**Université Ibn-Tofail Faculté des Sciences**

**Dép****artement d’Informatique**

Master spécialisé : BIG DATA & CLOUD COMPUTING

**Mini projet**

Présenté par

-Ahmed ADNANE

-Abdelkrim ACHAWAL

-Ilyas ZIAT

Méta-heuristiques : Les jeux du sudoku résolus par Recherche Harmonie

**Année Universitaire : 2022/2023**

**Université Ibn-Tofail | Faculté des Sciences | Département d’Informatique, B.P. : 133, Kénitra, Maroc**

Introduction sur le jeu :

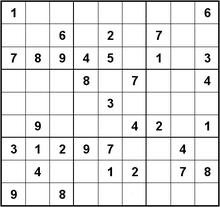
**Sudoku** est un jeu où l'on doit compléter des cases à l'aide de chiffres allant de 0 à 9, sur une grille de 9 cases sur 9. La règle est qu'aucun chiffre ne doit se retrouver deux fois dans la même ligne, la même colonne ou le même carré.

Travail a faire :

# Définitions :

## Représentation de la solution du Sudoku:

Dans le cadre de la méthode de recherche Harmony Search, les solutions du Sudoku sont représentées sous forme de texte. Chaque grille de Sudoku est décrite comme une séquence de 81 chiffres (ou caractères) représentant les valeurs des cellules dans une grille de 9x9. Les chiffres de 1 à 9 représentent les valeurs valides dans une grille de Sudoku, tandis qu'un espace ou un caractère spécial est utilisé pour représenter les cellules vides.



## Fonction objective :

## Représentation des solutions

**Variables:**

* x\_ij : Variable binaire indiquant si la valeur i est placée dans la case (j, k) (1) ou non (0).
* i : Valeur possible pour une case (1 à 9).
* j : Numéro de ligne (1 à 9).
* k : Numéro de colonne (1 à 9).

**Domaine de définition:**

* x\_ij ∈ {0, 1}

**Contraintes:**

* **Chaque case doit contenir une seule valeur:**

Σ\_i x\_ij = 1 ∀j, k

* **Chaque valeur ne peut apparaitre qu'une seule fois par ligne:**

Σ\_j x\_ij = 1 ∀i

* **Chaque valeur ne peut apparaitre qu'une seule fois par colonne:**

Σ\_k x\_ij = 1 ∀i

* **Chaque valeur ne peut apparaitre qu'une seule fois dans chaque bloc 3x3:**

Σ\_(j,k) in B\_l x\_ij = 1 ∀i, l

où B\_l est l'ensemble des cases du bloc l.

## Fonction objectif

**Minimiser le nombre de cases non assignées:**

f(x) = Σ\_(i,j,k) (1 - x\_ij)

**Alternativement, maximiser le nombre de cases assignées:**

f(x) = Σ\_(i,j,k) x\_ij

**Remarques:**

* D'autres formulations de la fonction objectif sont possibles, en fonction des besoins spécifiques.
* La fonction objectif peut être pondérée pour tenir compte de l'importance relative des différentes contraintes.

**Exemple:**

Supposons que nous avons un Sudoku partiellement rempli:

. 2 . | . . .

. . . | 6 . .

. . . | . . .

-------+-------+-------

. . . | . . .

. 8 . | . . .

. . . | . . .

-------+-------+-------

. . . | . . .

. . . | . . 4

Les variables x\_ij pour la case en haut à gauche (ligne 1, colonne 1) seraient:

* x\_11 = 0 (car la case n'est pas encore assignée)
* x\_21 = 1 (car la case est assignée à la valeur 2)
* x\_31 = 0 (car la case n'est pas encore assignée)
* ...

En utilisant la première formulation de la fonction objectif, on obtient:

f(x) = 16 - (2 + 1) = 13

Le but est de minimiser cette valeur en assignant des valeurs aux cases vides de sorte que toutes les contraintes soient satisfaites.



