

## Master Mathématique et Application au Calcul Scientifique (MACS)

### MEMOIRE DE FIN D'ETUDES

### Pour l'obtention du Diplôme de Master Sciences et Techniques (MST)

## La confection des tableaux de service des enseignants dans un établissement d'enseignement secondaire

**Réalisé par :** MOUATASSIM Khadija

**Encadrant :** Pr. EL HILALI ALAOUI Ahmed

**Co-encadrant:** Pr. EL KHOMSSI Mohammed

**Soutenu le 18/06/2018**

**Devant le jury composé de:**

Pr. BENCHEIKH Ghizlane

Faculté des Sciences Juridiques Economiques  
et Sociales de Meknès

Pr. EL AYADI Rachid

Faculté des Sciences et Techniques de Fès

Pr. EL HILALI ALAOUI Ahmed

Faculté des Sciences et Techniques de Fès

Pr. EL KHOMSSI Mohammed

Faculté des Sciences et Techniques de Fès

Pr. HILALI Abdelmajid

Faculté des Sciences et Techniques de Fès

**Année Universitaire 2017 / 2018**

# Remerciements

Après avoir rendu grâce à Dieu le tout Puissant et le Miséricordieux, je tiens à exprimer ma profonde gratitude à notre Professeur et Encadrant Monsieur EL HILALI ALAOUI Ahmed, également je tiens à remercier mon Professeur et Co-encadrant Monsieur EL KHOMSSI Mohammed pour leur suivi et pour leur énorme soutien, qu'ils n'ont cessé de me prodiguer le long de la période du projet.

J'adresse mes vifs remerciements aux :

- Pr. BENCHEIKH Ghizlane
- Pr. EL AYADI Rachid
- Pr. EL HILALI ALAOUI Ahmed
- Pr. EL KHOMSSI Mohamed
- Pr. HILALI Abdelmajid

Pour avoir bien voulu examiner et juger ce travail.

Mes remerciements vont également à tout le personnel que j'ai contacté durant mon travail de recherche

Finalement, je remercie chaleureusement ma famille pour la confiance, l'amour et les encouragements.

# Dédicaces

Je dédie ce projet de fin d'études...

**A mes très chers parents**

Tous les mots du monde ne sauraient exprimer l'immense amour que je vous porte, ni la profonde gratitude que je vous témoigne, pour tous les efforts et les sacrifices, que vous n'avez jamais cessé de consentir, pour mon instruction et mon bien-être.

J'espère que vous trouverez dans ce modeste travail un témoignage de ma gratitude, ma profonde affection et mon profond respect.

Que DIEU tout puissant, vous garde et vous procure santé, bonheur et longue vie pour que vous demeuriez toujours une étoile dans mon ciel.

**A nos chers professeurs**

*Pour leur aides et soutien pendant toute notre période d'étude.*

**A tous ceux qui ont participé de près ou de loin à la réalisation de ce travail.**

# Sommaire

<i>Introduction</i> .....	7
<i>Chapitre 1 : Généralités sur la confection des tableaux de service des enseignants dans un établissement d'enseignement secondaire</i> .....	8
<i>Introduction</i> .....	8
<b>1. définition</b> .....	9
<b>2. Historique</b> .....	10
<b>3. Notions de base</b> .....	12
<b>4. Les contraintes</b> .....	13
<i>Conclusion</i> .....	15
<i>Chapitre 2 : Les Algorithmes Génétiques</i> .....	16
<i>Introduction</i> .....	16
<b>1. Algorithmes génétiques</b> .....	17
<i>Principe de base d'un AG standard</i> .....	17
1.1. Codage .....	19
1.2. Sélection .....	20
1.3. Croisement .....	22
1.4. Mutation .....	23
<b>2. Paramètres d'un AG</b> .....	24
<i>Conclusion</i> .....	24
<i>Chapitre 3 : formulation du problème de confection des tableaux de service des enseignants dans un lycée privé ...</i>	25
<i>Introduction</i> .....	25
<b>1. Position du problème</b> .....	26
1.1. Définition .....	26
1.2. La vérification de la suffisance d'enseignants et des salles .....	26
<b>2. Modélisation</b> .....	27
2.1. Les données .....	27
2.2. Variables .....	28
2.3. Les contraintes .....	29
2.4. La fonction objective .....	35
<b>3. Le Modèle Mathématique</b> .....	37
<i>Conclusion</i> .....	37
<i>Chapitre 4 : Application au problème d'un lycée privé</i> .....	38
<i>Introduction</i> .....	38

<b>1. Les données du lycée .....</b>	<b>40</b>
<b>2. Vérification de faisabilité du problème .....</b>	<b>42</b>
<b>3. L'adaptation des algorithmes génétiques .....</b>	<b>45</b>
<b>4. Programmation en langage C .....</b>	<b>47</b>
4.1. Génération d'un individu .....	47
4.2. fonction d'adaptation .....	54
<i>Conclusion.....</i>	<i>54</i>
<i>Conclusion générale et Perspectives.....</i>	<i>55</i>
<i>Bibliographie et web-graphie.....</i>	<i>56</i>

## Liste des figures

Figure : Organigramme d'un Algorithme G génétique standard .....	18
Figure : les cinq niveaux d'organisation d'un algorithme génétique .....	19
Figure : exemple d'un chromosome.....	20
Figure : Représentation d'une sélection par tournoi d'individus pour un critère de maximisation Chaque individu représente une solution possible .....	21
Figure : Représentation d'un croisement en un point de deux chaînes .....	22
Figure : Représentation d'un croisement en deux points .....	22
Figure : Représentation d'un croisement uniforme.....	23
Figure : Représentation d'une mutation de bits dans une chaîne .....	23
Figure : initialisation de la matrice (indv) .....	47
Figure : initialisation de la matrice P_C .....	47
Figure : initialisation de la matrice M_P.....	48
Figure : initialisation de la matrice M_C .....	48
Figure : initialisation du tableau des groupes .....	49
Figure : initialisation du tableau de nombre d'heures exigés.....	49
Figure : fonction qui retourne l'indice de la colonne qui représente le nombre maximal des enseignants disponibles .....	50
Figure : première étape pour générer un individu .....	51
Figure : deuxième étape pour générer un individu.....	51
Figure : un extrait d'individu obtenu .....	52
Figure : programme de la fonction d'adaptation ( <i>fitness</i> ).....	54

## Liste des tableaux

Tableau : les horaires du lundi d'enseignant $i$ avec la classe $j$ .....	33
Tableau : les horaires d'une semaine .....	34
Tableau : exemple d'un décalage d'une heure .....	35
Tableau : exemple d'un décalage de deux heures.....	35
Tableau : Un décalage de plus de trois heures .....	35
Tableau : Tableaux des abréviations.....	39
Tableau : Nombre de classes pour chaque branche.....	40
Tableau : le nombre d'enseignants disponibles .....	40
Tableau : les groupes des enseignants .....	41
Tableau : nombres d'heures du travail fixés par chaque enseignant .....	41
Tableau : la distribution des matières et des séances pour chaque niveau .....	42
Tableau ; vérification de suffisance des enseignants.....	43
Tableau : un exemple de codage d'un individu.....	45
Tableau : exemple de mutation dans un individu .....	46

# *Introduction*

Au cours de notre vie professionnelle, on se trouve confronté au problème de la planification d'horaire de travail soit dans les usines, les hôpitaux, les lycées ou les universités. Et cela nécessite une élaboration périodique et stratégique de plannings de travail du personnel.

Parmi la vaste famille des problèmes de planification d'horaire, on trouve celui de la confection d'emploi du temps dans les établissements éducatifs, notamment la confection des tableaux de service. Ce problème est très important et ardu, dont la réalisation à la main est une tâche draconienne qui peut mobiliser plusieurs personnes à plusieurs jours de travail, sans oublier, que toute modification des données du problème peut complètement remettre en cause la solution trouvée.

Ces difficultés ont conduit à l'idée d'assister par ordinateur l'élaboration des tableaux de service, en adoptant des outils basés sur des algorithmes d'optimisation permettant de faciliter cette tâche sachant que les contraintes considérées peuvent différer d'un problème à un autre suivant les caractéristiques attendues du tableau de service recherché.

L'objectif de ce travail est de trouver un tableau de service des enseignants d'un lycée privé, qui minimise le nombre d'heures creuses en respectant toutes les contraintes.

Notre mémoire est structuré comme suit :

- ✓ Dans le premier chapitre, nous allons présenter quelques généralités sur la confection des tableaux de service des enseignants de l'enseignement secondaire, en particulier la présentation de la séquence historique des études réalisées sur ce sujet, et des termes connus au domaine pédagogique.
- ✓ Dans Le second chapitre, nous allons présenter le principe des algorithmes génétiques et nous allons expliquer les différentes étapes suivies pour obtenir une meilleure solution.
- ✓ Dans Le troisième chapitre, nous allons formuler le problème de confection de tableau de service des enseignants d'un lycée privé.
- ✓ dans le quatrième chapitre nous allons appliquer les algorithmes génétiques au problème d'un lycée privé en utilisant la programmation en langage C.
- ✓ Et nous allons clôturer par une conclusion.

# *Chapitre 1 : Généralités sur la confection des tableaux de service des enseignants dans un établissement d'enseignement secondaire*

---

## *Introduction*

La confection des tableaux de service dans les établissements scolaires est un travail très important et difficile à réaliser ; c'est typiquement un problème de résolution de contraintes, NP-complet, dont la solution n'est pas à priori connue dans le cas général.

Fournir une solution, nécessite d'être capable de s'adapter aux changements dynamiques de l'environnement en tenant compte de la diversité des contraintes.

Dans ce chapitre nous définissons un tableau de service des enseignants, présentons les deux catégories des contraintes, puis nous présentons la séquence historique des études réalisées sur ce sujet, et par la suite nous citons plusieurs termes connus au domaine pédagogique que nous utiliserons dans la suite de ce mémoire.



## *1. définition*

Le tableau de service est l'organisation des enseignants travaillant au sein d'un établissement scolaire. Il s'agit d'une gestion complexe vue les contraintes multiples et les paramètres dont il faut tenir compte. Trouver une salle libre au jour et à l'horaire où l'enseignant est disponible et où le groupe des étudiants ne suit pas un autre cours

Cette tâche n'est pas aisée. Se complique davantage lorsqu'il s'agit d'un établissement dépassant ses capacités du point de vue effectif des élèves et enseignants. Dans ce cas, le travail manuel de préparation de l'emploi du temps devient quasi-impossible et les responsables se trouvent obligés de faire fonctionner leurs établissements jusqu'aux heures tardives et aussi, à répartir les emplois du temps des enseignants sur plusieurs journées, ce qui les empêche de se concentrer sur leurs activités de recherche ou bien même de ramener des étudiants pour ne suivre qu'une seule séance pendant toute une journée.

Les contraintes ne se limitent pas à ce que nous venons de citer, elles sont plus nombreuses et nous les découvrirons dans la suite du présent rapport.

## *Types de plannings dans le domaine de la pédagogie*

La confection d'horaires (ou confection de tableau de service) dans les établissements scolaires est un travail très important, difficile à réaliser ; c'est typiquement un problème de résolution de contraintes, NP-complet, dont la solution n'est pas, a priori connue dans le cas général. Pour fournir une solution, nécessite d'être capable de s'adapter aux changements dynamiques de l'environnement en tenant compte de la diversité des contraintes telles que l'interdépendance des programmes d'enseignement, la multitude des matières étudiées, la durée des cours, les contraintes de disponibilité des enseignants, la disponibilité limitée des salles. C'est un problème qui peut être défini comme un problème qui fait assigner quelques événements dans un nombre limité de périodes [1].

