

# Aplicação de Evolução Diferencial para Problema do Sudoku

William Sdayle Marins Silva<sup>1</sup>

**Resumo**— Neste artigo serão tratados métodos de evolução utilizando algoritmos genéticos para solução do problema do Sudoku, usando tipos distintos de esquemas de Evolução Diferencial buscando diversificar os indivíduos evoluídos buscando não mante-los em ótimos vales.

**Palavras-chave:** Evolução Diferencial, Sudoku, Algoritmos Genéticos.

**Abstract**— In this paper will be trated methods of evolution using genetic algorithms to solve the problem of Sudoku, using multiple types of schema of differential evolution searching diversify the individuals got and not keeping them in valleys.

**Keywords :** Differential Evolution, Sudoku, Genetic Algorithms.

## I. INTRODUÇÃO

Jogos de computadores representam um campo de pesquisa importante da área de inteligência artificial [1]. Boa parte das pessoas se identificam com o hábito de utilizar games para distração. Com o avanço tecnológico, os jogos eletrônicos evoluíram ao seu lado [2]. Porém alguns ainda preferem jogos de tabuleiro como xadrez, dama, etc. [3] afirma que jovens podem até desenvolver raciocínio lógico com jogos similares ao Sudoku. O objetivo do presente artigo é abordar métodos de evolução com algoritmos genéticos para solucionar o jogo Sudoku, abordando esquemas de DE (*Differential Evolution*) e realizar avaliações sobre quais métodos os indivíduos podem convergir de maneira mais rápida. Aplicando diferentes métodos de mutação com DE, métodos de *Crossover* para permutar os indivíduos e avaliar o fitness dos indivíduos de acordo com as regras do jogo Sudoku.

### A. Definição do jogo Sudoku

Sudoku é jogado em uma matriz 9x9, dentro da matriz existem células menores que terão um valor específico de 1 até 9. A matriz 9x9 é subdividida em 9 matrizes menores 3x3, uma ao lado da outra [4]. A ideia principal do jogo é conseguir adicionar um número a cada célula da matriz obedecendo a 3 condições:

- 1) Dentro de uma submatriz não pode haver nenhum valor repetido
- 2) Nenhum valor pode se repetir na mesma linha
- 3) Nenhum valor pode se repetir na mesma coluna

Valores iguais nunca podem estar no mesmo eixo x e y, nem na mesma submatriz, conforme mostra a figura 1.

## II. FUNDAMENTAÇÃO

Nesta seção serão tratados assuntos base para o artigo, tais como evolução dos indivíduos em determinados cenários, como ocorrem as evoluções. Apresentando como funciona um algoritmo genético baseado na evolução natural humana, algoritmos que buscam

4		1	9	8				
8	5			7				
	2				5	1		4
	1	6						
3			8		2			
					6			
3	4					8		
8		9	4	3				

6	4	3	5	1	7	9	2	8
8	1	5	3	2	9	7	4	6
2	9	7	8	6	4	3	1	5
9	2	8	1	7	5	6	3	4
4	7	1	6	3	2	5	8	9
5	3	6	9	4	8	1	7	2
7	5	9	4	8	3	2	6	1
3	6	4	2	5	1	8	9	7
1	8	2	7	9	6	4	5	3

Fig. 1. Ilustração de como o jogo é iniciado e como deve acabar, Fonte - [5].

solucionar fórmulas matemáticas e algoritmos de otimização de problemas. Na seção 2.1 serão tratados assuntos como evolução natural humana.

### A. Evolução Genética

Para [6] a evolução é uma ideia bela e de fácil compreensão, afirmando que evolução significa mudança, na forma e no comportamento dos organismos ao longo das gerações. Basicamente, evolução trata-se realmente de alterações, porém baseadas em uma informação primária. Na biologia, de acordo com [7], mutações são quaisquer alterações da sequência de nucleotídeos em um organismo que possuam frequência inferior a 1. Essas alterações podem ser causadas por erros de cópia do material durante a divisão celular no próprio indivíduo ou ainda transmitidas pelos ancestrais. Afirmando também que recombinação *Crossing over* é o fenômeno responsável por recombinar o material genético a partir de cromossomos pais conforme mostra na figura 2.

