

Sudoku Geneticsharp

Paola PERETTI, Samy BENRHHALATE, Sébastien ROUVIÈRE, Valentin PETOIN

-FIREFIGHTER ESCAPE ROOM-



- Inspiré de la théorie de l'évolution de Darwin
- Développé aux US dans les années 1970.
- **Domaine d'application:** Optimisation discrète
- **Caractéristiques:** Pas très rapide, bonne heuristique pour les problèmes combinatoires
- **Caractéristiques spéciales:** Combinaison d'informations provenant de bons parents (crossover). De nombreuses variantes, par exemple des modèles de reproduction, des opérateurs.



ABOUT GS

L'heuristique ou euristique est « l'art d'inventer, de faire des découvertes » en résolvant des problèmes à partir de connaissances incomplètes. Ce type d'analyse permet d'aboutir en un temps limité à des conclusions vraisemblables. Celles-ci peuvent s'écarter de la solution optimale.

GENETIC ALGORITHM FLOW CHART

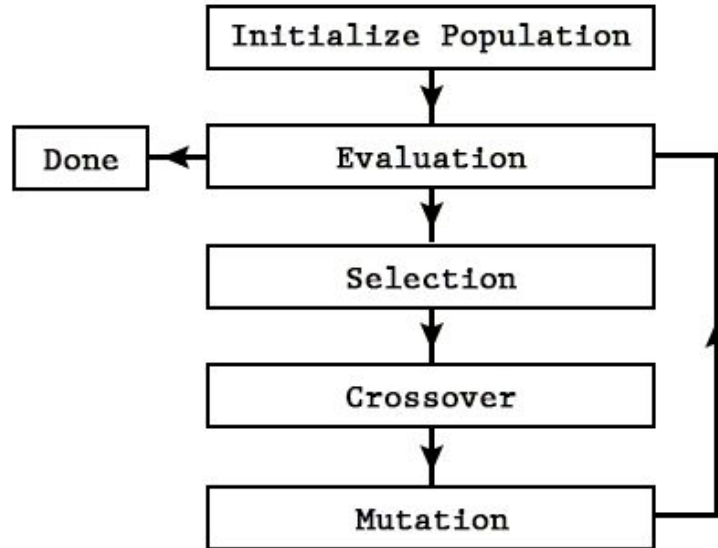


FIGURE 2

SUDOKU

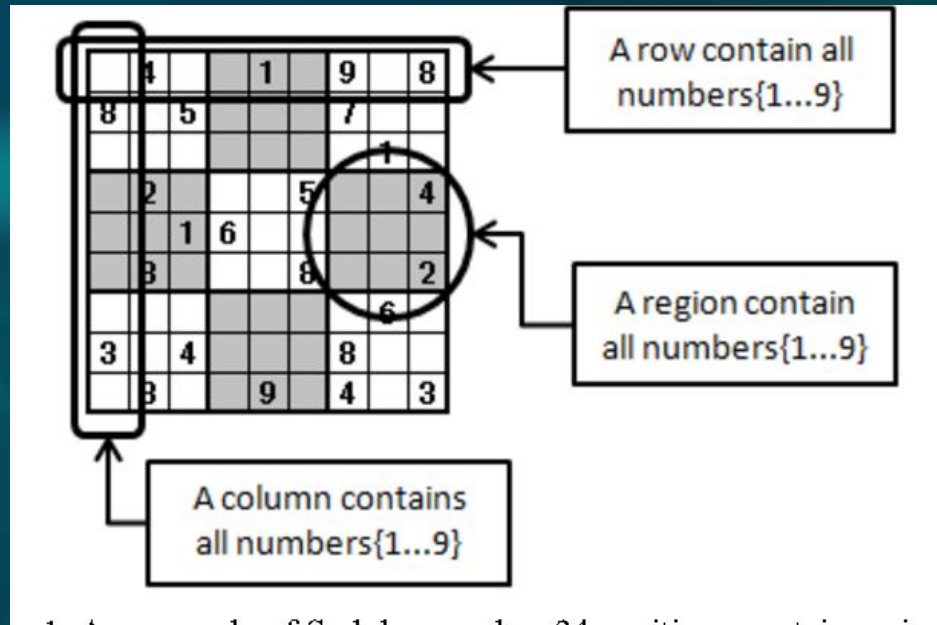


Fig. 1. An example of Sudoku puzzles, 24 positions contain a given number, the other position should be solved.

