

# Algoritmo genético para resolução de Sudoku

DANIEL TEODORO GONÇALVES MARIANO, KEIJI YAMANAKA (Ph.D)

Universidade Federal de Uberlândia  
Uberlândia-MG, Brasil  
dtgmariano@gmail.com, keiji@ufu.br

## Abstract

*Algoritmo genético é uma heurística de busca que simula o processo de evolução das espécies observado em sistemas biológicos. A aplicação vigente visa desenvolver um algoritmo genético para a resolução do puzzle Sudoku.*

## I. INTRODUÇÃO

### 1. Algoritmos Evolucionários

Algoritmos evolucionários utilizam modelos computacionais baseados em processos naturais de evolução como um instrumento para a resolução de problemas. Os modelos computacionais propostos se orientam pela simulação de conceitos de evolução das espécies: seleção, mutação e reprodução. Os algoritmos evolucionários operam proporcionando a manutenção de uma população de estruturas, chamadas de indivíduos ou cromossomos. Esses indivíduos se comportam de maneira similar à evolução das espécies, sendo submetidos à operadores genéticos (recombinação e mutação). Cada cromossomo recebe uma avaliação de acordo com o nível de aptidão que este possui dentro do contexto de um problema.

### 2. Algoritmos Genéticos

Algoritmos genéticos são um ramo dos algoritmos evolucionários, podendo ser definido como uma técnica de busca baseada numa metáfora do processo biológico de evolução natural. São técnicas heurísticas de otimização global. A otimização global é uma questão que opõe os GAs aos métodos como gradiente (hill climbing), que seguem a derivada de uma função de forma a encontrar o máximo de uma

função, ficando facilmente retidos em máximos locais. Nos GAs, populações de indivíduos são criadas e submetidas aos operadores genéticos: seleção, recombinação (crossover) e mutação. Tais operadores utilizam uma caracterização da qualidade de cada indivíduo como solução do problema em questão chamada de avaliação. Dessa forma geram um processo de evolução natural destes indivíduos, que eventualmente devera gerar um indivíduo que caracterizara uma boa solução para o nosso problema.

### 3. Etapas de um AG

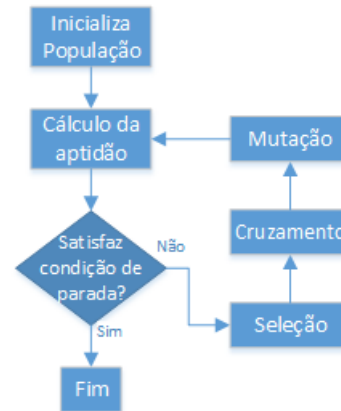
Um AG pode ser dividido nas seguintes etapas:

1. Inicialização: A população inicial de soluções candidatas são usualmente geradas randomicamente ao longo do espaço de busca. Conhecimentos ou informações específicas do domínio do problema podem ser incorporadas ao algoritmo.
2. Avaliação: Uma vez que a população é iniciada ou uma população descendente é criada, os valores de aptidão das soluções candidatas é avaliada.
3. Seleção: Aloca mais cópias dessas soluções com valores de aptidão superior e, assim, impõe o mecanismo de sobre-

vivência do mais apto nas soluções candidatas. A idéia principal da seleção é de preferir as melhores soluções em relação as piores.

4. **Recombinação:** Combina partes de dois ou mais cromossomos (soluções/indivíduos parentais) para gerar um novo cromossomo. Existem diferentes maneiras de se realizar a combinação entre indivíduos. A performance dessa etapa está relacionado com o quão apropriado é o mecanismo de recombinação projetado para aquele contexto. A prole sob recombinação não será idêntico a um pai em particular, pois irá recombinar traços dos pais de uma maneira nova
5. **Mutação:** Enquanto a recombinação ocorre envolvendo dois ou mais cromossomos a mutação ocorre individualmente. A mutação promove a alteração da solução de maneira randômica. Existem diversas variações de mutações, geralmente envolvendo uma ou mais alterações dos traços de um cromossomo.
6. **Reposição:** A população criada através da seleção, recombinação e mutação, substituirá os cromossomos pais.
7. **Verificação da condição parada:** Enquanto a condição de parada não foi encontrada, repete-se os procedimentos do passo 2 ao 6.