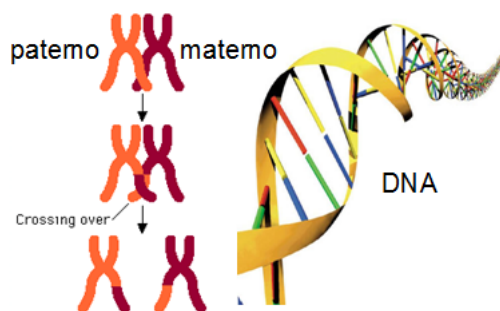


Fig. 2. Ilustração de como é ocorrido a recombinação de cromossomos humanos, Fonte - [7].

### B. Inteligência Artificial

Inteligência artificial é uma das ciências mais recentes, alguns autores como [8] tratam-as da seguinte forma:

- 1) Sistemas que pensam como seres humanos
- 2) Sistemas que atuam como seres humanos
- 3) Sistemas que pensam racionalmente
- 4) Sistemas que atuam racionalmente

De acordo com [8] todas as quatro dimensões para o estudo da inteligência artificial têm sido seguidas. Afirmando que a inteligência artificial é um ramo da Ciência da Computação cujo interesse é fazer com que os computadores pensem ou se comportem de forma inteligente.

<sup>1</sup>William Sdayle Marins Silva, Mestrado em Computação Aplicada, Universidade Tecnológica Federal do Paraná, Cornélio - Procópio - PR, 2019, williamsdy30@gmail.com

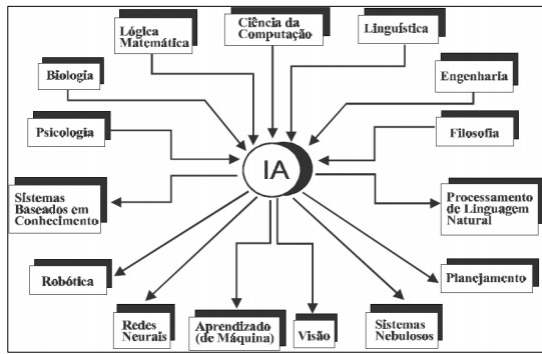


Fig. 3. Áreas de aplicação para Inteligência Artificial, Fonte - [8].

### C. Heurística e Meta-Heurística

De acordo com [9] heurísticas são procedimentos que tratam problemas de otimização sem dispor de garantias teóricas, nem que a solução ótima exata seja obtida, tratando heurística como procedimento de busca de boas soluções. Tratando-se de metaheurística [10] define como estratégias de alto nível que guiam uma heurística subjacente, buscando aumentar o desempenho desta, tendo como objetivo principal não se manter em ótimos locais.

### D. Evolução no meio da computação

Baseado na evolução humana, a evolução de indivíduos ocorre de maneira similar, buscando propagar os melhores indivíduos para melhorar sua geração. Para [11], algoritmos genéticos imitam o processo de evolução, se baseando no processo de sobrevivência proposto por Darwin. Em um processo regido por seleção natural, populações competem umas com as outras para converterem-se em progenitores, em função de um valor chamado ajuste. Depois de selecionados os progenitores, aplicam-se uma série de funções denominadas operadores genéticos: mutação, inversão e cruzamento (*Cross over*).

### E. Algoritmos Genéticos

Os Algoritmos Genéticos trabalham com uma população de soluções iniciais, chamadas cromossomos, que através de diversas operações são evoluídas até que cheguem a uma solução que melhor atenda a algum critério específico de avaliação [12]. De acordo com [13] um ciclo do GA consiste nos seguintes estágios:

- 1) Criação de uma população de soluções potenciais;
- 2) Codificação dos indivíduos;
- 3) Avaliação dos indivíduos;
- 4) Seleção dos indivíduos mais capazes;
- 5) Geração da nova população através de manipulações genéticas.