INITIALISATION

```
/// <summary>
/// Generates genes with digits for each index within the 81 Sudoku cells
/// </summary>
/// <param name="geneIndex"></param>
/// <returns>a gene with a digit for the corresponding cell index</returns>
public override Gene GenerateGene(int geneIndex)
{
    //If a target mask exist and has a digit for the cell, we use it.
    if (_targetSudoku != null && _targetSudoku.Cells[geneIndex] != 0)
    {
        return new Gene(_targetSudoku.Cells[geneIndex]);
    }
    var rnd = RandomizationProvider.Current;
    // otherwise we use a random digit.
    return new Gene(rnd.GetInt(1, 10));
}
```

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

il nous faut une population P qui correspond à l'ensemble des grilles possibles(sans respecter les règles)

EVALUATION

On utilise la fonction fitness via l'interface IFITNESS de Genetic Sharp:

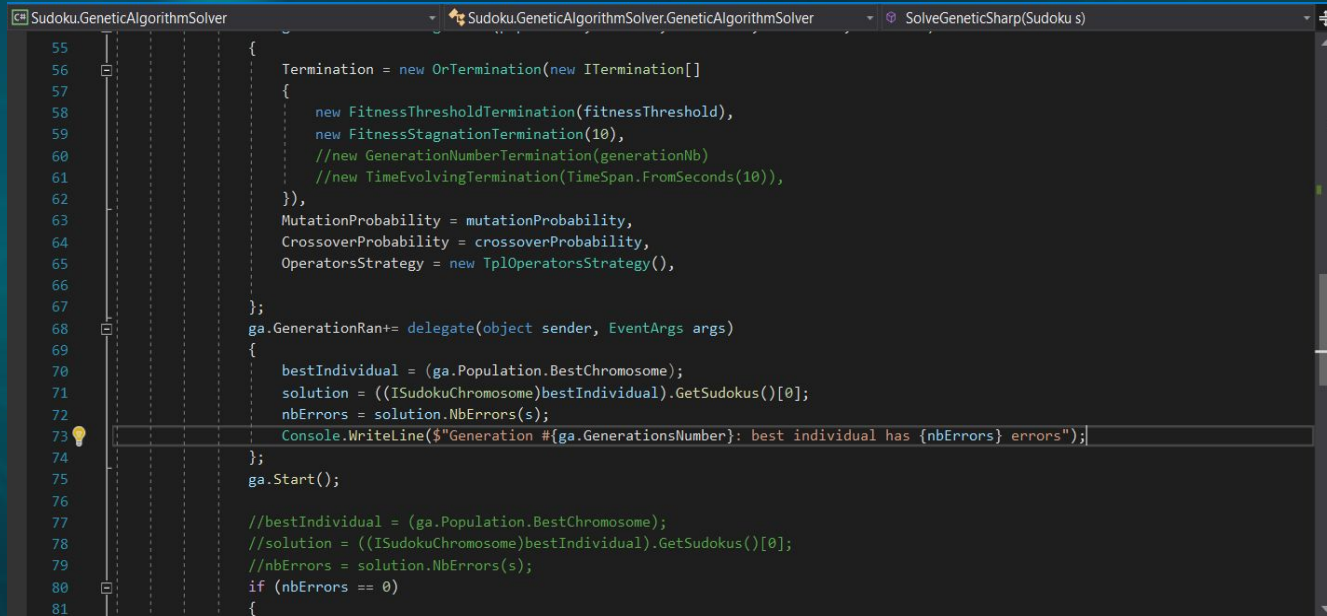
La fitness mesure, la qualité de l'individu exprimée sous forme d'un nombre ou d'un vecteur.