**Figure 1:** Diagrama de blocos de um AG básico.



#### 4. Sudoku

Sudoku é um puzzle de inserção de números que alcançou notável popularidade nos últimos anos. Em sua forma clássica, o objetivo é preencher uma grade de dimensões 9x9 com os dígitos de 1 a 9, observadas as seguintes restrições: cada linha, cada coluna e cada um dos nove subgrids de dimensões 3x3 devem conter uma permutação dos dígitos pertencente ao espaço amostral  $U = 1, 2, 3, 4, 5, 6, 7, 8, 9$ . O solucionador de Sudoku deve preencher os elementos da matriz para obter uma solução válida, que não viole as restrições do puzzle.

**Figure 2:** Exemplo de um sudoku.

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

**Figure 3:** Exemplo de um sudoku solucionado.

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

**Figure 4:** Representação por cada espaço do sudoku.

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

Cromossomo = { 8, 2, 1, ..., 3, 2, 9 }

A outra estratégia adotada foi criar uma lista composta por cada um dos 9 sub-grid (3x3) do sudoku. Cada sub-grid é expresso por uma outra lista composta por valores inteiros de tamanho 9, representando o conteúdo de cada espaço que compõe o sub-grid.

## II. MÉTODOS

O artigo vigente apresentará os princípios básicos de um algoritmo genético no desenvolvido de uma aplicação para encontrar soluções para um problema de Sudoku. O software da aplicação foi desenvolvido na plataforma .NET, em linguagem computacional CSharp (C#).

### 1. Representação cromossomial

Para o desenvolvimento do problema é necessário que a informação do problema seja traduzida em uma linguagem apta para o computador manipular. O sudoku trata-se de um puzzle de 81 espaços, em sua representação tradicional 9x9. Para representar o puzzle foram adotadas duas estratégias distintas. A primeira estratégia consiste em efetuar a representação de cada posição do cromossomo em uma lista de valores inteiros de tamanho 81. Cada item da lista representa um espaço do sudoku e o seu valor indica o número que o espaço possui.

**Figure 5:** Representação por cada subgrid do sudoku.

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

Cromossomo = { SG1, SG2, ..., SG8, SG9 }

SG1 = { 8, 2, 1, 5, 4, 9, 7, 6, 3 }

SG9 = { 1, 7, 5, 8, 4, 6, 3, 2, 9 }

A vantagem da última estratégia é que com a sua implementação permite resolver o problema da restrição onde os itens do sub-grid devem ser compostos pela permutação de  $U=\{1...9\}$ .

## 2. Cálculo de aptidão

O cálculo de aptidão é a etapa do algoritmo genético em que se determina a qualidade de um indivíduo como solução do problema em questão. Cada cromossomo da população é avaliado recendo um valor numérico para representar o seu grau de aptidão. Essa nota servirá de base para a etapa de seleção de pais. Dessa forma os indivíduos com melhor avaliação terão maior probabilidade de serem escolhidos como pais para a próxima geração de cromossomos. Para o problema tratado nesse artigo, o cálculo de aptidão levou em consideração o seguintes fatores:

1. Número de números faltando do espaço amostral  $U=\{1:9\}$  para cada coluna (i);
2. Número de números faltando do espaço amostral  $U=\{1:9\}$  para cada linha (j);

$$fit = \frac{100.0}{penaltyscore} \quad (1)$$

$$penaltyScore = \sum_{i=1}^n g_i + \sum_{j=1}^n g_j \quad (2)$$

$$g_i(x) = |A - x_i| \quad (3)$$

$$g_j(x) = |A - x_j| \quad (4)$$

$$A = 1, 2, 3, 4, 5, 6, 7, 8, 9 \quad (5)$$

Para tratamento de excessão, no caso de uma divisão por zero, caso o valor penalidade seja igual a 0 ( $penaltyScore = 0.0$ ) o valor de fitness é 100.0.

## 3. Elitismo

Trata-se de uma modificação no módulo de população com o intuito de garantir o crescimento positivo do desempenho do AG ao decorrer das gerações. No elitismo os  $n$  melhores indivíduos de cada geração são preservados para a próxima geração. Em um exemplo de aplicação de elitismo, uma população com 10 cromossomos (AG com elitismo de valor  $n$

igual a 2), a cada geração, dois dos melhores indivíduos da população são reservado. Em seguida serão selecionados 8 pais para gerarem 8 descendentes. Os indivíduos selecionados como elite são adicionados a população após o método de seleção completando os 10 indivíduos da população.