O primeiro passo é codificar os indivíduos, sendo este o mais importante, que está diretamente ligado ao problema para [14]. Podem ser utilizadas codificações como necessário [15], podendo cada gene carregar uma informação binária, um valor inteiro, um *string* ou até mesmo valores reais, conforme a necessidade. Cada gene é uma informação, definidas por um *array*, que formam os cromossomos [15].

Posteriormente é necessário a criação da população, esse passo pode ser definido como for indagado, porém o tamanho da população interfere diretamente no resultado obtido pelos indivíduos. Quanto maior o número de indivíduos, mais difícil é de se chegar a uma solução ótima [16].

Após gerar a população de acordo com a codificação escolhida, o próximo passo é avaliar os indivíduos (cromossomos) gerados, para começar a utilizar os processos genéticos de evolução a partir da população.

A função de fitness é de extrema importância para o processo evolutivo, porque a partir dela é possível saber o quão bom ruim o indivíduo é. A função de avaliação estará ao lado do problema a ser resolvido [17].

Para [14], o próximo passo é a recombinação dos cromossomos, nesse processo, dois pares de pais são escolhidos. No momento da escolha, existem alguns algoritmos que facilitam a escolha e não "menosprezam" os indivíduos mais fracos escolhendo assim apenas os mais fortes. Dois possíveis métodos são o roleta e o torneio. Para [14], trata o algoritmo por torneio como, uma escolha de 3 indivíduos aleatórios e o com a maior aptidão é selecionado, já o método roleta baseia-se em um ranking de o quão bom o indivíduo é perante a população inteira, o método roleta é eficiente, porém prejudica os indivíduos mais fracos, já o método por torneio é baseado nos melhores indivíduos possíveis, habilitando espaço para aqueles que são "piores". Existem algoritmos para melhorar ainda a seleção dos indivíduos, aplicando regras de normalização para evitar um super indivíduo. Após a seleção do indivíduo é realizado o processo de reprodução, conhecido como *cross over*. [16], afirma que de acordo com Darwin, os melhores indivíduos, têm maior chance de serem reproduzidos, buscando melhorar a geração. O *cross over* é realizado de uma maneira simples é escolhido um método de corte, havendo um ou mais pontos de corte, onde são recombinados os genes que aparecem dentro do(s) ponto(s) de secção conforme mostrado na figura 4 e 5.

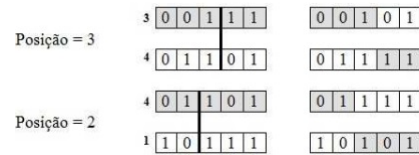


Fig. 4. Figura 4 - Recombinação de cromossomos com um único ponto de corte. Fonte - [16].

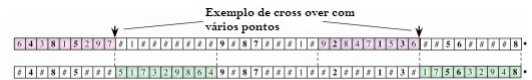


Fig. 5. Recombinação de cromossomos com um vários pontos de corte. Fonte - [5].

Após realizar o *cross over* o indivíduo passa pelo processo de mutação. Podendo ser em posição aleatória, posição definida por outro indivíduo no qual está sendo combinado, mutação em posição aleatória com valor aleatório ou mutação realizada por meio de DE (*Differential Evolution*).

### F. DE Differential Evolution

De acordo com [17] o algoritmo foi introduzido em 1995 por Storn e Price, afirmando que o algoritmo DE funciona de maneira simples e muito eficiente, sem a necessidade de vários parâmetros a serem ajustados durante a evolução, tornando-o mais simples de ser implementado. Pode-se considerar esses parâmetros como o tamanho da população, constante de cruzamento e o coeficiente de variação diferencial [18]. No momento da mutação, é utilizado um esquema, a facilitar a distinção das principais variantes, a anotação é a seguinte:

$$DE/x/y/z \quad (1)$$

Onde x é o vetor a ser mutado, y o número de vetores utilizados e z é o esquema utilizado. Três dos esquemas são citados a seguir:

$$DE/rand/1/bin \quad (2)$$

$$DE/rand/2 \quad (3)$$

$$DE/best/2 \quad (4)$$

Cada esquema realiza seu processo através de sua respectiva equação.