Ces problèmes sont soumis à beaucoup de contraintes qui sont d'habitude divisées en deux catégories :

- La première regroupe **les contraintes dures** (c'est-à-dire un emploi du temps qui ne satisfait pas ce type de contraintes est impossible ou inadmissible)

- La deuxième catégorie regroupe des contraintes (souples ou de privilège) dont la satisfaction a différents degrés d'importances mais dont le non-respect n'empêche pas une application plus ou moins acceptable de l'emploi du temps trouvé.

Automatiquement ces contraintes sont utilisées pour exprimer ce que doit être un « bon » emploi du temps. Ces contraintes sont plus difficiles à formaliser que les contraintes dures et leur traitement est plus délicat. Ainsi la majorité des approches existantes reposent les contraintes de préférence et les introduisent comme une fonction objective dont l'optimisation permet de se rapprocher le plus possible de la satisfaction des contraintes.

La confection de plannings d'horaires est donc une tâche très difficile et sa solution manuelle peut exiger beaucoup d'effort ce qui a attiré énormément l'attention de la communauté scientifique.

Dans la mesure où notre travail se rapporte au problème de gestion de tableau de service des enseignants au sein d'un établissement privé d'enseignement secondaire, qui est lié automatiquement à la gestion d'un emploi du temps pour les élèves, il semble naturel de refaire une synthèse des différentes études retrouvées dans la littérature et afférentes à ce thème

## **2. Historique**

Une large variété d'approches et modèles ont été proposés pour traiter une variété de problèmes d'emploi du temps. Les problèmes s'étendent de la construction des emplois du temps semestriels ou annuels dans les universités, écoles ou collèges aux emplois du temps d'examens à la fin de ces périodes.

Les premières activités d'emploi du temps ont été effectuées manuellement et un emploi du temps typique, une fois construit est resté statique avec seulement quelques changements nécessaires.

Cependant la nature des enseignements a changé considérablement au cours des années et ainsi les exigences en matière de confection d'emploi du temps sont devenues beaucoup plus compliquées qu'ils ont eu l'habitude de l'être. Par conséquent, le besoin de la

génération automatisée d'emploi du temps augmente et ainsi le développement d'un système de génération d'emploi du temps qui produit des solutions valables est essentiel.

En conséquence, pendant les 30 dernières années, beaucoup d'approches liées à l'automatisation des emplois du temps ont été publiées aux conférences et journaux. De plus, plusieurs applications ont été développées et mises en œuvres avec divers succès [10]. Les premières techniques employées dans la résolution du problème d'emploi du temps ont été basées sur la simulation de l'approche humaine dans la résolution du problème, ces techniques ont été appelées « les heuristiques directes », elles sont basées sur l'idée de créer un emploi du temps partiel en planifiant d'abord le cours le plus contraint, ensuite, cette solution partielle est étendue jusqu'à ce que tous les cours soient planifiés.

L'étape suivante était l'application de techniques générales telles que la programmation linéaire et la coloration de graphes pour résoudre ce problème d'emploi du temps. De là, les premières publications sur la construction d'emploi du temps employant ces techniques générales sont attribuées à Kuhn et Haynes [8].

L'intérêt de génération d'emploi du temps a augmenté dramatiquement dans les années 60 principalement en la raison de la disponibilité d'ordinateurs pour exécuter les algorithmes développés. Autour de la fin des années 60 quelques tentatives qui ont traité le problème en considérant des études de cas commençaient à être publiées. Par exemple en 1969, Lawrie a développé un modèle pour le problème de confection d'horaire en employant l'approche de programmation linéaire. Pendant les années 1970, plusieurs publications ont abordé le problème d'emploi du temps. Les principales techniques qui semblent avoir été plus répandues dans les années 1970 et les années 1980 sont les techniques ayant pour racine l'intelligence artificielle et sont basées sur les méthodes du recuit simulé, la recherche tabou et **les algorithmes génétiques**.

En 1985, De Werra, a décrit les divers problèmes traitant le problème d'emploi du temps d'une façon formelle et a fourni les différentes formulations dans une tentative de les résoudre. Il a aussi décrit les approches considérées les plus importantes à ce temps-là [4].

En 1986, Carter, a fait une analyse sur de réelles applications de confection d'emploi du temps de plusieurs universités. Junginger, a décrit dans la même année, les recherches faites en Allemagne sur le problème d'emploi du temps scolaires et les approches qui étaient

basées sur des heuristiques directes, en particulier il a décrit les divers logiciels mis en oeuvre et leur utilisation dans les divers établissements.

En 1994, Corne, a fait une enquête sur l'application des algorithmes génétiques au problème d'emploi du temps et a discuté les futures perspectives de telles approches en comparant les résultats obtenus avec ceux obtenus avec d'autres approches.

Bien qu'il y ait des publications dans les années 1990 sur la résolution du problème d'emploi du temps en employant les techniques basées sur l'IA, il y avait une l'apparition d'une nouvelle approche, aussi enracinée dans l'IA appelée la programmation de satisfaction de contraintes (CSP).

Dans les dernières décennies, les sujets de résolution du problème d'emploi du temps ont été principalement limités à la (RO) (les techniques employées étaient naturellement mathématiques). Dans la décennie actuelle, la contribution de l'IA (l'intelligence artificielle) a fourni au problème de résolution de l'emploi du temps une heuristique moderne telle que les algorithmes génétiques, le recuit simulé et la recherche tabou [10].

À la suite de ce mémoire, nous allons utiliser plusieurs vocabulaires connus au domaine pédagogique, essayons tout d'abord de les définir.

### **3. Notions de base**

#### **➤ Les entités temporelles**

- Une date : désigne un instant défini par un triplet (Jour, Mois, Année), à partir de ce triplet, on détermine la valeur qui lui est associée sur l'axe des jours.
- Une durée : est un nombre compris entre  $D_{\min}$  et  $D_{\max}$ , tel que  $D_{\min}$  représente la plus petite unité temporelle disponible, et  $D_{\max}$  représente la plus grande unité temporelle disponible.
- Un créneau : horaire désigne un intervalle temporel dans une journée. Ainsi, un créneau est caractérisé par un couple (H, D) où :
  - **H** représente l'heure de début du créneau.
  - **D** sa durée.

- Un calendrier : est un ensemble de dates auxquelles on associe un état ou une valeur parmi (disponible ou non disponible)

### ➤ *Les séances et les réservations*

- Une séance : correspond à une instance temporelle d'un enseignement à une date donnée, pendant un créneau précis, les caractéristiques d'une séance sont :
  - Son Enseignement.
  - Sa Date
  - Son Créneau
  - Ses Matériels
  - Sa Salle.
- Une réservation : correspond à une option posée sur l'occupation de cette ressource, les ressources considérées sont les entités physiques nécessaires à l'élaboration des emplois du temps, il s'agit des :
  - Salles
  - Enseignants
  - Groupe
  - Elèves
  - Matériels.

### **4. Les contraintes**

D'une manière générale l'affectation des matières, enseignants, locaux à des périodes est soumise à des contraintes qui diffèrent selon leurs priorités, une contrainte ne revêt pas nécessairement un aspect absolu soit elle est vérifiée ou (violée), mais peut être formulée sous forme d'un objectif qui doit être approché autant que possible, selon ce critère, les contraintes peuvent être réparties en deux grandes classes :

- Les contraintes dures (absolues).
- Les contraintes de préférences (souples).

### ➤ *Les contraintes dures*

Ce type de contraintes doit être obligatoirement satisfait dans toutes les situations, car la violation de l'une de ces contraintes rend l'emploi du temps inefficace dans la réalité, on peut distinguer les contraintes dures suivantes

- Un enseignant ne peut pas être affecté à deux séances différentes à la même période
- Une salle ne peut pas accueillir deux séances différentes à la même période
- Une matière doit respecter le nombre d'heures hebdomadaires c'est-à-dire si une matière est enseignée trois fois par semaine à une classe, alors il doit apparaître trois fois dans l'emploi du temps d'enseignant affecté à cette classe.
- Un emploi du temps doit comporter toutes les matières d'une classe
- La charge journalière d'un enseignant ne doit pas être dépassée.

### ➤ *Les contraintes de préférence*

Contrairement au type de contraintes précédent, les contraintes de préférences n'exigent pas la vérification stricte, mais d'approcher au maximum de l'objectif voulu, ces contraintes sont utilisées pour exprimer ce que doit être un «Bon » emploi du temps, sont plus difficiles à formaliser que les contraintes dures et leur traitement est plus délicats, ainsi la majorité des approches existantes relaxent les contraintes de préférence et les introduisent comme une Fonction objectif dont l'optimisation permet de se rapprocher le plus possible de la satisfaction des contraintes.

Par exemple :

- Essayer d'éviter aux (enseignants ou élèves) des pertes de temps par de trop Longs espacements entre deux séances d'une même journée (minimiser le nombre d'heures creuses)
- Éviter que certains jours se trouvent surcharger alors que d'autres le sont moins
- Les matières de coefficient minimal ne doivent pas occuper les séances de la matinée d'une journée donnée, au détriment des matières de coefficients élevés
- Eviter d'affecter une période jugée non convenable à un enseignant, sauf si cela est inévitable

- Minimiser les déplacements des élèves dans l'établissement
- Libérer quelques après-midi pour les enseignants.

D'après les différents types des contraintes et les différents exemples cités un bon tableau de service pour les enseignants (ou un bon emploi du temps pour les élèves) influe positivement sur la fiabilité d'enseignant (niveau d'acquisition des élèves).

## ***Conclusion***

Dans ce chapitre nous avons présenté tout ce qui est en relation avec la confection des tableaux de service des enseignants (respectivement emploi du temps des élèves) dans les établissements scolaires, précisément les différents types de contraintes avec des exemples qui montrent l'importance d'un bon tableau de service (respectivement un bon emploi de temps), et son influence sur la fiabilité des enseignants (respectivement niveau d'acquisition des élèves)

## *Chapitre 2 : Les Algorithmes Génétiques*

---

### *Introduction*

Comme il est indiqué au chapitre 1, parmi les méthodes utilisées dans la résolution des problèmes d'emploi du temps, nous trouvons les algorithmes génétiques (AG), qui servent à la résolution des problèmes d'optimisation.

Dans ce chapitre nous essayerons de présenter les algorithmes génétiques, en particulier leur principe, les différentes étapes suivies à la construction des solutions, et les critères approuvés pour trouver ou générer la meilleure solution.



Les algorithmes génétiques (AG) sont des méthodes utilisées dans les problèmes d'optimisation. Les AG tirent leur nom de l'évolution biologique des êtres vivants dans le monde réel. Ces algorithmes cherchent à simuler le processus de la sélection naturelle dans un environnement défavorable en s'inspirant de la théorie de l'évolution proposée par C. Darwin. Dans un environnement, « **les individus** » les mieux adaptés tendent à vivre assez longtemps pour se reproduire alors que les plus faibles ont tendance à disparaître. Par analogie avec l'évolution naturelle, les AG font évoluer un ensemble de solutions candidates, appelé une « **population d'individus** ». Un « individu » n'est autre qu'une solution possible du problème à résoudre. Chaque individu de cette population se voit attribuer une fonction appelée fonction d'adaptation (**fitness**) qui permet de mesurer sa qualité ou son poids; cette fonction d'adaptation peut représenter la fonction objectif à optimiser. Ensuite, les meilleurs individus de cette population sont sélectionnés, subissent des croisements et des mutations et une nouvelle population de solutions est produite pour la génération suivante. Ce processus se poursuit, génération après génération, jusqu'à ce que le critère d'arrêt soit atteint, comme par exemple le nombre maximal de générations

## ***1. Algorithmes génétiques***

Les techniques de recherche et d'optimisation sont en général classées en trois catégories [2] énumératives, déterministes et stochastiques. Les AG font partie de la troisième catégorie et quatre caractéristiques les distinguent des autres techniques d'optimisation [7]

- ils utilisent un codage des paramètres et non les paramètres eux-mêmes;
- ils travaillent sur une population d'individus (ou de solutions);
- ils n'utilisent que les valeurs de la fonction à optimiser, pas sa dérivée, ou une autre connaissance auxiliaire;
- ils utilisent des règles de transition probabilistes et non déterministes.