## 4. Seleção dos pais

A etapa de seleção dos pais é baseado no mecanismo de seleção natural que atua sobre as espécies biológicas. Indivíduos de uma população que possuem maior aptidão terão maior probabilidade de se tornarem pais e consequentemente gerar descendentes que carreguem parte de sua informação genética. Indivíduos menos aptos também podem se tornar pais, embora a probabilidade desse evento ocorrer seja menor. Caso a possibilidade de reprodução seja restrita apenas aos melhores indivíduos, a população tenderá a ser composta de indivíduos cada vez mais semelhantes e faltará diversidade a esta população para que a evolução possa prosseguir de forma satisfatória. A tal efeito é denominado como convergência genética, podendo ser minimizado ou evitado através da seleção equilibrada de indivíduos menos aptos da população. Existem diversos métodos de seleção propostos, tais como: seleção por roleta, seleção universal estocástica, seleção por ranking, seleção por torneio, entre outros. Para a aplicação foi utilizado o método de seleção por roleta.

### 4.1 Método da roleta viciada

Dada uma população de cromossomos, cada componente é avaliado de forma a determinar o seu nível de aptidão.

**Table 1:** *Dados de uma dada população*

Indivíduo	Avaliação
A	4
B	1
C	3
D	4
E	5
F	6

A probabilidade de seleção de cada indivíduo é determinada de acordo com o seu nível de aptidão em relação ao somatório dos níveis de aptidão de toda a população 6.

$$ps(i) = \frac{fit(i)}{\sum_{j=1}^n fit(j)} * 100 \quad (6)$$

Para estabelecer a probabilidade de seleção acumulada dos cromossomos da população é realizada a somatória das probabilidades de seleção do n-ésimo cromosso com a dos cromossomos anteriores conforme demonstrado na equação 7.

$$psa(i) = \sum_{j=1}^i fit(j) \quad (7)$$

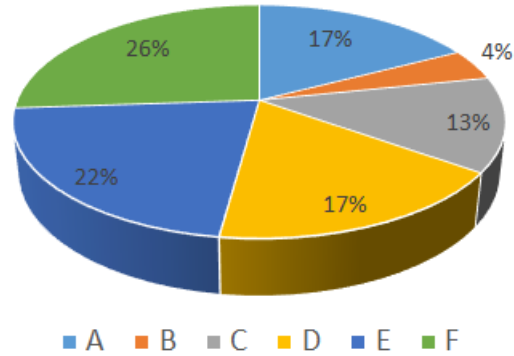
A tabela ?? mostra os indivíduos da população com seus valores de avaliação, probabilidade de seleção e probabilidade acumulada de seleção.

**Table 2:** *Distribuição das probabilidades (seleção e acumulada)*

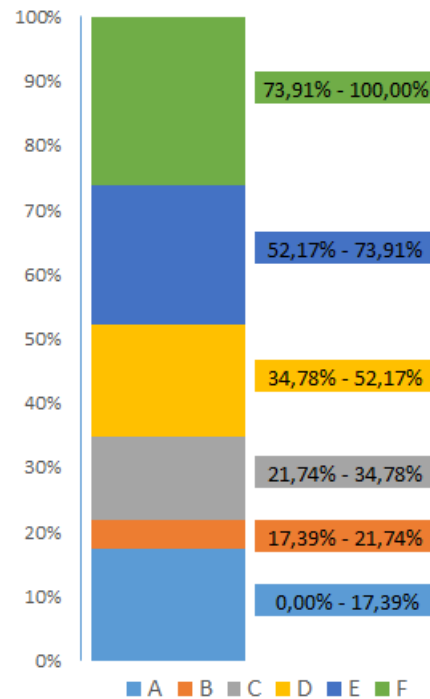
Indivíduo	Avaliação	Ps (%)	Psa (%)
A	4	17,39	17,39
B	1	4,35	21,74
C	3	13,04	34,78
D	4	17,39	52,17
E	5	21,74	73,91
F	6	26,09	100,00

A partir desses dados é possível visualizar a configuração da roleta conforme a figura ??.