$$V^{i,G+1} = X_{\alpha}^G + F(X_{\beta}^G - X_{\gamma}^G) \quad (5)$$

$$V^{i,G+1} = X_{r_1,G} + F(X_{r_3,G} - X_{r_2,G}) + F(X_{r_2,G} - X_{r_4,G}) \quad (6)$$

$$V^{i,G+1} = X_{best,G} + F(X_{r_2,G} - X_{r_1,G}) + F(X_{r_4,G} - X_{r_3,G}) \quad (7)$$

Sendo  $X_{\alpha}^G$ , considerado por [19] como o vetor alvo e  $(X_{\beta}^G - X_{\gamma}^G)$  é o vetor diferencial criado a partir de dois indivíduos, a escolha dos indivíduos é feita de acordo com o esquema.

### III. METODOLOGIA

O escopo do trabalho é comparar os diferentes esquemas de evolução diferencial para resolução do problema do Sudoku, vendo que utilizando apenas o esquema (2) foi possível de acordo com [19]. Neste sentido, o presente capítulo tem como intuito apresentar os métodos adotados a fim de que fosse possível alcançar os objetivos. A figura 6 mostra a metodologia aplicada para realização do projeto.

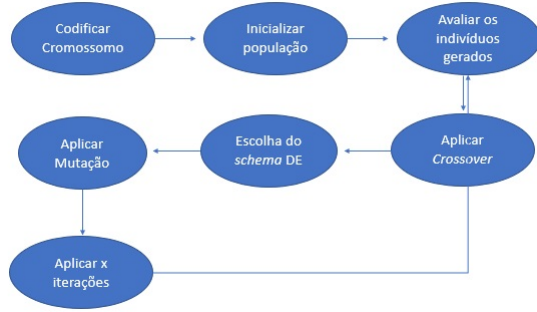


Fig. 6. Metodologia aplicada para o trabalho.

#### A. Codificação do Cromossomo

Para codificação do cromossomo foi aplicado um método de valores inteiros sendo de 1 até 9 para cada gene, formando um cromossomo de 81 posições. A cada 9 genes é representada uma submatriz dentro da matriz principal 9x9 do Sudoku, conforme a figura 7.

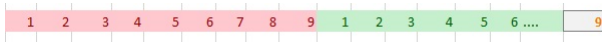


Fig. 7. Codificação do cromossomo para resolução do problema.

#### B. Inicialização da população

A população inicial foi criada com 20 indivíduos, cada um possuindo 81 genes em cada cromossomo. Com a finalidade de diversificar o problema, o número da população inicial é preenchida no momento da execução.

#### C. Avaliar os indivíduos gerados

A aptidão do indivíduo é calculada a partir dos elementos que ainda faltam em cada submatriz e em cada coluna, à medida que um valor não existe dentro da submatriz e no eixo y [14].

$$Faltantes = totaldefaltasnasubmatriz + totaldefaltasnacoluna \quad (8)$$

$$Fitness = (81 * faltantes) / 100 \quad (9)$$

A equação (9) é realizada para transformar em porcentagem os valores que o indivíduo acertou durante a execução.

#### D. Escolha dos Indivíduos

Para escolha dos indivíduos a serem modificados, foi implementado o método de torneio, de maneira que são escolhidos três indivíduos aleatórios e então é aplicado o torneio entre eles, é escolhido o indivíduo com maior aptidão para solucionar o problema.

#### E. Mutação e Aplicação do Crossover

A mutação é o foco do trabalho, explorar os esquema de DE para mutação de indivíduos e verificar qual esquema pode se tornar melhor utilizando taxas de mutação variáveis. Os esquemas utilizados foram os 3 esquemas citados a cima na seção II F (2), (3), (4). O método utilizado para o *crossover* foi o de um único ponto de corte [14] como mostrado na seção E.