## ***Principe de base d'un AG standard***

Un **AG** standard nécessite en premier **le codage** de l'ensemble des paramètres du problème d'optimisation en une chaîne de longueur finie. Le principe d'un AG est simple, il s'agit de simuler l'évolution d'une population d'individus jusqu'à un critère d'arrêt.

On commence par générer **une population initiale** d'individus (solutions) de façon aléatoire. Puis, à chaque génération, des individus sont sélectionnés, cette sélection est effectuée à partir

d'une fonction objectif appelée *fonction d'adaptation* (fitness) Puis, les opérateurs de *croisement* et de *mutation* sont appliqués et une nouvelle population est créée.

Ce processus est itéré jusqu'à un critère d'arrêt. Le critère le plus couramment utilisé est le nombre maximal de générations que l'on désire effectuer.

L'AG débute par :

- la génération d'une population initiale et l'évaluation de la fonction d'adaptation de tous les individus qui composent cette première population.
- Puis, des individus sont sélectionnés aléatoirement pour la reproduction selon le principe de la survie du plus adapté.
- Ensuite, des individus « enfants » (ou les descendants) sont générés en appliquant les deux opérateurs génétiques suivants :
  - le croisement
  - la mutation.

Ces enfants sont placés dans une nouvelle population  $P(t)$  et vont se substituer, en tout ou en partie, à la population de la génération précédente. De nouvelles populations d'individus vont ensuite se succéder, d'une génération ( $t$ ) à la génération ( $t+1$ ), chaque génération représentant une itération jusqu'à l'atteinte du critère d'arrêt. L'AG présenté ci-dessus est dit générationnel car tous les individus enfants générés sont placés dans une population et vont remplacer entièrement la population des individus parents.

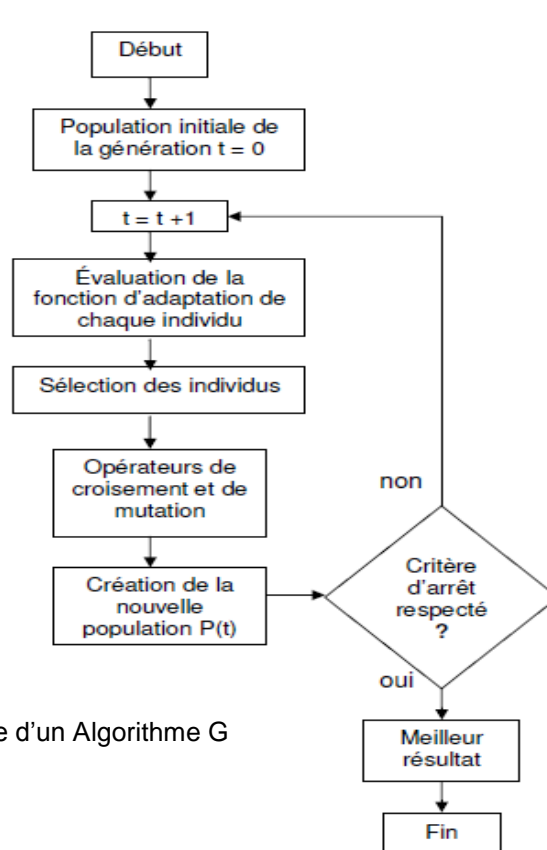


Figure 1 : Organigramme d'un Algorithme G génétique standard

## 1.1. Codage

Chaque paramètre d'une solution est assimilé à un gène, toutes les valeurs qu'il peut prendre sont les allèles de ce gène, on doit trouver une manière de coder chaque allèle différent de façon unique (établir une bijection entre l'allèle "réel" et sa représentation codée).

Un chromosome est une suite de gène, on peut par exemple choisir de regrouper les paramètres similaires dans un même chromosome (chromosome à un seul brin) et chaque gène sera repérable par sa position : son locus sur le chromosome en question.

Chaque individu est représenté par un ensemble de chromosomes, et une population est un ensemble d'individus.

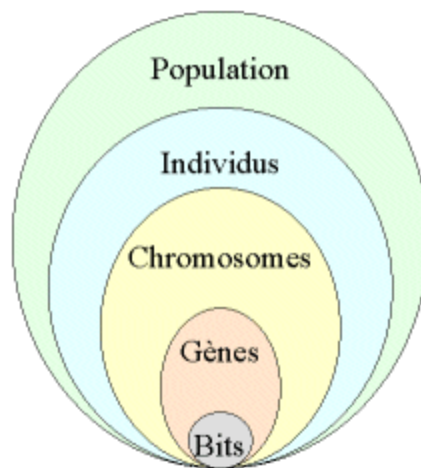


Figure 2: les cinq niveaux d'organisation d'un algorithme génétique

Il y a trois principaux types de codage utilisables, et on peut passer de l'un à l'autre relativement facilement :

- le codage binaire : c'est le plus utilisé. Chaque gène dispose du même alphabet binaire  $\{0, 1\}$ , un gène est alors représenté par un entier long (32 bits), les chromosomes qui sont des suites de gènes sont représentés par des tableaux de gènes et les individus de notre espace de recherche sont représentés par des tableaux de chromosomes.

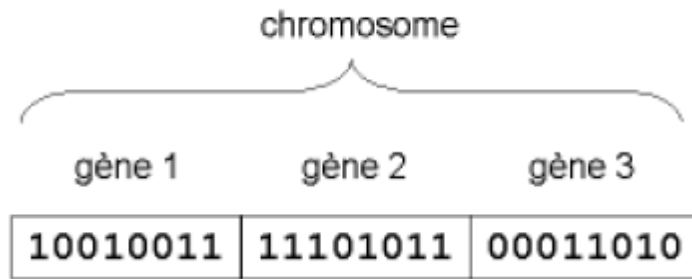


Figure 3 : exemple d'un chromosome

Ce cas peut être généralisé à tout alphabet allélique n-aire permettant un codage plus intuitif, par exemple pour le problème du voyageur de commerce on peut préférer utiliser l'alphabet allélique  $\{c_1, c_2, c_3, \dots, c_n\}$  où  $c_i$  représente la ville de numéro  $i$ .

- le codage réel : cela peut-être utile notamment dans le cas où l'on recherche le maximum d'une fonction réelle.
- le codage de Gray

Il existe de nombreuses méthodes pour la sélection, le croisement et la mutation.

## 1.2. Sélection

La sélection a pour objectif d'identifier les individus qui doivent se reproduire. Cet opérateur ne crée pas de nouveaux individus mais identifie les individus sur la base de leur fonction d'adaptation, les individus les mieux adaptés sont sélectionnés alors que les moins bien adaptés sont écartés [3].

La sélection doit favoriser les meilleurs éléments selon le critère à optimiser (minimiser ou maximiser). Ceci permet de donner aux individus dont la valeur est plus grande une probabilité plus élevée de contribuer à la génération suivante.

Il existe plusieurs méthodes de sélection, les plus connues étant la « roue de la fortune » et la « sélection par tournoi » :

- La « **roue de la fortune** » est la plus ancienne, où chaque individu, de la population de taille maximale  $J_{max}$ , occupe une section de la roue proportionnellement à sa fonction d'adaptation  $Fitness(j)$ , la probabilité de sélection d'un individu ( $j$ ) s'écrit :

$$\text{Prob}(j) = \frac{\text{Fitness}(j)}{\sum_{j=1}^{J_{\max}} \text{Fitness}(j)}$$

À chaque fois qu'un individu doit être sélectionné, un tirage à la loterie s'effectue et propose un candidat, les individus possédant une plus grande fonction d'adaptation ayant plus de chance d'être sélectionnés.

- A chaque fois qu'il faut sélectionner un individu, la « sélection par tournoi » consiste à tirer aléatoirement ( $k$ ) individus de la population, sans tenir compte de la valeur de leur fonction d'adaptation, et de choisir le meilleur individu parmi les  $k$  individus. Le nombre d'individus sélectionnés a une influence sur la pression de sélection, lorsque  $k = 2$ , la sélection est dite par « tournoi binaire ».

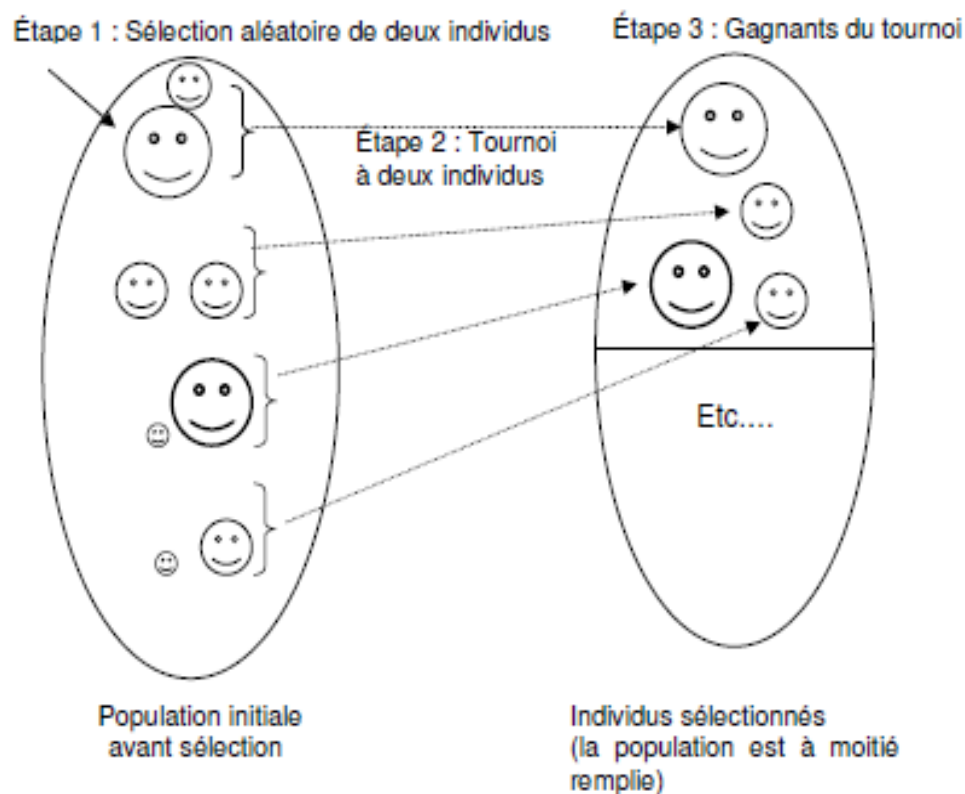


Figure 4 : Représentation d'une sélection par tournoi d'individus pour un critère de maximisation Chaque individu représente une solution possible

### 1.3.Croisement

Le croisement permet de créer de nouvelles chaînes en échangeant de l'information entre deux chaînes. Le croisement s'effectue en deux étapes.

- D'abord les nouveaux éléments produits par la reproduction sont appariés
- ensuite chaque paire de chaînes subit un croisement comme suit :
  - un entier  $k$  représentant une position sur la chaîne est choisi aléatoirement entre 1 et la longueur de chaîne ( $l$ ) moins un ( $l - 1$ ).
  - Deux nouvelles chaînes sont créées en échangeant tous les caractères compris entre les positions  $k + 1$  et  $l$  inclusivement.