On dit qu'un individu i est meilleur que l'individu j quand i est plus proche de la solution que j .

```
/// <summary>
/// Evaluates a single Sudoku board by counting the duplicates in rows, boxes
/// and the digits differing from the target mask.
/// </summary>
/// <param name="testSudoku">the board to evaluate</param>
/// <returns>the number of mistakes the Sudoku contains.</returns>
public double Evaluate(Core.Sudoku testSudoku)
{
    // We use a large lambda expression to count duplicates in rows, columns and boxes
    var cells = testSudoku.Cells.Select((c, i) => new { index = i, cell = c }).ToList();
    var toTest = cells.GroupBy(x => x.index / 9).Select(g => g.Select(c => c.cell)) // rows
        .Concat(cells.GroupBy(x => x.index % 9).Select(g => g.Select(c => c.cell)) //columns
        .Concat(cells.GroupBy(x => x.index / 27 * 27 + x.index % 9 / 3 * 3).Select(g => g.Select(c => c.cell)))) //boxes
    var toReturn = -toTest.Sum(test => test.GroupBy(x => x).Select(g => g.Count() - 1).Sum()); // Summing over duplicates
    toReturn -= cells.Count(x => _targetSudoku.Cells[x.index] > 0 && _targetSudoku.Cells[x.index] != x.cell); // Mask
    return toReturn;
}
```

Selection

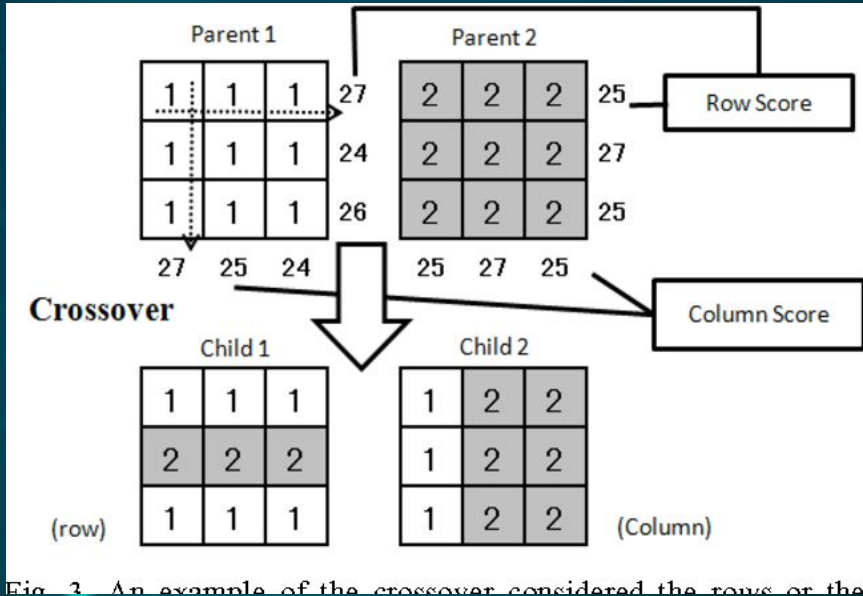
Pour déterminer quels individus sont plus enclins à obtenir les meilleurs résultats, une sélection est opérée. Nous gardons les sudokus avec le moins d'erreurs possibles qui représentent la génération "suivante"



```
Sudoku.GeneticAlgorithmSolver
Sudoku.GeneticAlgorithmSolver.GeneticAlgorithmSolver
SolveGeneticSharp(Sudoku s)

55 {
56     Termination = new OrTermination(new ITermination[]
57     {
58         new FitnessThresholdTermination(fitnessThreshold),
59         new FitnessStagnationTermination(10),
60         //new GenerationNumberTermination(generationNb)
61         //new TimeEvolvingTermination(TimeSpan.FromSeconds(10)),
62     }),
63     MutationProbability = mutationProbability,
64     CrossoverProbability = crossoverProbability,
65     OperatorsStrategy = new TplOperatorsStrategy(),
66 };
67
68 ga.GenerationRan += delegate(object sender, EventArgs args)
69 {
70     bestIndividual = (ga.Population.BestChromosome);
71     solution = ((ISudokuChromosome)bestIndividual).GetSudokus()[0];
72     nbErrors = solution.NbErrors(s);
73     Console.WriteLine($"Generation #{ga.GenerationsNumber}: best individual has {nbErrors} errors");
74 };
75 ga.Start();
76
77 //bestIndividual = (ga.Population.BestChromosome);
78 //solution = ((ISudokuChromosome)bestIndividual).GetSudokus()[0];
79 //nbErrors = solution.NbErrors(s);
80 if (nbErrors == 0)
81 {
```