Cada indivíduo é representado por uma fatia da roleta.

**Figure 6:** *Roleta.*

Observa-se que os indivíduos com maior probabilidade de seleção, ocupam uma área maior da roleta. A probabilidade de seleção de cada indivíduo da população é distribuída em intervalos conforme a figura ??

**Figure 7:** *Distribuição.*

A partir dessa distribuição, um número

randômico dentro do intervalo [0.00%; 100.00%] é gerado. Dentro dos intervalos de distribuição da roleta, escolhe-se o cromossomo cujo intervalo o número gerado pertencer. O procedimento é repetido  $n$  vezes onde  $n$  é o tamanho da população.

## 4.2 Ranking

A seleção por ranking é um método de seleção que evita a convergência prematura e a dominância de um superindivíduo. O seu princípio consiste em ordenar todos os elementos de acordo com a sua função de avaliação e usar este ranking como base da seleção, ao invés de usar diretamente o valor da avaliação. Uma vez definidos os novos valores de avaliação destes indivíduos, um método tradicional, tal como o da roleta vi-ciada, pode ser utilizado para a escolha dos pais que serão submetidos aos operadores genéticos.

## 4.3 Torneio

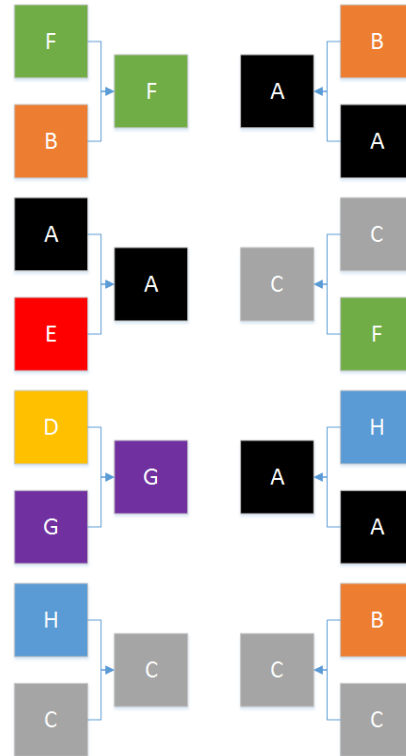
O método de seleção baseada em um torneio consiste em selecionar uma série de indivíduos da população para competições diretas. Os cromossomos competem diretamente, tendo como critério a função de avaliação de cada um. O cromossomo com melhor avaliação na competição é escolhido como pai. Um parâmetro para o torneio é definir o número de participantes por competição ( $k$ ). Em um exemplo de AG tem-se a seguinte população, composta por 8 indivíduos:

**Table 3:** Exemplo de população

Indivíduo	Avaliação
A	10
B	5
C	9
D	1
E	3
F	7
G	6
H	2

Para um critério  $k$  com valor igual a 2, serão selecionados aleatoriamente oito duplas. De cada dupla será reservado o indivíduo com função de avaliação superior.

**Figure 8:** Exemplo de torneio.



A partir dos indivíduos selecionados por torneio, é que se efetua o processo de cruzamento.

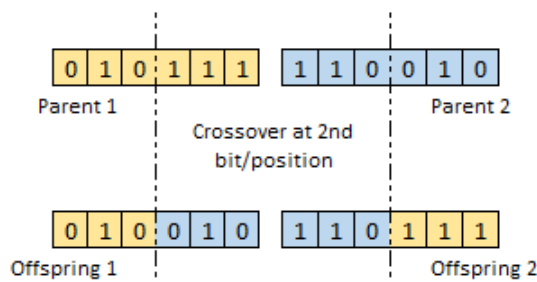
## 5. Cruzamento (Crossover)

Crossover é um operador genético usado para reprogramar a configuração de um cromossomo de uma geração para a próxima. Este operador é análogo à reprodução biológica de forma a promover a recombinação genética de duas ou mais soluções progenitoras e consequentemente gerando uma solução descendente. Existem diferentes técnicas de recombinação: com um ponto, com dois pontos, "corta e emenda", uniforme, metade uniforme, com três progenitores, cromossomos ordenados, tendenciosa, entre outras.