#### F. Aplicar iterações

A princípio foram realizadas 50 iterações. Buscando diversificar o problema, o número de iterações é preenchido no momento da execução. O critério de parada foi mantido como o número de iterações, ou seja, o algoritmo terminará assim que o número de iterações seja alcançado.

### IV. RESULTADOS

Para execução dos algoritmos foi utilizada linguagem python em uma máquina com processador i7 com 8 gigabytes de memória. Visando diversificar os resultados do problema, foram criadas 3 matrizes constantes de acordo com a dificuldade do jogo, sendo elas em níveis: fácil, médio e difícil. O tempo de execução variou de acordo com o número de iterações, quantidade de indivíduos e o esquema utilizado.

O algoritmo foi executado com 20 indivíduos 10 vezes para cada

TABLE I  
TEMPO DE EXECUÇÃO DE ACORDO COM ITERAÇÕES E ESQUEMA UTILIZADO.

Esquema/Iterações	50	100
1	2.52s	4.75s
2	2.26s	4.11s
3	2.76	5.63s

nível, em 50 iterações e para cada esquema possível. Posteriormente foi executado novamente 10 vezes com 20 indivíduos agora em 100 iterações para cada nível e para cada esquema.

TABLE II  
MÉDIA DOS VALORES OBTIDOS DURANTE 10 EXECUÇÕES EM 50 ITERAÇÕES.

Esquema	Fácil	Médio	Difícil
1	73,3%	71.75%	73.3%
2	61.22%	61.9%	61.9%
3	63.57%	62.42%	62.42%

TABLE III  
MÉDIA DOS VALORES OBTIDOS DURANTE 10 EXECUÇÕES EM 100  
ITERAÇÕES.

Esquema	Fácil	Médio	Difícil
1	73.77%	73.83%	73.77%
2	61.87%	62%	62%
3	63.5%	64.42%	64.42%

Os graficos a seguir demonstram como a média dos indivíduos alteraram de acordo com o número de iterações aplicados no algoritmo e com o esquema utilizado.

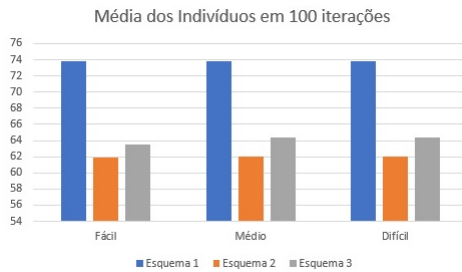
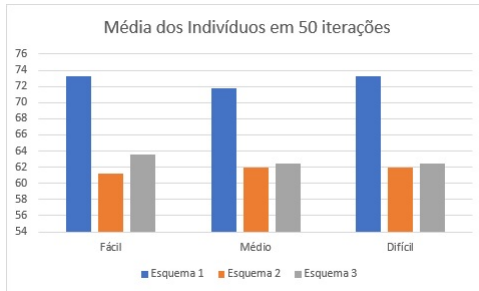


TABLE IV  
DESVIO PADRÃO DOS INDIVÍDUOS EM 50 ITERAÇÕES

Esquema	Fácil	Médio	Difícil
1	3.23%	3.15%	2.97%
2	1.30%	1.79%	1.21%
3	1.39%	1.61%	1.03%

TABLE V  
DESVIO PADRÃO DOS INDIVÍDUOS EM 100 ITERAÇÕES

Esquema	Fácil	Médio	Difícil
1	3.49%	3.06%	4.26%
2	1.55%	2.17%	2.16%
3	0.54%	1.39%	1.78%

