, consideramos o jogo completo e a função de custo atinge seu mínimo. Assim, o recozimento simulado se empenha em encontrar essa configuração mínima ao explorar e ajustar as soluções iterativamente.

Para encontrar esses mínimos, o algoritmo começa com uma solução inicial aleatória e, em seguida, realiza iterações para tentar melhorá-la. Durante cada iteração, o algoritmo aceita soluções piores com uma certa probabilidade, o que permite escapar de mínimos locais e explorar o espaço de busca de maneira mais abrangente. Com o tempo, a probabilidade de aceitar soluções piores diminui, simulando o processo de resfriamento, e o algoritmo converge para uma solução próxima da ótima.

O recozimento simulado é uma técnica flexível que encontra aplicação em uma ampla diversidade de problemas de otimização. Sua efetividade está diretamente relacionada à seleção criteriosa de parâmetros fundamentais, tais como a taxa de resfriamento e a função de custo. A determinação desses valores ideais envolveu uma exploração manual, na qual foram experimentados diversos conjuntos de parâmetros até que fossem identificados aqueles que permitissem a resolução do problema no menor tempo possível, sem comprometer a precisão da solução.

7.3 Comparação dos algoritmos de solução.

Utilizando o AWS Sagemaker, que será discutido em detalhes na seção 8.1, e aproveitando uma base de dados abrangente, conduzimos uma análise comparativa da precisão e do tempo de execução dos algoritmos mencionados anteriormente. Para essa finalidade, empregamos uma base de dados sólida adquirida no site Kaggle, composta por um milhão de quebra-cabeças de Sudoku, cada um acompanhado de sua solução correspondente.

Nesse contexto, escolhemos aleatoriamente 10.000 quebra-cabeças de Sudoku, com o objetivo de que ambos os algoritmos os resolvessem. Durante esse processo, registramos o tempo necessário e, por fim, comparamos os resultados obtidos com as soluções existentes na base de dados. Os resultados estão disponíveis nas Figuras 1 e 2, exibidas adiante.