### 5.1 Crossover de um ponto

Neste cruzamento, um único ponto para recombinação é selecionado randomicamente para os cromossomos progenitores. Todos os genes a partir deste ponto são trocados entre os cromossomos. Os organismos resultantes são os filhos. A figura ?? ilustra um cruzamento de um ponto.

Figure 9: Cruzamento de um ponto.

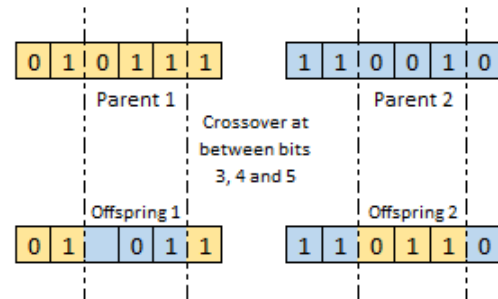


Entretanto existe a possibilidade de os pais não realizarem a recombinação. Isto ocorre porque existe uma probabilidade de cruzamento ( $P_c$ ) a ser considerada inicialmente. Um valor de  $P_c$  é definido nas configurações do AG antes do algoritmo ser iniciado. Para cada cromossomo parental é gerado um valor randômico (random) variando entre [0%; 100%]. Se a variável random possuir um valor menor ou igual ao valor de  $P_c$ , o cruzamento é efetuado. Caso contrário, os indivíduos não efetuam o cruzamento mantendo suas características para a próxima geração.

### 5.2 Crossover de dois pontos

Este cruzamento é semelhante ao cruzamento de um ponto, embora sejam selecionados dois pontos para recombinação. O conteúdo genético delimitado por estes pontos é trocado dos cromossomos progenitores para os cromossomos descendentes. Os demais alelos fora deste intervalo são preservados. A figura ?? ilustra o cruzamento de dois pontos.

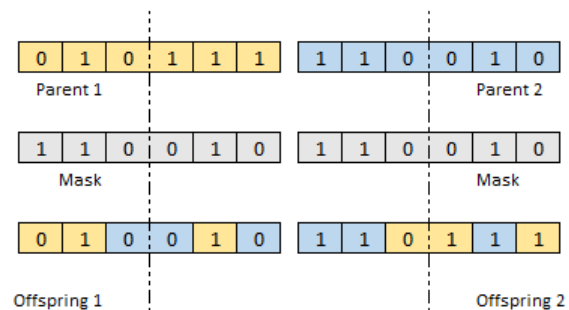
Figure 10: Cruzamento de dois pontos.



### 5.3 Crossover uniforme

Para cada gene é sorteado um número 0 ou 1. Se o valor sorteado for igual a 1, o primeiro filho recebe o gene da mesma posição do primeiro pai (de maneira idêntica o segundo filho recebe o gene da mesma posição do segundo pai). Se o valor sorteado for 0, as atribuições serão invertidas: o primeiro filho recebe o gene da posição corrente do segundo pai e o segundo filho recebe o gene corrente do primeiro pai.

Figure 11: Cruzamento uniforme.



O crossover uniforme é uma alternativa ao crossover de dois pontos. Enquanto este último possui uma maior probabilidade de separar partes do cromossomo de grande cromossomo durante o cruzamento, o cruzamento uniforme tende a conservar esquemas longos com a mesma probabilidade que preserva esquemas de menor comprimento, desde que ambos tenham a mesma ordem. Por outro lado, devido ao fato de fazer um sorteio para cada posição, este cruzamento tem uma grande pos-

sibilidade de estragar todo e qualquer esquema, mas em média o seu desempenho é superior ao cruzamento de um ponto e ao cruzamento de dois pontos.

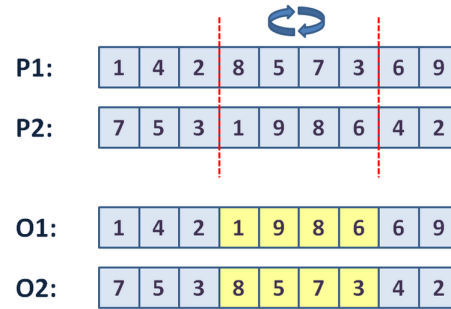
#### 5.4 Partially Matched Crossover (PMX)

Sugerido por Goldberg e Lingle, o PMX é um operador de cruzamento que age trocando blocos de genes entre os pais, gerando novos filhos.