## V. CONCLUSÃO

Os algoritmos genéticos conseguem chegar a resultados ótimos de acordo com os métodos de *cross over*, mutação entre outros. Tomando como comparatório os resultados de [19], onde foi utilizado apenas o esquema (2) e de acordo com o número de iterações respectivas utilizados no artigo base o algoritmo é eficiente para minimizar os valores repetidos dentro da matriz principal do Sudoku. Para o problema, o algoritmo desenvolvido chegou mais perto de resolver a matriz completa com o esquema 1 no nível fácil, médio e difícil, aplicando respectivamente o algoritmo em 50 e 100 iterações, onde o fitness do cromossomo em todos testes foi superior a 70% da matriz preenchida de maneira correta de acordo com suas restrições do jogo. O esquema 2, foi o que obteve os menores valores mediante as execuções e iterações. Pode-se perceber que o algoritmo, variável ao esquema utilizado e ao número de iterações, consegue convergir, porém não converge em uma solução ótima para o problema.

## REFERENCES

- [1] G. M. N. C. R. B. B. M. L. R. A. F. Alex F. V. Machado, Esteban W. G. Clua, "Heurísticas para a resolução de problemas np-completos no domínio de jogos," pp. 1 – 4.
- [2] F. R. C. Leoncio José de Almeida Reis, "Electronic Games and the Search for Excitement," pp. 1 – 10, 2002.
- [3] O. A. R. Junior, "Desenvolvimento de raciocínio lógico por meio do jogo sudoku," Master's thesis.
- [4] J. F. Crook, "A Pencil-and-Paper Algorithm for Solving Sudoku Puzzles," pp. 1–9, 2007.
- [5] I. H. I. Yuji Sato, Member, "Solving Sudoku with Genetic Operations that Preserve Building Blocks," pp. 1 – 10.
- [6] J. H. Holland, *Evolução*, vol. 3. ARTMED EDITORA SA, 2006.
- [7] R. B. d. B. J. R. B. d. A. F. R. F. P. C. T. Renata Iani Werneck, Renata Helena Monteiro Sindeaux Cleber Machado de Souza, *Introdução ao estudo da genética*, vol. 2.
- [8] D. dos Santos Gomes, "Inteligência artificial: Aplicações e conceitos," *Revista Olhar Científico*, vol. 1, no. 2, pp. 1 – 13, 2010.
- [9] C. H. A. Antônio Gaspar-Cunha, Ricardo Takahashi, *Manual de computação evolutiva e metaheurística*, vol. 1. Editora da Universidade Federal de Minas Gerais, 2012.
- [10] A. R. Christian Blum, *Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison*, vol. 3. ACM Computing Surveys, 2003.
- [11] D. A. C. B. Sidney Renato Silveira, "Jogos educativos computadorizados utilizando a abordagem de algoritmos genéticos," in *IV Congresso RIBIE*, (Brasília), pp. 1 – 13, 1998.
- [12] F. S. O. Milton Roberto Heinen, "Algoritmos genéticos aplicados ao problema de roteamento de veículos," Master's thesis, 2006.
- [13] E. N. C. J. A. d. V. Frederico Antônio Pinheiro, Alan Neiba Moreira, "Algoritmo genético aplicado ao controle de tensão em sistemas elétricos de potência," pp. 1 – 6.
- [14] C. C. Rückauer, "OPERADORES GENETICOS SOBRE PERMUTACIONES APLICADOS A LA RESOLUCIÓN DEL SUDOKU," Master's thesis.
- [15] A. C. B. D. Paulo Henrique Ribeiro Gabriel, "Fundamentos de algoritmos evolutivos," Master's thesis.
- [16] E. Ferneda, "Aplicando algoritmos genéticos na recuperação de informação," *Revista de Ciência da Informação*, vol. 10, no. 1, pp. 1 – 9, 2009.
- [17] R. T. Jakob Vesterstrøm, "A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems," pp. 1 – 8.

- [18] C. A. C. Efrén Mezura-Montes, Jesús Velazquez-Reyes, “*A Comparative Study of Differential Evolution Variants for Global Optimization*,” pp. 1 – 8.
- [19] P. J. Urszula Boryczka, “*Solving the Sudoku with the differential evolution*,” pp. 1 – 12.