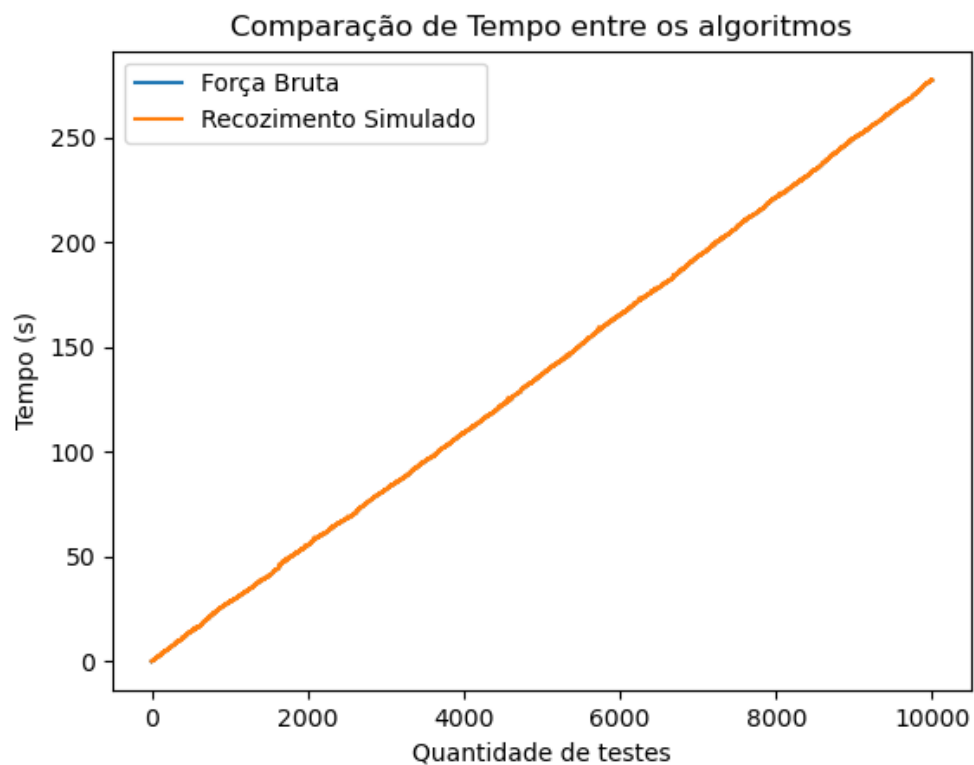


Figura 01. Comparação de tempo entre os algoritmos

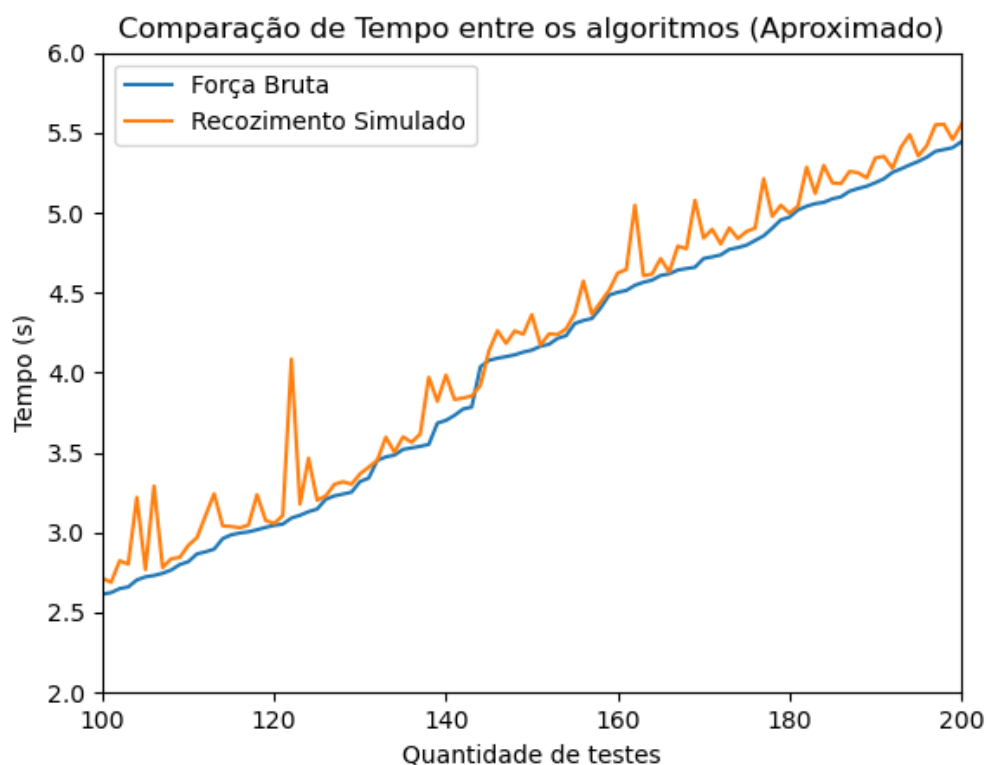


Figura 02. Comparação de tempo entre os algoritmos (Resolução ampliada)

Conforme observado, a discrepância no desempenho por quebra-cabeça é mínima, mas, de maneira consistente, o algoritmo de força bruta apresentou tempos de resposta mais ágeis. No desfecho da análise, contabilizamos as respostas corretas, dividindo-as pelo total de quebra-cabeças, onde ambos os algoritmos atingiram uma taxa de acerto de 100%.

8. Computação em nuvem

A inclusão da computação em nuvem no âmbito deste projeto amplia consideravelmente as possibilidades de propriedades que podem ser aplicadas em conjunto com o jogo de Sudoku. Inicialmente, dois objetivos primordiais foram delineados para a utilização na nuvem: implementação de aprendizado de máquina e distribuição das ferramentas desenvolvidas.

8.1 Amazon Sagemaker

Para realizar a implementação dos algoritmos e da função de otimização descritos na seção 7.1 e 7.2, utilizamos a ferramenta Sagemaker, uma plataforma repleta de funcionalidades. No entanto, devido às limitações impostas pela plataforma AWS

Academy, utilizamos o Sagemaker apenas como um ambiente de desenvolvimento de códigos, em conjunto com o JupyterLab. Nos notebooks do JupyterLab, começamos por criar funções que permitiam a leitura e verificação de arquivos no formato '.csv', contendo os desafios de Sudoku a serem resolvidos. Em seguida, prosseguimos com a implementação dos algoritmos e a comparação de suas capacidades em termos de tempo de execução e precisão. Por fim, realizando uma demonstração gráfica dessa diferença.

8.2 Amazon S3 e EC2

O Amazon S3 é um serviço de armazenamento em nuvem oferecido pela Amazon Web Services. Ele fornece uma infraestrutura escalável, durável e altamente disponível para armazenar e recuperar dados de maneira eficiente. O S3 permite que usuários armazenem e gerenciem uma vasta quantidade de dados, desde arquivos simples até grandes conjuntos de dados, e oferece recursos de segurança, controle de acesso, redundância de dados e uma variedade de classes de armazenamento para atender a diferentes necessidades de armazenamento e orçamentos.