1. Randomicamente seleciona uma porção dos alelos do pai 1 e copia essa mesma porção para o filho 1. Os alelos do filho são organizados de forma que, os alelos que estão fora zona de corte, recebem um valor '\*\*'
2. Procurando no mesmo segmento do Pai 2, seleciona cada valor que ainda não foi copiado para o filho 1.
3. Para cada um desses valores:
  - (a) Observa o index  $i$  desse valor no Pai 2. Localiza-se o valor  $v$  que encontra-se nesse no index  $i$  no Pai 1. Procura-se o valor  $v$  no Pai 2.
  - (b) Se o index desse valor  $v$  no Pai 2 fizer parte da porção original, retorne ao passo anterior, utilizando esse valor.
  - (c) Se o index desse valor  $v$  no Pai 2 não fizer parte da porção original, insira esse valor no cromossomo do filho 1 na mesma posição.
4. Os valores com '\*\*' no filho 1 recebem os valores de index correspondente do pai 2.

A figura ?? ilustra o operador PMX.

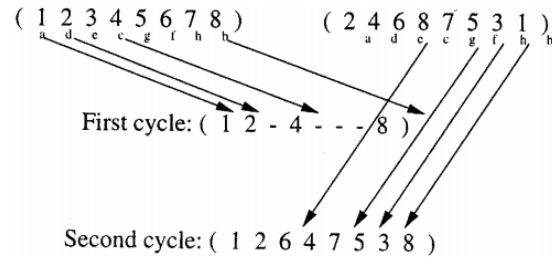
Figure 12: PMX



#### 5.5 Cycle Crossover (CX)

Proposto por Oliver et al. Este operador gera filhos que preservam a posição absoluta das cidades provenientes dos cromossomos pais. Ele trabalha sobre um subconjunto de cidades que ocupa um mesmo subconjunto de posições em ambos os pais. Estes pontos são copiados de um dos pais, nas mesmas posições, para um filho. As posições remanescentes são completadas com as cidades do outro pai. Assim, cada cidade e sua posição são herdadas de um dos pais. A figura ?? ilustra o operador CX.

Figure 13: CX



#### 5.6 Sudoku

Para a aplicação do Sudoku, optou-se por uma estratégia de crossover de maneira a travar os valores estipulados pelo usuário. Utilizou-se uma máscara, que é composta pelas entradas determinadas pelo usuário. Os espaços do sudoku onde o usuário não inseriu entradas, são preenchido por valores nulos (0). O crossover se dá entre os subgrids e não entre os conteúdos dos mesmos (Para a estratégia de representação cromossomial por subgrids), conforme



pode ser verificado na figura abaixo:

**Figure 14:** *Crossover para cada subgrid do sudoku.*

192365874	125346789	125678493	123456789	23456789	23456789	1742139685	418236579	173952648	916245738
194367825	835429716	267158493	519682743	368547291	742319685	458236971	173954682	926871534	
x9x36x8xx	xx5xxx7xx	xxxxxx4x3	xxxxxx7xx	xxxxxx91	742xx9685	4x8236xxx	17395x6xx	9x6xxxx30	
090360800	005000700	000000403	000000700	000000091	742009685	408236000	173950600	906000030	

## 6. Mutação

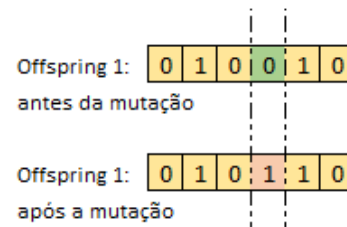
A mutação é um operador genético utilizado para manter a diversidade gênica de uma ger-

ação de cromossomos para a próxima. É análoga à mutação presente nos sistemas biológicos. Este operador pode alterar um ou mais genes de um cromossomo. De maneira semelhante à etapa de cruzamento, a mutação possui uma probabilidade associada ( $P_m$ ), que define a chance de tal processo ocorrer em um determinado momento. Um variável randômica, pertencente ao intervalo  $[0\%; 100\%]$ , é gerada. Se o valor dessa variável for inferior ou igual à  $P_m$ , a mutação ocorre. Se for superior, a operação não é realizada e o cromossomo mantém suas características sem alterações. Existem diferentes tipos de mutação: inversão de um único bit, inversão de todos os bits do cromossomo, por limites, não uniforme, uniforme, gaussiana.