## EXPLIAUER la représentation des solutions et la fonction objectif

Show drafts

## Représentation des solutions

**Variables:**

* x\_ij : Variable binaire indiquant si la valeur i est placée dans la case (j, k) (1) ou non (0).
* i : Valeur possible pour une case (1 à 9).
* j : Numéro de ligne (1 à 9).
* k : Numéro de colonne (1 à 9).

**Domaine de définition:**

* x\_ij ∈ {0, 1}

**Explication:**

* Chaque case du Sudoku est représentée par une variable x\_ij.
* La valeur de x\_ij est 1 si la valeur i est assignée à la case (j, k), et 0 sinon.
* Le domaine de définition de x\_ij est l'ensemble {0, 1} car il ne peut y avoir que deux états possibles pour une case : vide ou remplie.

**Contraintes:**

* **Chaque case doit contenir une seule valeur:**

Σ\_i x\_ij = 1 ∀j, k

**Explication:**

* Cette contrainte garantit que chaque case du Sudoku est remplie par exactement une valeur.
* La somme des variables x\_ij pour une ligne ou une colonne donnée doit être égale à 1.
* **Chaque valeur ne peut apparaitre qu'une seule fois par ligne:**

Σ\_j x\_ij = 1 ∀i

**Explication:**

* Cette contrainte garantit que chaque valeur ne peut apparaitre qu'une seule fois par ligne.
* La somme des variables x\_ij pour une valeur i donnée doit être égale à 1.
* **Chaque valeur ne peut apparaitre qu'une seule fois par colonne:**

Σ\_k x\_ij = 1 ∀i

**Explication:**

* Cette contrainte garantit que chaque valeur ne peut apparaitre qu'une seule fois par colonne.
* La somme des variables x\_ij pour une valeur i donnée et une colonne k donnée doit être égale à 1.
* **Chaque valeur ne peut apparaitre qu'une seule fois dans chaque bloc 3x3:**

Σ\_(j,k) in B\_l x\_ij = 1 ∀i, l

**Explication:**

* Cette contrainte garantit que chaque valeur ne peut apparaitre qu'une seule fois dans chaque bloc 3x3.
* La somme des variables x\_ij pour une valeur i donnée et un bloc l donné doit être égale à 1.

**Exemple:**

Supposons que nous avons un Sudoku partiellement rempli:

. 2 . | . . .

. . . | 6 . .

. . . | . . .

-------+-------+-------

. . . | . . .

. 8 . | . . .

. . . | . . .

-------+-------+-------

. . . | . . .

. . . | . . 4

Les variables x\_ij pour la case en haut à gauche (ligne 1, colonne 1) seraient:

* x\_11 = 0 (car la case n'est pas encore assignée)
* x\_21 = 1 (car la case est assignée à la valeur 2)
* x\_31 = 0 (car la case n'est pas encore assignée)
* ...

## Fonction objectif

**Minimiser le nombre de cases non assignées:**

f(x) = Σ\_(i,j,k) (1 - x\_ij)

**Explication:**

* Cette fonction objectif vise à minimiser le nombre de cases vides dans le Sudoku.
* Pour chaque case vide, la valeur 1 - x\_ij est ajoutée à la fonction objectif.
* En minimisant cette fonction objectif, on obtient le nombre minimal de cases vides, ce qui correspond à une solution complète du Sudoku.

**Alternativement, maximiser le nombre de cases assignées:**

f(x) = Σ\_(i,j,k) x\_ij

**Explication:**

* Cette fonction objectif est équivalente à la première, mais elle vise à maximiser le nombre de cases remplies.
* En maximisant cette fonction objectif, on obtient le nombre maximal de cases remplies, ce qui correspond également à une solution complète du Sudoku.

**Remarques:**

* D'autres formulations de la fonction objectif sont possibles, en fonction des besoins spécifiques.