L'exemple suivant montre deux chaînes (A1 et A2) de longueur  $l = 5$  appartenant à la population initiale. Les deux nouvelles chaînes (A3 et A4) appartenant à la nouvelle population sont obtenues par croisement à la position  $k = 4$ .

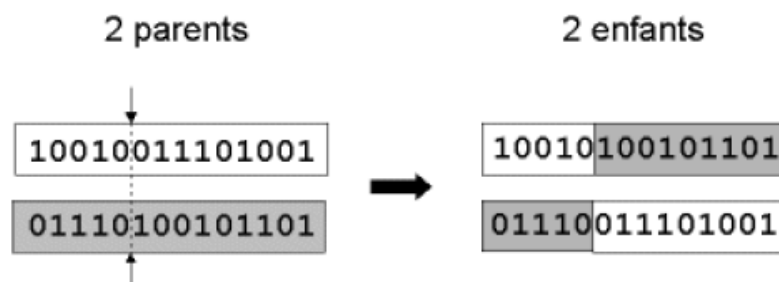


Figure 5 : Représentation d'un croisement en un point de deux chaînes

Il existe d'autres opérateurs de croisement :

- **Croisement en deux points** : on choisit au hasard deux points de croisement et on échange les parties de chaîne situées entre ces deux points .

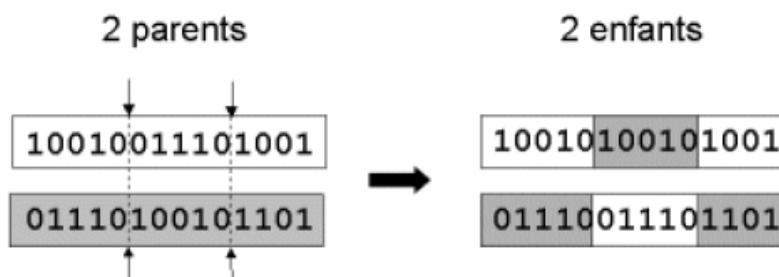


Figure 6 : Représentation d'un croisement en deux points

- **Croisement uniforme** : dans ce type de croisement, on utilise un masque de croisement (mask), qui consiste en un vecteur généré aléatoirement, de longueur identique aux chaînes parents, et composé de 0 et 1. Lorsque le bit du masque vaut 0, l'enfant hérite le bit du premier parent, sinon il hérite de celui du second parent. Le second enfant est le complémentaire du premier. Ce croisement peut être considéré comme une généralisation du croisement multipoint sans connaissance préalable du point de croisement.

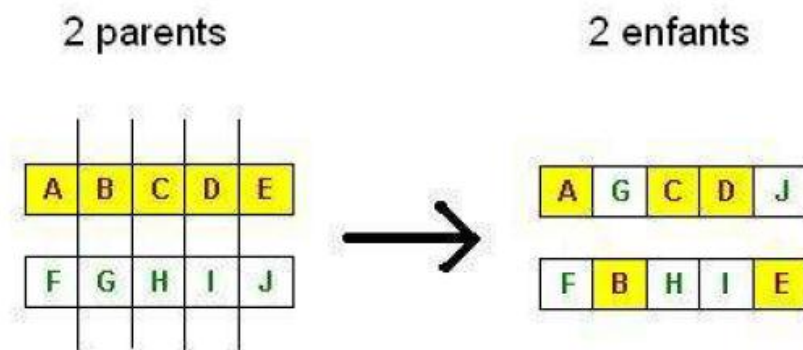


Figure 7 : Représentation d'un croisement uniforme

#### 1.4. Mutation

La mutation est exécutée seulement sur une seule chaîne. Elle représente la modification aléatoire et occasionnelle de faible probabilité de la valeur d'un caractère de la chaîne, pour un codage binaire cela revient à changer un 1 en 0 et vice versa, Cet opérateur introduit de la diversité dans le processus de recherche des solutions et peut aider l'AG à ne pas stagner dans un optimum local.

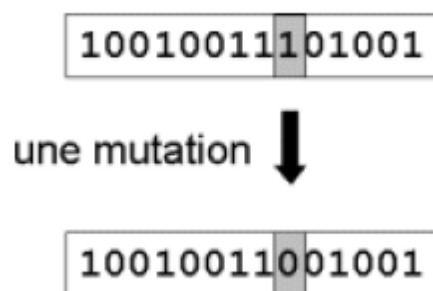


Figure 8: Représentation d'une mutation de bits dans une chaîne

## ***2. Paramètres d'un AG***

Pour appliquer un AG à un problème réel, on doit posséder les éléments suivants :

1. Initialiser la population initiale P.
2. Evaluer P.
3. Tant Que (Pas Convergence) faire :
  - P' = Sélection des Parents dans P
  - P' = Appliquer Opérateur de Croisement sur P'
  - P' = Appliquer Opérateur de Mutation sur P'
  - P = Remplacer les Anciens de P par leurs Descendants de P'
  - Evaluer P

Fin Tant Que

Le critère de convergence peut être de nature diverse, par exemple :

- Un taux minimum qu'on désire atteindre d'adaptation de la population au problème,
- Un certain temps de calcul à ne pas dépasser,
- Une combinaison de ces deux points.

## ***Conclusion***

Nous avons présenté dans ce chapitre le principe d'algorithmes génétiques, et leurs différentes étapes afin d'être utilisées (dans la suite de notre mémoire), dans la résolution d'un problème de construction d'un tableau de service des enseignants d'un lycée privé.



# *Chapitre 3 : formulation du problème de confection des tableaux de service des enseignants dans un lycée privé*

---

## ***Introduction***

Dans ce chapitre, nous considérons le problème de confection des tableaux de service des enseignants, comme un problème d'optimisation mono-objectif (il a une seule fonction objective), Jusqu'à maintenant ce sont principalement les modèles utilisés lors de la résolution de ce type de problèmes.

Ainsi nous expliquons le problème ou le travail demandé, précisément les données nécessaires au cours de la construction d'un tableau de service, les variables utilisées pour formuler les contraintes exigées et l'objectif à atteindre.

## ***1. Position du problème***

Étant donné un ensemble d'informations d'un lycée privé (nombre de classes, nombre des salles, nombre d'enseignants, spécialités d'enseignants,...), le problème consiste à affecter les séances nécessaires pour chaque classe et vérifier les données de telle sorte que le nombre de salles et d'enseignants soit suffisant, chaque classe doit prendre les matières de son programme, et ne dépasse pas le nombre d'heures exigées pour chaque matière.

L'objectif est minimiser le nombre d'heures creuses dans les tableaux de service des enseignants, essayons tout d'abord de définir ou expliquer une heure creuse.

### ***1.1. Définition***

Une heure creuse (une heure de faible activité) n'est pas forcément au milieu des séances, bien au contraire, un emploi du temps d'un enseignant qui comporte une heure de cours de 8h à 9h ou de 11h à midi pendant toute la matinée, ou encore une heure de 17h à 18h pendant tout l'après-midi ; Ce cas de figure pourrait donner lieu à un déséquilibre dans le temps de travail d'un enseignant puisqu'il est obligé de se rendre à l'école pour seulement une heure, pendant la matinée ou l'après-midi.

### ***1.2. La vérification de la suffisance d'enseignants et des salles***

Cette partie est très importante d'où nous vérifions la faisabilité de tableau de service des enseignants (respectivement d'emploi du temps des élèves) à partir des calculs concernant :

- le nombre d'heures pour chaque enseignant pendant la semaine.
- le nombre de salles nécessaire pour que tous les classes prennent leurs cours et utilisent le même types des salles (générales ou spéciales) pendant un créneau précis.

#### ***➤ La vérification de nombre d'heures de travail pour chaque enseignant***

Nous faisons la somme d'heures de chaque matière ***m*** pendant une semaine, puis on la divise par le nombre d'enseignants disponibles pour cette matière et la compare avec  $(\sum_{i \in G_m} NH_i)$  :

- si le résultat est inférieur ou égal à  $(\sum_{i \in G_m} NH_i)$  alors, le nombre d'enseignants disponibles est suffisant.
- sinon le nombre d'enseignants disponibles est insuffisant.

### ➤ *La vérification des salles*

Pour chaque type de salles, nous comparons le nombre de séances qui peuvent se dérouler pendant un créneau  $k$  avec le nombre de salles disponibles pour le même type de salles, il est préférable que le résultat soit inférieur ou égal au nombre de salles disponibles, car sinon on peut trouver des problèmes au niveau de construction des tableaux de service.

## **2. Modélisation**

La modélisation d'un problème réel, consiste à le rendre sous forme d'un ensemble d'équations mathématiques. Dans cette partie nous présentons la modélisation du problème de confection des tableaux de service des enseignants d'un lycée privé. Tel que la recherche du modèle mathématique d'un problème revient à identifier les Composantes suivantes :

- Les données.
- Les variables
- Les contraintes
- La fonction objective (objectif).

### **2.1. Les données**

- $N$  : nombre d'enseignants
- $NS$  : nombre de salles générales
- $NS_m$  : nombre de salles réservées à la matière  $m$
- $C$  : nombre de classes
- $M$  : nombre de matières
- $NCR$  : nombre de créneaux

- Chaque classe a un nombre d'heures hebdomadaire d'une matière  $m$  noté  $NM_{mj}$
- Chaque enseignant a une spécialité ( $m$ ) et un nombre d'heures de travail précis  $NH_i$
- La disponibilité des enseignants.
- $NH_j$  : nombre d'heures hebdomadaires pour la classe  $j$

On va regrouper les enseignants selon leurs spécialités

$$G_1 = \{1, \dots, \}$$

$$G_2 = \{., \dots, \}$$

$$\vdots$$

$$G_M = \{., \dots, N\}$$

## 2.2. Variables

Pour les variables nous avons besoins de deux variables la première relie trois acteurs principaux l'enseignant  $i$ , la classe  $j$  et le créneau  $k$ , et il nous permet de savoir est ce qu'une classe  $j$  a une séance avec un enseignant  $i$  pendant un créneau  $k$  ou non.

$$x_{ij}^k = \begin{cases} 1 & \text{si l'enseignant } i \text{ enseigne la classe } j \text{ pendant le créneau } k \\ 0 & \text{sinon} \end{cases}$$

La deuxième variable nous permet de connaître les classes affectées à un enseignant, ce qui est très important pour formuler la contrainte qui exige qu'une classe  $j$  doit être affectée à un seul enseignant de même spécialité.

$$y_{ij} = \begin{cases} 1 & \text{si la classe } j \text{ est affectée à l'enseignant } i \\ 0 & \text{sinon} \end{cases}$$

### 2.3. Les contraintes

Dans ce problème nous disposons de plusieurs types de contraintes :

- contraintes de disponibilité ;
- contraintes de vérification de nombre d'heures exigées ;

Prenons en considération tour à tour chacune de ces contraintes.

#### ➤ La disponibilité des salles

Le nombre total de salles générales réservées pendant un créneau  $k$ , ne doit pas dépasser le nombre de salles générales disponibles dans le lycée, car sinon on trouvera qu'un enseignant ne peut pas donner son cours, ou il est obligé de partager une salle avec un autre enseignant ce qui est anti-pédagogique, et pour vous approcher de la formule de cette contrainte voici une explication.