### 6.1 Mutação de um único bit

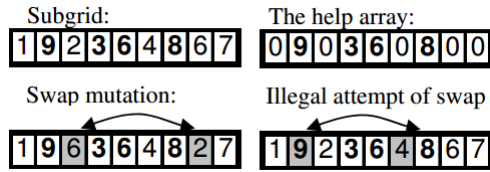
A mutação de um bit consiste em inverte um único bit para um outro valor. Para a aplicação do TSP, o uso dessa mutação não é adequada.

**Figure 15:** *Mutação: inversão de um único bit*



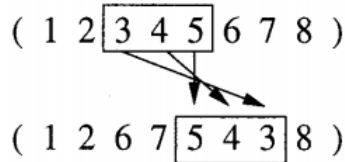
### 6.2 Mutação SWAP

A mutação swap, implementada neste trabalho, consiste basicamente na permutação de dois genes de um mesmo cromossomo. Este operador de reordenamento pode ser utilizado em problemas de ordenação, para prevenir a duplicação de genes, que poderia ocorrer com a mutação cartesiana.

Figure 16: *Mutação: Swap*

### 6.3 Mutação inversão

A inversão consiste em um outro operador de reordenamento. Neste operador, uma parte aleatória do cromossomo (situada entre dois pontos de mutação) é invertida. Esta operação é apresentada na figura ??

Figure 17: *Mutação inversão*

## III. RESULTADOS

Foi desenvolvido a seguinte interface para a aplicação: uma janela composta por diversas caixas de texto, correspondentes a cada espaço de um puzzle de sudoku. Assim que o algoritmo genético é iniciado, os valores do sudoku e o número de penalidades do cromossomo da melhor solução da geração atual, além do número da geração naquele instante, são atualizadas na interface.

Foram construídos sudokus 3x3, 6x6 e 9x9.

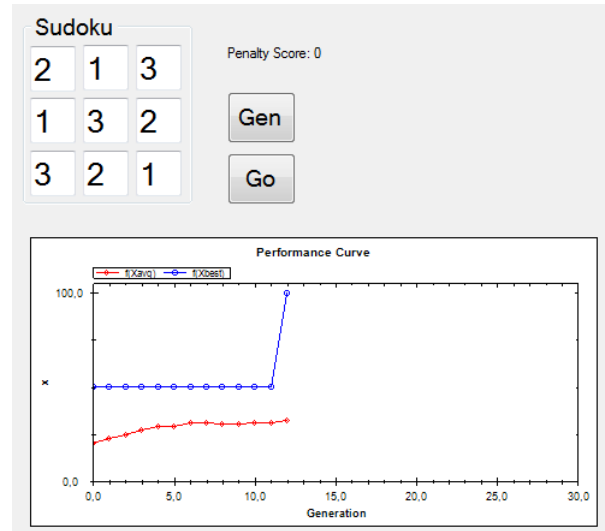
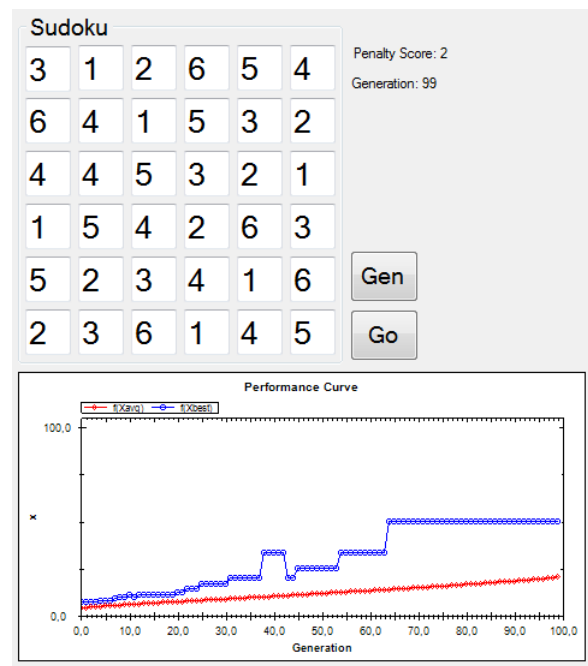
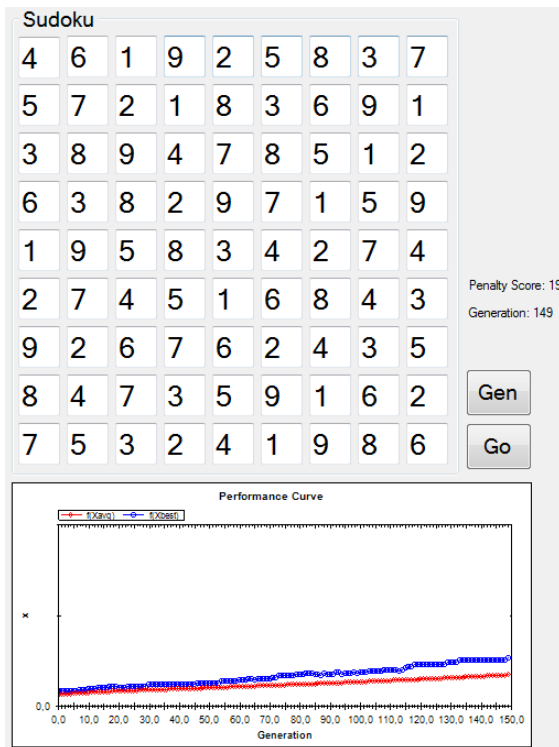
Figure 18: *Solucionador de Sudoku 3x3*Figure 19: *Solucionador de Sudoku 6x6*