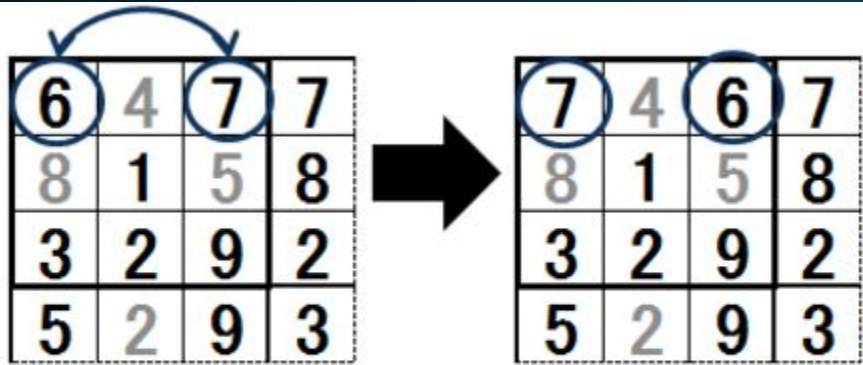
Crossover



Crossover: Lorsque deux enfants sont générés à partir de deux parents, des scores sont obtenus pour chacune des trois rangées qui constituent les sous-blocs des parents, et un enfant hérite de ceux qui ont les scores les plus élevés. Ensuite, les colonnes sont comparées de la même manière et l'autre enfant hérite de celles qui ont les scores les plus élevés.

Fig. 2. An example of the crossover considered the rows or the columns that constitute the sub-blocks.

Mutation



GY:initial placement

Fig. 3. An example of the swap mutation. Two numbers inside the sub block are selected randomly if the numbers are free to change.

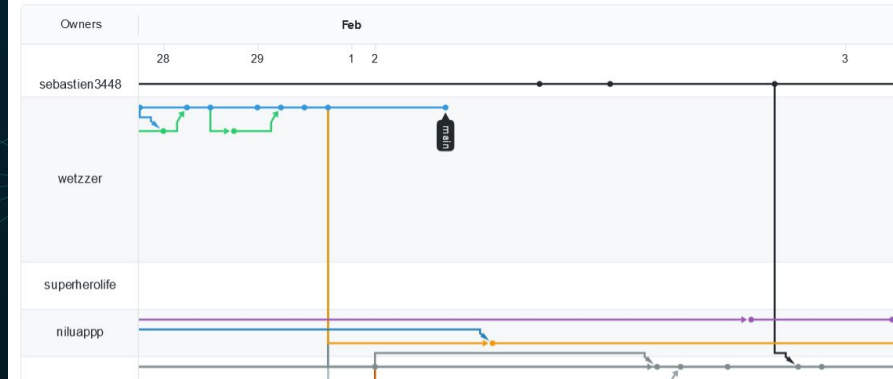
Mutation : Des mutations sont effectuées pour chaque sous-bloc en fonction du taux de mutation. Deux chiffres à l'intérieur d'un sous-bloc qui ne sont pas donnés dans le point de départ sont choisis au hasard et leurs positions sont échangées. Cette opération évite la duplication des chiffres à l'intérieur d'un sous-bloc et l'inclusion des chiffres initialement donnés dans les chiffres choisis au hasard.

Aspect collaboratif

- Des problèmes avec Github
- Les ordinateurs ont mal supporté VS code

Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.





**RESOLVONS LE
SUDOKU**

RESOURCES

- **GITHUB GENETIC SHARP:**

<https://github.com/giacomelli/GeneticSharp>

- **GENETIC SHARP EXTENSION: SUDOKU @JSBOIGE**

<https://github.com/giacomelli/GeneticSharp/tree/master/src/GeneticSharp.Extension>

- **Solving Sudoku with Genetic Operations that Preserve Building Blocks**

<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.173.6500&rep=rep1&type=pdf>

- **STOCHASTIC OPTIMIZATION APPROACHES FOR SOLVING SUDOKU**

<https://arxiv.org/ftp/arxiv/papers/0805/0805.0697.pdf>

- **DIFFERENT STEPS OF GENETIC ALGORITHMS**

https://www.tutorialspoint.com/genetic_algorithms/index.htm