Dentro do nosso projeto, estabelecemos um bucket denominado "sudoku-lucas-cruvinel", desempenhando um papel essencial ao armazenar a entrada do usuário e servir como intermediário na execução do AWS Lambda. Contudo, devido a limitações impostas pelo AWS Academy e com o objetivo de proporcionar uma experiência mais amigável aos usuários, integramos o EC2 para hospedar um servidor web de pequena escala. Esse servidor web atua como uma interface intermediária entre o usuário e o S3, permitindo ao usuário realizar tanto o upload quanto o download de conteúdo a partir de pastas pré-definidas dentro do mencionado bucket.

8.3 Amazon Lambda

O Lambda é um serviço *serverless* de automatização de tarefas, que permite a execução de um código em resposta a eventos, como uploads de arquivos, chamadas de API ou modificações em bancos de dados. O serviço é altamente flexível, fornece escalabilidade automática e cobra apenas pelos recursos computacionais usados durante a execução do código.

Neste projeto, desenvolvemos duas funções Lambda: "sudoku-brute-force" e "verificar-entrada-sudoku". Ambas são acionadas quando um arquivo com extensão ".csv" é inserido nas pastas de entrada correspondentes. Além disso, configuramos um limite máximo de execução de um minuto para ambas, garantindo tempo suficiente para a conclusão de suas tarefas. Por fim, adicionamos camadas adicionais que possibilitam que os códigos façam uso de importações de bibliotecas robustas, como pandas e numpy, aprimorando a capacidade das funções Lambda.

8.4 Amazon CloudWatch

O Amazon CloudWatch é um serviço que rastreia e reúne dados de desempenho e registros de aplicativos, recursos e serviços na nuvem. Ele fornece informações cruciais

para supervisionar, diagnosticar e solucionar problemas relacionados à infraestrutura e aplicativos na AWS, contribuindo para garantir a confiabilidade e o desempenho dos sistemas na nuvem. Devido a essa importância, ele foi amplamente utilizado nesse projeto, sendo fundamental para a configuração, implementação e depuração de códigos no AWS Lambda.

8.5 WorkFlow

O fluxo deste projeto está condensado na Figura 03, e pode ser visualizado logo abaixo.

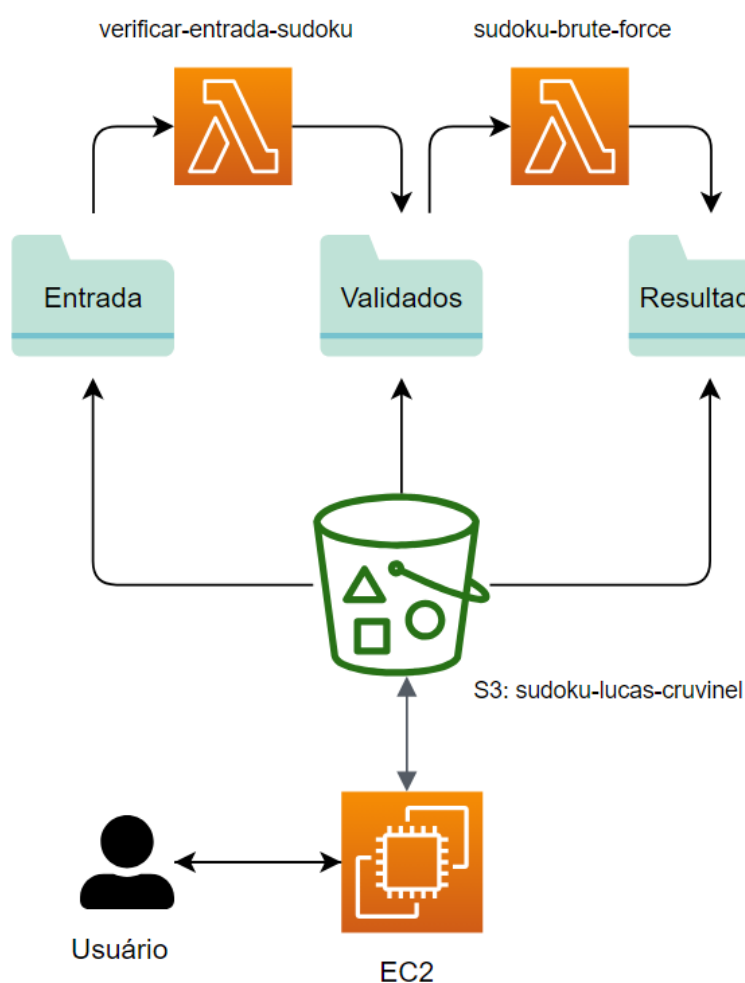


Figura 03. Demonstração do WorkFlow.