#### Explication :

Pour chaque créneau, nous vérifions que la somme de toutes les séances qui se déroulent (dans des salles générales) est inférieure ou égale au nombre de salles disponibles, À partir des variables qu'on a :

$$x_{ij}^k = \begin{cases} 1 & \text{si l'enseignant } i \text{ enseigne la classe } j \text{ pendant le créneau } k \\ 0 & \text{sinon} \end{cases}$$

La somme  $\sum_{j=1}^C x_{ij}^k$  égale 1 si l'enseignant  $i$  a une séance pendant le créneau  $k$  dans une salle générale, et 0 sinon, alors la double somme  $\sum_{i=1}^N \sum_{j=1}^C x_{ij}^k$  donne le nombre de séances qui se déroulent dans les salles générales pendant le créneau  $k$ , donc d'après la définition de cette contrainte, il faut vérifier que cette somme ne dépasse pas le nombre de salles générales  $NS$  d'où la formule de la contrainte:

$$\sum_{i=1}^N \sum_{j=1}^C x_{ij}^k \leq NS \quad \forall k \in \{1, 2, \dots, NCR\} \quad (1)$$

Et puisqu'il ya des matières qui ont des salles spéciales, par exemple la matière de physique chimie qui a des salles avec laboratoire, de même la matière de science de la vie et la terre, et la salle du sport qui a une capacité maximal, nous ajoutons la formule suivante :

$$\sum_{i \in G_m} \sum_{j=1}^C X_{ij}^k \leq NS_m \quad \forall k \in \{1, 2, \dots, NRC\} \quad \forall m \text{ une matière a des salles spéciales} \quad (2)$$

Où  $G_m$  représente le groupe qui contient tout les enseignants de la matière  $m$ , donc la double somme  $\sum_{i \in G_m} \sum_{j=1}^C X_{ij}^k$  égale le nombre de séances qui se déroulent dans les salles de la matière  $m$  pendant le créneau  $k$ , et pour que la contrainte de disponibilité des salles soit vérifier il faut que cette somme ne dépasse pas le nombre de salles de la matière  $m$ , notée  $NS_m$ , d'où la formule ci-dessus.

#### ➤ *Nombre d'heures de travaille exigé par chaque enseignant*

Chaque enseignant  $i$  a des heures où il est disponible et des autres où il est indisponible, ainsi chaque enseignant  $i$  a un nombre précis d'heures de travail hebdomadaires  $NH_i$  ne doit pas le dépassé, donc nous avons besoin d'une contrainte pour vérifier que le nombre des séances affecté a cet enseignant inferieur ou égal à  $NH_i$ .

#### Explication :

Pour chaque enseignant  $i$ , nous vérifions que la somme de ses séances hebdomadaire égale  $NH_i$ , d'où la somme  $\sum_{j=1}^C x_{ij}^k$  égale 1 si l'enseignant  $i$  a une séance pendant le créneau  $k$ , et égale 0 sinon, donc la double somme  $\sum_{k=1}^{NCR} \sum_{j=1}^C x_{ij}^k$  représente le nombre des séances hebdomadaire d'enseignant  $i$ , d'après la définition de la contrainte cette somme doit être inferieur ou égale  $NH_i$  :

$$\sum_{k=1}^{NCR} \sum_{j=1}^C x_{ij}^k \leq NH_i \quad \forall i \in \{1, 2, \dots, N\} \quad (3)$$

➤ *Une classe doit être affectée à un seul enseignant pendant un créneau  $k$*

Il est clair qu'une classe  $j$  ne peut pas suivre deux séances différentes (ou plus) au même temps (créneau), donc nous avons besoin d'une contrainte qui vérifie pendant un créneau  $k$  que chaque classe  $j$ , soit a une séance avec un seul enseignant, soit n'a aucune séance.

D'où, nous vérifions pendant chaque créneau  $k$  que le nombre des enseignants qui ont une séance avec la classe  $j$ , ne dépasse pas 1

$$\sum_{i=1}^N x_{ij}^k \leq 1 \quad \begin{array}{l} \forall j = \{1, 2, \dots, C\} \\ \forall k = \{1, 2, \dots, NCR\} \end{array} \quad (4)$$

Cette contrainte conduit-nous automatiquement à penser à une autre contrainte qui vérifie qu'un enseignant  $i$  ne doit pas être chargé d'enseigner au moins deux classes différentes au même créneau  $k$ .

➤ *un enseignant ne doit pas être chargé d'enseigner plus d'une classe pendant un créneau  $k$*

Cette contrainte vérifiée pendant chaque créneau  $k$  que chaque enseignant  $i$ , soit a une séance avec une seule classe, soit n'a aucune séance pendant ce créneau.

D'où, nous vérifions pendant chaque créneau  $k$  que le nombre de classes qui ont une séance avec l'enseignant  $i$ , ne dépasse pas 1

$$\sum_{j=1}^C x_{ij}^k \leq 1 \quad \begin{array}{l} \forall i = \{1, 2, \dots, N\} \\ \forall k = \{1, 2, \dots, NCR\} \end{array} \quad (5)$$

Tel que la somme  $\sum_{j=1}^C x_{ij}^k$  égale 0 si l'enseignant  $i$  n'a aucune séance pendant le créneau  $k$ , et elle est différente de zéro si l'enseignant  $i$  a au moins une séance pendant ce créneau.

➤ *Chaque classe doit être enseignée par un seul enseignant de même spécialité*

Pour maintenir le parcours normal des études, et pour ne mettre pas l'élève devant le problème d'être enseigné par plusieurs enseignants de même matière (spécialité), ce qui influe négativement sur la compréhension de cours pour les élèves .

Et pour formuler cette contrainte nous utilisons la deuxième variable

$$y_{ij} = \begin{cases} 1 & \text{si la classe } j \text{ est affecté à l'enseignant } i \\ 0 & \text{sinon} \end{cases}$$

Alors pour une matière  $m$ , la somme  $\sum_{i \in G_m} y_{ij}$  égale 1 si la classe  $j$  est affecté à un seul enseignant de spécialité (matière)  $m$ , 0 si la matière  $m$  n'existe pas dans le programme de la classe  $j$ , donc pour que la contrainte soit vérifiée il suffit que cette somme ne dépasse pas 1, d'où la formule :

$$\sum_{i \in G_m} y_{ij} \leq 1 \quad \begin{matrix} \forall m \text{ une matière} \\ \forall j \in \{1, 2, \dots, M\} \end{matrix} \quad (6)$$

➤ *Vérification de nombre d'heures hebdomadaires*

D'après le **calendrier 43** qui organise l'étude et la distribution des matières et des séances pour chaque classe dans l'enseignement secondaire au Maroc, nous associons à chaque classe  $j$  un nombre d'heures hebdomadaire précis d'une matière  $m$  et on le note  $NH_{mj}$ , donc le rôle de cette contrainte est de vérifier, pendant la construction de tableau de service, que le nombre d'heures d'une matière affecté à une classe soit égal le nombre posé ou exigé dans le calendrier 43.

Pour formuler cette contrainte, on fixe un enseignant  $i$  de spécialité  $m$  et une classe  $j$ , et on fait une sommation sur tout les créneaux cette somme représente le nombre d'heures hebdomadaire de l'apprentissage de la matière  $m$  par la classe  $j$  donc pour vérifier la contrainte il faut que cette somme soit égale à  $NH_{mj}$  :

$$\sum_{k=1}^{NCR} x_{ij}^k = NH_{mj} \quad \begin{matrix} \forall i = \{1, 2, \dots, N\} \text{ tel que } m \text{ sa spécialité} \\ \forall j = \{1, 2, \dots, C\} \end{matrix} \quad (7)$$



De même chaque classe  $j$  a un nombre d'heures d'étude précis noté  $NH_j$ , et puisque  $NH_j$  égal la somme des heures de chaque matière existe dans programme de la classe  $j$

$$\sum_{m=1}^M NH_{mj} = NH_j \quad \forall j = \{1, 2, \dots, C\}$$

Donc, la vérification de **(1.3.7)** implique automatiquement la vérification du nombre d'heures hebdomadaires pour chaque classe  $NH_j$ , représentée par la formule suivante :

$$\sum_{i=1}^N \sum_{k=1}^{NCR} x_{ij}^k = NH_j \quad \forall j \in \{1, 2, \dots, C\}$$


### ➤ *Contrainte des créneaux*

Cette contrainte impose que chaque élève ne doit pas étudier plus que deux heures d'une matière par jour. Pour traiter ce problème nous faisons des tests sur chaque jour, il faut que la somme sur ses créneaux soit inférieure ou égale à 2, pour chaque matière  $m$  enseigné par un enseignant  $i$ .

Par exemple :

Prenons les horaires du lundi d'enseignant  $i$  avec la classe  $j$

	Lundi							
	Matin				Soir			
créneaux	$CR_1$	$CR_2$	$CR_3$	$CR_4$	$CR_5$	$CR_6$	$CR_7$	$CR_8$
$x_{ij}^k$	1	1	0	0	0	0	0	0



Signifié que l'enseignant  $i$  enseigne la classe  $j$  pendant le créneau  $CR_1$

Signifié que l'enseignant  $i$  n'a aucune séance pendant le créneau  $CR_7$

Tableau 1 : les horaires du lundi d'enseignant  $i$  avec la classe  $j$

Dans ce cas, on n'a pas de problème, car la classe  $j$  ne dépasse pas deux heures de matière enseignée par l'enseignant  $i$  c.-à-d :

$$\sum_{k=1}^8 x_{ij}^k \leq 2$$

Comme il est indiqué auparavant nous partitionnons la semaine en NCR=44 créneaux, du Lundi au samedi avec un nombre de huit périodes (du lundi au vendredi) et un nombre de quatre périodes le samedi, La durée d'une période est une heure, donc il y a 44 créneaux hebdomadairement.

Pour généraliser la contrainte sur toute la semaine, Nous devons éliminer les créneaux qui représentent l'intervalle du temps (8h → 9h), utilisons les tableaux suivants pour les fixés

Lundi								Mardi							
Matin				Soir				Matin				Soir			
<b>CR<sub>1</sub></b>	CR <sub>2</sub>	CR <sub>3</sub>	CR <sub>4</sub>	CR <sub>5</sub>	CR <sub>6</sub>	CR <sub>7</sub>	CR <sub>8</sub>	CR <sub>9</sub>	CR <sub>10</sub>	CR <sub>11</sub>	CR <sub>12</sub>	CR <sub>13</sub>	CR <sub>14</sub>	CR <sub>15</sub>	CR <sub>16</sub>

Mercredi								Jeudi							
Matin				Soir				Matin				Soir			
<b>CR<sub>17</sub></b>	CR <sub>18</sub>	CR <sub>19</sub>	CR <sub>20</sub>	CR <sub>21</sub>	CR <sub>22</sub>	CR <sub>23</sub>	CR <sub>24</sub>	CR <sub>25</sub>	CR <sub>26</sub>	CR <sub>27</sub>	CR <sub>28</sub>	CR <sub>29</sub>	CR <sub>30</sub>	CR <sub>31</sub>	CR <sub>32</sub>

Vendredi								Samedi			
Matin				Soir				Matin			
<b>CR<sub>33</sub></b>	CR <sub>34</sub>	CR <sub>35</sub>	CR <sub>36</sub>	CR <sub>37</sub>	CR <sub>38</sub>	CR <sub>39</sub>	CR <sub>40</sub>	CR <sub>41</sub>	CR <sub>42</sub>	CR <sub>43</sub>	CR <sub>44</sub>

Tableau 2 : les horaires d'une semaine

Donc la formule de la contrainte sera comme suite :

$$\sum_{k=l}^{l+7} x_{ij}^k \leq 2 \quad \begin{array}{l} \forall i \in \{1, 2, \dots, N\} \\ \forall j \in \{1, 2, \dots, C\} \\ \forall l \in \{1, 9, 17, 25, 33\} \end{array} \quad (8)$$