Figure 20: Solucionador de Sudoku 9x9



#### IV. DISCUSSÃO

Foi possível constatar que a medida em que a ordem do problema foi incrementando, o AG não conseguiu fornecer soluções para a resolução do problema. Embora as mesmas estruturas de código (Classe GA, Indivíduo) e operadores de cruzamento e mutação tenham sido empregadas para a análise dos três tipos de Sudokus, o aumento da dimensão do problema mostrou-se uma variável extremamente impactante na performance da aplicação. Para o Sudoku 9x9, um problema maior detectado, foi a dificuldade de controlar a restrição dos sub-grids 3x3. É possível que a outra proposta de realizar a implementação pela composição de sub-grids 3x3 tenha melhor aproveitamento, uma vez que a restrição de permutar os valores dentro dos sub-grids seja eliminado. Alguns dos operadores de cruzamento como o de 1 e 2 pontos mostraram-se ineficientes para a resolução do problema, uma vez que o operador atua de forma a desorganizar outras

partes do sudoku. Pode-se observar que o operador de mutação, para uma taxa de probabilidade de mutação elevada, contribuiu positivamente para o incremento do fitness da população. Uma outra estratégia a ser adotada para tentar solucionar o problema seria o emprego de outros tipos de operadores de cruzamento como o Product Geometric Crossover, que foi construído propriamente para a resolução de problemas como o do Sudoku.

#### REFERENCES

- [1] Kedar Nath Das, Sumit Bhatia, Shubhin Puri, Kusum Deep, A Retrievable GA for Solving Sudoku Puzzles
- [2] Dr. John M. Weiss, Genetic Algorithms and Sudoku, Department of Mathematics and Computer Science, South Dakota School of Mines and Technology, MICS 2009
- [3] Alberto Moraglio, Julian Togelius, Simon Lucas, Product Geometric Crossover for the Sudoku Puzzle
- [4] Timo Mantere, Janne Koljonen; Solving, Rating and Generating Sudoku Puzzles with GA
- [5] Linden, R. (2006). Algoritmos Genéticos - Uma importante ferramenta da Inteligência Computacional.
- [6] Goldberg and R. Lingle. Alleles, loci, and the traveling salesman problem. In Proceedings of the First International Conference on Genetic Algorithms and Their Applications: July 24-26, 1985 at the Carnegie-Mellon University, Pittsburg, PA, volume 98423, page 154. Lawrence Erlbaum, 1988.
- [7] Larranaga, C. Kuijpers, R. Murga, I. Inza, and S. Dizdarevic. Genetic algorithms for the travelling salesman problem: A review of representations and operators. Artificial Intelligence Review, 13(2):129-170, 1999.

- [8] Oliver, D. Smith, and J. Holland. A study of permutation crossover operators on the travelling salesman problem. In Genetic Algorithms and Their Applications: Proceedings of the 2nd International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, Hilladale, NJ, 1987.