Na ilustração, os usuários acessam nosso servidor web hospedado no EC2, possibilitando o upload de um arquivo ".csv" contendo um desafio de Sudoku. O EC2 é encarregado de armazenar esse arquivo no bucket "sudoku-brute-force" do S3, mais precisamente em uma pasta designada como "entrada".

Quando um arquivo é adicionado a essa pasta e, se o arquivo estiver no formato ".csv", um evento é acionado e encaminhado para a função Lambda "verificar-entrada-sudoku". Essa função é responsável por ler o arquivo no S3 e realizar uma validação dos dados para assegurar que se trata de um quebra-cabeça de Sudoku viável. Todas as verificações efetuadas são registradas para criar um arquivo de log da operação. Em última instância, se a entrada for confirmada como válida, ela será movida para a pasta "validados" junto com o arquivo de log correspondente. Caso a entrada não cumpra os requisitos, somente o arquivo de log será arquivado, para que o usuário verifique os erros.

No momento em que esse arquivo chega na pasta "validados", um novo evento é enviado para o Lambda, dessa vez para a função "sudoku-brute-force", que lê esse novo arquivo utilizando o pandas e aplica o algoritmo descrito na seção 7.1. Caso o resultado tenha sido descoberto, ele é salvo também como ".csv" na pasta "resultados". Note que o usuário tem acesso a todas essas pastas através do EC2, logo podendo realizar o download do resultado no fim dessas operações.

9. Conclusão

A história do Sudoku é uma jornada fascinante que combina matemática, lógica, cultura e entretenimento. Sua evolução desde os quadrados mágicos de Euler até os desafios modernos que enfrentamos hoje é um exemplo intrigante de como os conceitos matemáticos podem se manifestar na vida cotidiana de maneira surpreendente e cativante. Além disso, a matemática subjacente ao Sudoku, juntamente com os métodos de resolução, representa um campo de pesquisa rico e desafiador para entusiastas da matemática e entusiastas de quebra-cabeças. O Sudoku continua a exercer um fascínio duradouro que transcende fronteiras culturais e geográficas.

A abordagem de força bruta a qual executamos apresenta como uma estratégia direta e minuciosa para resolver quebra-cabeças de Sudoku. Nesse método, cada espaço vazio do tabuleiro é preenchido de forma sistemática, submetendo-se a uma exploração exaustiva de todas as combinações concebíveis de números, um por um, até que uma solução que atenda de maneira integral e impecável às regras estritas do Sudoku seja alcançada. Essa abordagem implica que, para cada espaço vazio, o solucionador testa todas as possíveis alternativas, de 1 a 9 (ou 1 a N, no caso de variantes maiores do Sudoku), garantindo que a solução final seja congruente com as regras do jogo.

Apesar da implementação de um algoritmo mais sofisticado, como o recozimento simulado, ele não conseguiu superar seu concorrente neste cenário. No entanto, é importante destacar que os parâmetros utilizados foram determinados por meio de testes manuais e a aplicação de uma heurística para otimizar esses parâmetros pode potencialmente alterar significativamente esse resultado.

10. Referencias

"Amazon Web Services – Arquiteturas Multicamadas com Servidor da AWS." https://d0.awsstatic.com/whitepapers/International/pt/AWS_Serverless_Multi-Tier_Architectures_PT. [Online: accessed 2023-10-21]

“1 million Sudoku games.”<https://www.kaggle.com/datasets/bryanpark/sudoku>. [Online; accessed 2023-10-21].

Berthier, Dominique. (2005). "Sudoku: Le Jeu." Paris: Da Capo Press.

Felgenhauer, Bertram, & Jarvis, Frazer. 2005. Enumerating possible Sudoku grids.

Klarreich, Erica. (2005). "Sudoku Squared." *Science*, 310(5752), 981. DOI: 10.1126/science.310.5752.981

Rosenhouse, Jason, & Taalman, Laura. (2014). "The Mathematics of Sudoku." New Jersey: Princeton University Press.

Smith, A. and Jones, B. (1999). On the complexity of computing. In *Advances in Hecht, Jeff. (2006). "The Sudoku Effect: Universities worldwide launch research into grids." New Scientist*, 189(2533), 22-23.

“Sudoku.” <https://pt.wikipedia.org/wiki/Sudoku>. [Online; accessed 2023-10-17].