Et pour le samedi matin, nous ajoutons la condition suivante :

$$\sum_{k=41}^{44} x_{ij}^k \leq 2 \quad \begin{array}{l} \forall i \in \{1, 2, \dots, N\} \\ \forall j \in \{1, 2, \dots, C\} \end{array} \quad (9)$$

La vérification de cette contrainte n'empêche pas de tomber dans d'autres cas qui ne sont pas souhaitables :

- Un décalage d'une ou de deux heures entre deux séances de la même matière

	Matin				Soir			
$x_{ij}^k$	1	0	1	0	0	0	0	0

Tableau 3: exemple d'un décalage d'une heure

	Matin				Soir			
$x_{ij}^k$	1	0	0	1	0	0	0	0

Tableau 4 : exemple d'un décalage de deux heures

- Une séance de la même matière peut se dérouler le matin et le soir du même jour

	Matin				Soir			
$x_{ij}^k$	1	0	0	0	1	0	0	0

Tableau 5 : Un décalage de plus de trois heures

## 2.4. La fonction objective

Comme il est indiqué auparavant, nous cherchons à minimiser le nombre d'heures creuses dans le tableau de service des enseignants d'un lycée privé, c'est-à-dire rendre la somme des différences entre deux créneaux consécutifs aussi petite que possible pour tous les enseignants nous écrivons alors :

$$\text{Min} \sum_{i=1}^N \sum_{k=1}^{44} \sum_{j=1}^C \left| \frac{1}{C} \sum_{j=1}^C x_{ij}^{k+1} - x_{ij}^k \right| \quad (10)$$

pour  $k \neq \{4, 8, 12, 16, 20, 24, 28, 32, 36, 40\}$

Supposant que l'enseignant  $i$  a le cas suivant :

	Créneaux (k)							
$\sum_{j=1}^C x_{ij}^k$	1	1	0	1	0	0	0	0

↑  
L'enseignant  $i$  a une séance  
pendant ce créneau

↑  
L'enseignant  $i$  n'a aucune  
séance pendant ce créneau

Puisque :

$$x_{ij}^k = \begin{cases} 1 & \text{si l'enseignant } i \text{ enseigne la classe } j \text{ pendant le créneau } k \\ 0 & \text{sinon} \end{cases}$$

Alors la formule :

$$\sum_{j=1}^C \left| \frac{1}{C} \sum_{j=1}^C x_{ij}^{k+1} - x_{ij}^k \right|$$

Représente la différence entre deux créneaux successifs, donc la somme sur tous les créneaux indique la somme des différences pour l'enseignant  $i$  pendant toute la semaine

$$\sum_{k=1}^{44} \sum_{j=1}^C \left| \frac{1}{C} \sum_{j=1}^C x_{ij}^{k+1} - x_{ij}^k \right|$$

Et pour éviter le problème de faire la différence entre les créneaux de 12h et 14h, aussi entre les créneaux de 18h et 8h du lendemain nous supposons que :

$k \neq \{4,8,12,16,20,24,28,32,36,40\}$ , D'où la formule de la fonction objective

### 3. Le Modèle Mathématique

$$\left\{ \begin{array}{ll}
 \text{Min } z = \sum_{i=1}^N \sum_{k=1}^{44} \sum_{j=1}^C \left| \frac{1}{C} \sum_{j=1}^C x_{ij}^{k+1} - x_{ij}^k \right| & k \neq \{4,8,12,16,20,24,28,32,36,40\} \\
 \\
 \sum_{i=1}^N \sum_{j=1}^C x_{ij}^k \leq NS & \forall k \in \{1,2, \dots, 44\} \\
 \\
 \sum_{i \in G_m} \sum_{j=1}^C x_{ij}^k \leq NS_m & \forall k \in \{1,2, \dots, 44\} \\
 & \forall m \text{ une matière a des salles spéciales} \\
 \\
 \sum_{k=1}^{44} \sum_{j=1}^C x_{ij}^k \leq NH_i & \forall i \in \{1,2, \dots, N\} \\
 \\
 \sum_{i=1}^N x_{ij}^k \leq 1 & \forall j = \{1,2, \dots, C\} \\
 & \forall k = \{1,2, \dots, 44\} \\
 \\
 \sum_{j=1}^C x_{ij}^k \leq 1 & \forall i = \{1,2, \dots, N\} \\
 & \forall k = \{1,2, \dots, NCR\} \\
 \\
 \sum_{k=l}^{l+7} x_{ij}^k \leq 2 & \forall i \in \{1,2, \dots, N\} \\
 & \forall j \in \{1,2, \dots, C\} \\
 & \forall l \in \{1,9,17,25,33\} \\
 \\
 \sum_{k=41}^{44} x_{ij}^k \leq 2 & \forall i \in \{1,2, \dots, N\} \\
 & \forall j \in \{1,2, \dots, C\} \\
 \\
 \sum_{i \in G_m} y_{ij} \leq 1 & \forall m \text{ une matière} \\
 \\
 \sum_{k=1}^{44} x_{ij}^k = NH_{mj} & \forall i = \{1,2, \dots, N\} \text{ tel que } m \text{ sa spécialité} \\
 & \forall j = \{1,2, \dots, C\}
 \end{array} \right.$$

### Conclusion

L'administration au cours de la construction d'un tableau de service prend en considération du fait que la plupart des enseignants ne soient pas permanents, d'où elle donne une importance à la disponibilité des enseignants comme on a précédemment indiqué, aussi nous avons présenté le modèle mathématique du problème en expliquant ses différentes composantes (les données, les variables, les contraintes et la fonction objective).

## *Chapitre 4 : Application au problème d'un lycée privé*

---

### *Introduction*

Nombreux sont les logiciels qui ont été conçus pour gérer les services d'enseignement tels que massar, FET.. Malgré leur diversité, ces logiciels se basent sur les mêmes principes que ce soit au niveau des données, ou au niveau des traitements.

Tableau de service est l'un des exemples les plus fréquents des problèmes d'optimisation dans un établissement scolaire. Il peut se manifester sous différentes formes selon la spécificité de l'établissement.

Dans ce chapitre, nous découvrons le problème de confection du tableau de service d'un lycée privé, et nous proposons un programme en *langage C*, basé sur le principe des algorithmes génétiques pour résoudre ce problème

## Tableaux des abréviations

Dans ce chapitre nous utilisons plusieurs abréviations, essayons tout d'abord de les définir :

<i>Niveaux et branches</i>	
<b>N<sub>1</sub></b> : Tronc commun science	
<b>N<sub>2</sub></b> : Tronc commun lettre	
<b>N<sub>3</sub></b> : Première année du baccalauréat sciences expérimentales	
<b>N<sub>4</sub></b> : Première année du baccalauréat sciences mathématiques	
<b>N<sub>5</sub></b> : Première année du baccalauréat sciences économiques	
<b>N<sub>6</sub></b> : Première année du baccalauréat lettres et sciences humaines	
<b>N<sub>7</sub></b> : Deuxième année du baccalauréat sciences physiques	
<b>N<sub>8</sub></b> : Deuxième année du baccalauréat sciences de la vie et de la terre	
<b>N<sub>9</sub></b> : Deuxième année du baccalauréat sciences économiques	
<b>N<sub>10</sub></b> : Deuxième année du baccalauréat sciences mathématiques	
<b>N<sub>11</sub></b> : Deuxième année du baccalauréat lettres	
<b>N<sub>12</sub></b> : Deuxième année du baccalauréat sciences humaines	
<i>Matières</i>	
<b>Math</b> : Mathématique	<b>Eco</b> : Economie
<b>Phy</b> : Physique	<b>L.Ar</b> : La langue Arabe
<b>Tra</b> : La traduction	<b>Info</b> : L'informatique
<b>L.F</b> : La langue Française	<b>Phil</b> : La philosophie
<b>H.G</b> : L'histoire et géographie	<b>E.Is</b> : L'éducation islamique
<b>E.phy</b> : L'éducation physique	<b>L.An</b> : La langue Anglaise
<b>SVT</b> : Science de la vie et de la terre	

Tableau 6 : Tableaux des abréviations

Ainsi nous intéressons au problème de confection de tableau de service d'un lycée privé qui a les données suivantes

## 1. Les données du lycée

Ce lycée comprend 12 Branches et niveaux et chaque branche a un nombre précis de classes, ainsi a un nombre précis d'enseignants classés selon leurs matières de spécialité, les tableaux suivants définissent les branches, et le nombre de classes pour chacun d'elles (tableau 7), aussi le nombre d'enseignants disponibles selon leurs matières de spécialité (tableau 8).

<i>branches</i>	$N_1$	$N_2$	$N_3$	$N_4$	$N_5$	$N_6$	$N_7$	$N_8$	$N_9$	$N_{10}$	$N_{11}$	$N_{12}$	Totale
<i>Nombre de classes</i>	4	2	4	1	1	3	4	2	2	1	5	1	30

Tableau 7 : Nombre de classes pour chaque branche

<i>matières</i>	<i>Math</i>	<i>Phy</i>	<i>SVT</i>	<i>Eco</i>	<i>L.Ar</i>	<i>L.F</i>	<i>L.An</i>	<i>E.Is</i>	<i>H.G</i>	<i>Phil</i>	<i>Tra</i>	<i>Info</i>	<i>E.Phy</i>
<i>Nombre d'enseignants</i>	8	4	4	2	6	8	6	4	4	4	2	1	4

Tableau 8: le nombre d'enseignants disponibles

Ainsi un ensemble de créneaux horaires étalés sur une semaine de six jours, du Lundi au samedi avec un nombre de huit périodes (du lundi au vendredi) et un nombre de quatre périodes le samedi, La durée d'une période est une heure, donc il y a 44 créneaux hebdomadairement.

À la suite de ce chapitre nous associons à :

- Chaque classe un nombre allant de 1 à 30
- Chaque enseignant un nombre allant de 1 à 57
- Chaque créneau un nombre allant de 1 à 44
- Chaque matière un nombre allant de 1 à 13

Et nous allons regrouper les enseignants selon leurs matières de spécialité :

<i>Matières</i>	<i>Groupes des enseignants</i>
Mathématique	$G_1 = \{1, 2, \dots, 8\}$
Physique	$G_2 = \{9, \dots, 12\}$
Science de la vie et de la terre	$G_3 = \{13, \dots, 16\}$
Economie	$G_4 = \{17, 18\}$



La langue Arabe	$G_5 = \{19, \dots, 24\}$
La langue Française	$G_6 = \{25, \dots, 32\}$
La langue Anglaise	$G_7 = \{33, \dots, 38\}$
L'éducation islamique	$G_8 = \{39, \dots, 42\}$
L'histoire et géographie	$G_9 = \{43, \dots, 46\}$
La philosophie	$G_{10} = \{47, \dots, 50\}$
La traduction	$G_{11} = \{51, 52\}$
L'informatique	$G_{12} = \{53\}$
L'éducation physique	$G_{13} = \{54, \dots, 57\}$

Tableau 9 : les groupes des enseignants

Chaque enseignant du lycée ne doit pas dépasser le nombre d'heures du travail fixé au début avec l'administration par semaine, noté  $NH_i$  tel que l'indice  $i$  représente le nombre associé à l'enseignant. Nous pouvons résumer les  $NH_i$  de tous les enseignants dans ce tableau :

Enseignants	$NH_i$	Enseignants	$NH_i$	Enseignants	$NH_i$
L'enseignant 1	15	L'enseignant 20	17	L'enseignant 39	20
L'enseignant 2	16	L'enseignant 21	16	L'enseignant 40	14
L'enseignant 3	10	L'enseignant 22	16	L'enseignant 41	15
L'enseignant 4	19	L'enseignant 23	18	L'enseignant 42	18
L'enseignant 5	18	L'enseignant 24	18	L'enseignant 43	20
L'enseignant 6	20	L'enseignant 25	16	L'enseignant 44	14
L'enseignant 7	8	L'enseignant 26	18	L'enseignant 45	18
L'enseignant 8	20	L'enseignant 27	18	L'enseignant 46	17
L'enseignant 9	21	L'enseignant 28	21	L'enseignant 47	20
L'enseignant 10	18	L'enseignant 29	20	L'enseignant 48	14
L'enseignant 11	14	L'enseignant 30	18	L'enseignant 49	15
L'enseignant 12	10	L'enseignant 31	20	L'enseignant 50	18
L'enseignant 13	17	L'enseignant 32	21	L'enseignant 51	20
L'enseignant 14	20	L'enseignant 33	18	L'enseignant 52	14
L'enseignant 15	14	L'enseignant 34	14	L'enseignant 53	18
L'enseignant 16	18	L'enseignant 35	13	L'enseignant 54	18
L'enseignant 17	20	L'enseignant 36	16	L'enseignant 55	20
L'enseignant 18	14	L'enseignant 37	16	L'enseignant 56	14
L'enseignant 19	15	L'enseignant 38	16	L'enseignant 57	14

Tableau 10 : nombres d'heures du travail fixés par chaque enseignant

Et d'après le calendrier 43 qui organise l'étude et la distribution des matières et des séances pour chaque niveau et branche au Maroc, nous obtenons tableau suivant :

<i>Niveau</i>	<i>Math</i>	<i>Phy</i>	<i>SVT</i>	<i>Eco</i>	<i>L.Ar</i>	<i>L.F</i>	<i>L.An</i>	<i>E.Is</i>	<i>H.G</i>	<i>Phil</i>	<i>Tra</i>	<i>Info</i>	<i>E.Phy</i>	<i>Total</i>
<i>N<sub>1</sub></i>	5	4	3	0	2	4	3	2	2	2	0	2	2	31
<i>N<sub>2</sub></i>	2	0	1	0	5	4	4	2	4	2	0	2	2	28
<i>N<sub>3</sub></i>	5	4	4	0	2	4	3	2	2	2	2	0	2	32
<i>N<sub>4</sub></i>	7	5	2	0	2	4	3	2	2	2	2	0	2	33
<i>N<sub>5</sub></i>	4	0	0	11	2	2	3	2	2	2	0	1	2	31
<i>N<sub>6</sub></i>	2	0	1	0	5	5	4	2	4	2	0	0	2	27
<i>N<sub>7</sub></i>	5	6	4	0	2	4	3	1	0	2	2	0	2	31
<i>N<sub>8</sub></i>	5	4	6	0	2	4	3	1	0	2	2	0	2	31
<i>N<sub>9</sub></i>	4	0	0	11	2	2	3	1	2	2	0	2	2	31
<i>N<sub>10</sub></i>	7	6	2	0	2	4	3	1	0	2	2	0	2	31
<i>N<sub>11</sub></i>	2	0	0	0	5	5	5	2	4	3	0	0	2	28
<i>N<sub>12</sub></i>	2	0	0	0	4	4	4	3	5	4	0	0	1	28

Tableau 11 : la distribution des matières et des séances pour chaque niveau

Certaines matières peuvent se dérouler dans des salles générales puisqu'elles n'exigent pas des matériels spéciaux, mais d'autres les exigent, donc les salles sont classées en deux types (Salles générales et Salles spéciales), parmi les matières qui ont des salles spéciales, nous trouvons la matière du physique, science de la vie et de la terre (SVT), informatique et l'éducation physique, aussi le nombre de salles disponible au lycée est comme suite :

- 30 salles générales
- 4 salles de physique
- 4 salles de SVT
- une salle d'informatique
- la cour du sport support au maximum 3 séances en parallèle

## **2. Vérification de faisabilité du problème**

Maintenant nous avons toutes les informations nécessaires pour fournir un tableau de service des enseignants de ce lycée, mais avant de commencer la confection du tableau de service il faut vérifier la faisabilité à partir des calculs concernant :

- Le nombre d'heures pour chaque enseignant pendant la semaine.
- le nombre de salles nécessaires pour que toutes les classes prennent leurs cours et utilisent le même type de salles (générales ou spéciales) pendant un créneau précis.

➤ *La vérification de nombre d'heures de travail pour chaque enseignant*

Niveaux	Nombre des classes	Math	Phy	SVT	Eco	L.Ar	L.F	L.An	E.Is	H.G	Phil	Tra	Info	E.Phy
$N_1$	5	25	20	15	0	10	20	15	10	10	10	0	10	10
$N_2$	5	10	0	5	0	25	20	20	10	20	10	0	10	10
$N_3$	5	25	20	20	0	10	20	15	10	10	10	10	0	10
$N_4$	1	7	5	2	0	2	4	3	2	2	2	2	0	2
$N_5$	2	8	0	0	22	4	4	6	4	4	4	0	2	4
$N_6$	5	10	0	5	0	25	25	20	10	20	10	0	0	10
$N_7$	6	30	36	24	0	12	24	18	6	0	12	12	0	12
$N_8$	3	15	12	18	0	6	12	9	3	0	6	6	0	6
$N_9$	2	8	0	0	22	4	4	6	2	4	4	0	4	4
$N_{10}$	1	7	6	2	0	2	4	3	1	0	2	2	0	2
$N_{11}$	13	26	0	0	0	65	65	65	26	52	39	0	0	26
$N_{12}$	1	2	0	0	0	4	4	4	3	25	4	0	0	2
<b>Totale d'heures</b>		118	75	65	33	92	122	106	52	69	67	24	17	59
<b>Nombre d'enseignants disponibles</b>		8	4	4	2	6	8	6	4	4	4	2	1	4
<b>Somme des <math>NH_i</math></b>		123	80	69	34	100	128	115	54	72	69	30	18	60

Tableau 12 ; vérification de suffisance des enseignants

Nous faisons la somme d'heures de chaque matière pendant une semaine (**Totale d'heures**), puis on compare le résultat avec (**Somme des  $NH_i$** ), tel que les  $i$  sont les enseignants de même spécialité, si le **Totale d'heures** est inférieur ou égal à **Somme des  $NH_i$**  alors, le nombre d'enseignants disponibles est suffisant, sinon le nombre d'enseignants disponibles pour cette matière n'est pas suffisant donc le problème n'est pas faisable, et d'après le (tableau 12) on a le nombre d'enseignants disponibles est suffisant.

➤ *Vérification de la suffisance de nombre de salles*

Comme il est indiqué au chapitre 3 , la vérification de nombre de salles consiste à comparer le nombre de séances qui peuvent se dérouler pendant un créneau k avec le nombre de salles disponibles, il est préférable que le résultat soit inférieur ou égal au nombre de salles disponibles, car sinon on peut trouver des problèmes au niveau de construction de notre tableau de service, le tableau suivant résume les calculs nécessaires :

<i>Niveaux</i>	$N_1$	$N_2$	$N_3$	$N_4$	$N_5$	$N_6$	$N_7$	$N_8$	$N_9$	$N_{10}$	$N_{11}$	$N_{12}$
<i>Nombre de classes</i>	5	5	5	1	2	5	6	3	2	1	13	1
<i>Total d'heures pour une seule classe</i>	20	23	22	24	28	24	19	24	27	21	26	26
<i>Total d'heures pour toutes les classes</i>	100	115	110	24	56	120	114	72	54	21	338	26
<i>total</i>									1150			
<i>Nombre de salles générales nécessaires</i>									$1150/44 = 26,14 \approx 27$			

Tableau 8 : Nombre de salles générales nécessaires

**Total** : signifie le nombre d'heures d'étude pour chaque niveau (pour les matières qui se déroule dans les salles générales) multiplié par le nombre de classes

- On a :

$$1150 / 44 = 26,14 \leq 30$$

Donc le nombre de salles générales est suffisant.

- on a la somme des heures de physiques pour tous les Niveaux égale à 141:

$$141 / 44 = 3,20 \leq 4$$

Donc le nombre de salles de physique est suffisant.

- Même chose pour les salles de SVT on a :

$$131 / 44 = 2,97 \leq 4$$

Donc le nombre de salles est suffisant.

- Pour les salles d'informatique on a :

$$26 / 44 \leq 1$$

D'où une salle d'informatique est suffisante.

- Pour la cour du sport :

$$98 / 44 = 2,227 \leq 3$$

Donc, le problème est réalisable.

### 3. L'adaptation des algorithmes génétiques

La résolution de ce problème se fait en appliquant les algorithmes génétiques, présentons tout d'abord les différentes étapes :

#### 3.1. Codage

Chaque chromosome représente l'emploi du temps d'un enseignant, est constitué de 44 gènes et chaque gène contient un entier de 0 à 30 tel que 0 signifie que l'enseignant n'a aucune séance et chaque nombre de 1 à 30 représente une classe, donc un individu est représenté par 57 chromosomes. et chaque population est constituée de 10 individus

<i>l'ensemble des créneaux</i>					
<b>Enseignant 1</b>	5	2	0	...	0 0
<b>Enseignant 2</b>	0	3	4	...	4 3
				⋮	
<b>Enseignant 56</b>	4	4	6	...	1 0
<b>Enseignant 57</b>	0	0	30	...	30 18

Tableau 13 : un exemple de codage d'un individu

➤ *Fonction d'adaptation (fitness)*

Comme fonction d'adaptation nous associons à chaque individu une valeur égale la valeur à minimisée dans la fonction objective.

$$\forall \text{ indiv} \in \{1, 2, \dots, 10\} \quad \text{fitness}(\text{indv}) = \sum_{i=1}^N \sum_{k=1}^{44} \sum_{j=1}^C \left| \frac{1}{C} \sum_{j=1}^C x_{ij}^{k+1} - x_{ij}^k \right|$$

$$k \neq \{4, 8, 12, 16, 20, 24, 28, 32, 36, 40\},$$

➤ *Sélection*

Pour la sélection nous utilisons la sélection par tournoi basée sur le principe suivant tirer aléatoirement (  $k = 6$  ) individus de la population, sans tenir compte de la valeur de leur fonction d'adaptation, et de choisir le meilleur individu parmi les  $k$  individus.

➤ *Mutation*

Pour la mutation choisissons un créneau  $k$  aléatoirement puis nous faisons une permutation entre deux gènes de deux chromosomes représentent deux enseignants de même spécialité, et nous ajoutons les modifications nécessaires pour que la solution soit réalisable

Créneau sélectionné

<i>Enseignant 1</i>	5	2	0	...	0	0
<i>Enseignant 2</i>	0	3	4	...	4	3
				⋮		
				⋮		
<i>Enseignant 56</i>	4	4	6	...	1	0
<i>Enseignant 57</i>	0	0	30	...	30	18

Tableau 14 : exemple de mutation dans un individu

## 4. Programmation en langage C

### 4.1. Génération d'un individu

Pour générer un individu nous suivons ces étapes :

#### ➤ Initialisation des données

Les données sont programmées sous forme des matrices et des tableaux comme suite :

- une matrice (*indv*) qui représente l'individu en fixant les créneaux où chaque enseignant est non disponible par zéro, et donner un nombre par exemple (-1), aux créneaux où on peut affecter une classe.

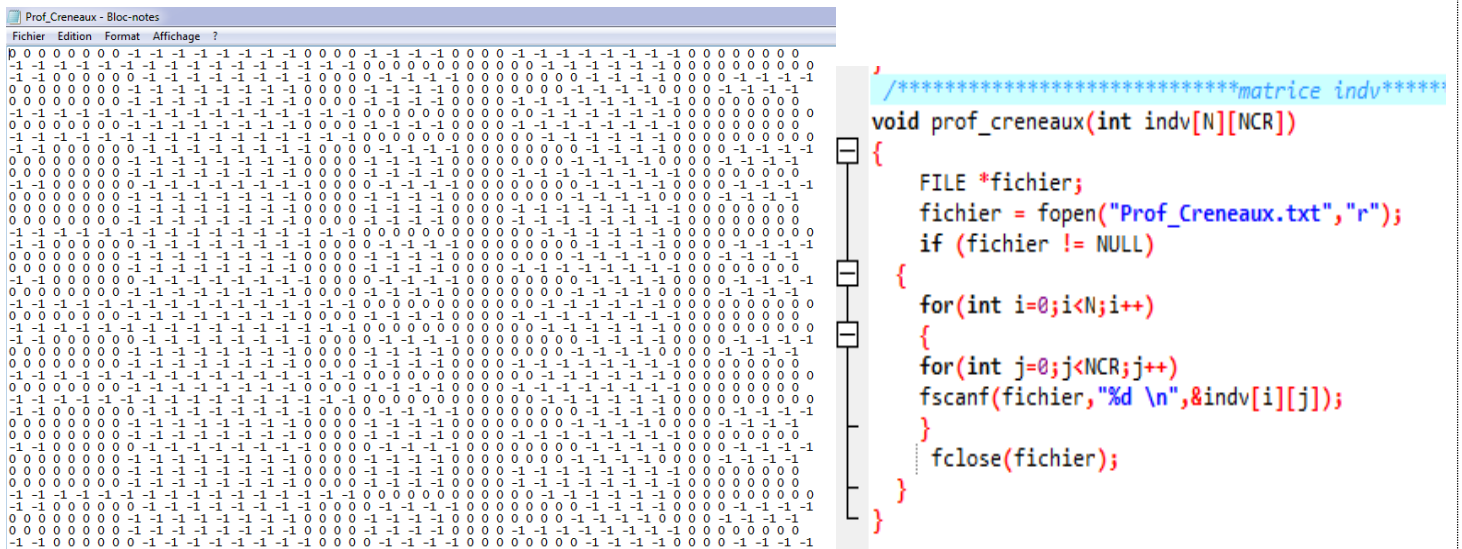


Figure 9 : initialisation de la matrice (indv)

- une matrice *P\_C* de 57 lignes et 30 colonnes, tel que les lignes représentent les enseignants et les colonnes représentent les classes, cette matrice représente la variable  $y_{ij}$  tel que chaque élément égal à 1 signifie que la classe *j* est affectée au enseignant *i*. au début nous initialisons cette matrice par des zéros.

```
 /*****prof_classe*****/
void init_matrice_P_C(int P_C[N][C])
{
    for(int i=0; i<N; i++)
    {
        for(int j=0; j<C; j++)
        {
            P_C[i][j]=0;
        }
    }
}
```

Figure 10 : initialisation de la matrice P\_C

- [illegible]

- une matrice  $M_C$  de 30 lignes et 13 colonnes, telles que les lignes représentent les classes et les colonnes représentent les matières, cette matrice représente le tableau 11

```

/****Matiere_classe_heures*****/
void Matiere_classe(int M_C[C][M])
{
    FILE *fichier;
    fichier = fopen("Matiere_classe.txt", "r");
    if (fichier != NULL)
    {
        for(int i=0; i<C; i++)
        {
            for(int j=0; j<M; j++)
            {
                fscanf(fichier, "%d \n", &M_C[i][j]);
            }
        }
        fclose(fichier);
    }
}

```

Figure 12 : initialisation de la matrice M\_C



- un tableau de taille 13 tel que chaque case contient le nombre des enseignants d'une matière

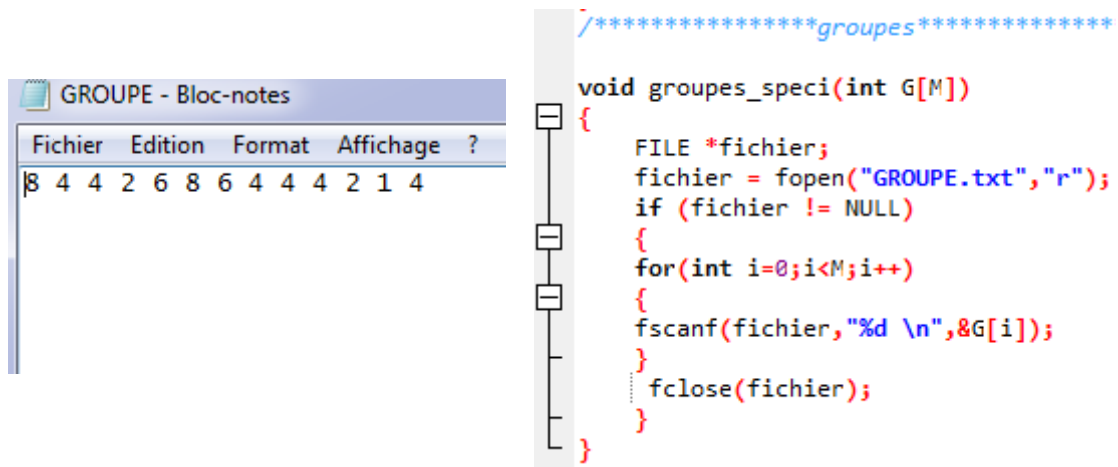


Figure 13 : initialisation du tableau des groupes

- un tableau de taille 57, chaque case contient le nombre d'heures de travail exigées par chaque enseignant.

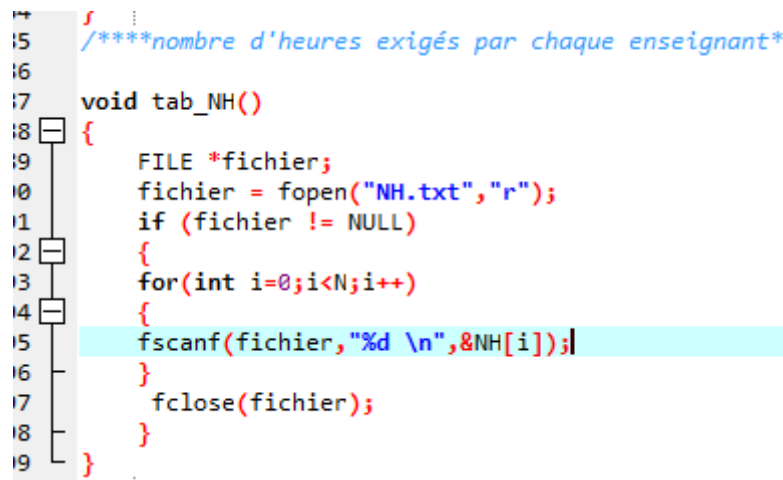


Figure 14 : initialisation du tableau de nombre d'heures exigées

### ➤ Génération d'un individu

Pour générer un individu nous utilisons les matrices et les tableaux initialisés au début et nous suivons l'algorithme suivant :

- Fixer les créneaux où les enseignants ne sont pas disponibles par des zéros

- Choisir le créneau où il ya le nombre maximum des enseignants disponible, en utilisons la fonction *indice\_max\_col* qui retourne l'indice de la colonne (de la matrice *indv*) qui a le nombre maximum des enseignants disponible, pour affecter le plus grand nombre possible des classes aux enseignants

```

393  ,/*****Max d'une colonne*****/
394  int indice_max_col(int indv[N][NCR])
395  {
396      int s,T[NCR];
397      int ind;
398      for(int k=0;k<NCR;k++)
399      {
400          s=0;
401          for(int i=0;i<N;i++)
402          {
403              if(indv[i][k]==-1)
404              {
405                  s++;
406              }
407          }
408          T[k]=s;
409      }
410      ind=indice_max_tab_cre(T);
411      return ind;
412  }
413  }

```

Figure 15 : fonction qui retourne l'indice de la colonne qui représente le nombre maximal des enseignants disponibles

- Remplir les cases où les enseignants sont disponibles en tenant compte les contraintes (1), (2) et (3).

```

489  void gener_indiv(int indv[N][NCR])
490  {
491      srand(time(NULL)); // initialisation de rand
492      int i,k,ci,specialite,x,indice,m,indice_prof,g,sclasse,i_depart;
493      int TB[C],G[M],DF[N],T[N],TC[NCR];
494      int s,depart,val_c,matiere,scompleter,kcompleter;
495      int ssalle,valeur_matiere,k_depart;
496      tab_NH(NH);
497      groupes_speci(G);
498      init_matrice_P_C(P_C);
499      Matiere_prof(M_P);
500      prof_creneaux(indv);
501      Matiere_classe(M_C);
502      init_creneaux(TC);
503      srand(time(NULL));
504
505      k_depart=indice_max_col(indv);
506      TC[k_depart]=1;//signifie qu'on a deja rempli la colonne d'indice k_depart
507      init_tab(TB,C);
508      ssalle=0;// conteur de salles
509      sclasse=0;// conteur de classe
510      i=0;
511      printf(" \n en train d'affecter les séances du creneau %d \n",k_depart);

```

```

512 do{
513     m=chercher_P_M(i);
514     if(indv[i][k_depart] ==(-1) && sclasse<C-1)
515     {
516         do
517         {
518             do{
519
520                 srand(time(NULL));
521                 x=rand()%(C-1)+1;
522             }while(M_C[x][m]==0);
523             }while(TB[x]!=0);
524             indv[i][k_depart]=x;
525             TB[x]++;
526             P_C[i][x]=1; /******l'affectation de la classe x à l'enseignant i*****/
527             valeur_matiere=chercher_P_M(i); //pour completer le nombre d'heures de la matiere pour la classe x
528             sclasse++;
529
530         }
531     }
532     else
533     {
534         if(sclasse==C-1)
535         break;
536     }
537     i++;
538     indice_prof=i;
539     ssalle++;
540     }while(i<N && ssalle<NS); // contrainte de disponibilité des salles
541     if(ssalle==NS)
542     {
543         for(g=indice_prof;g<N;g++)
544         {
545             indv[g][k_depart]=0;
546         }
547     }
548     else
549     {
550         if(sclasse==C-1)
551         {
552             for(g=indice_prof;g<N;g++)
553             {
554                 indv[g][k_depart]=0;
555             }
556         }
557     }
558 }
559
560
561
562
563
564
565
566
567
568
569
570

```

Figure 16 : première étape pour générer un individu

- Pour chaque gène remplie continue l'affectation de la calasse à des éléments de créneaux différents jusqu'à atteint le nombre des séances nécessaires de la matière enseigné par l'enseignant représenté par la ligne de la matrice *indv*.

```

valeur_matiere=chercher_P_M(i); //pour completer le nombre d'heures de la matiere pour la classe x
sclasse++;
printf(" \n scsse= %d\n",sclasse);

if(M_C[x][valeur_matiere]!=0)
{
    s=1;
    do{
        srand(time(NULL));
        k=rand()%(NCR);
        if(indv[i][k]==-1)
        {
            indv[i][k]=x;
            s++;
        }
    }while(s<M_C[x][valeur_matiere]);
}

```

Figure 17 : deuxième étape pour générer un individu

- A chaque affectation nous remplissons la matrice P\_C.
- Affecter les classes déjà affectées (pendant le créneau k\_départ) aux enseignants de spécialités différentes à celle de début en vérifiant les contraintes (1),(2),(3),(4),(5),(6),(7).

```

554 printf(" \n remplissage d'autres creneaux \n");
555 for(i_depart=0;i_depart<N;i_depart++)
556 {
557     if(indv[i_depart][k_depart]!=0)
558     {
559         matiere=chercher_P_M(i_depart);
560         for(int m=0;m<M;m++)
561         {
562             do{
563                 k=rand()%(NCR);
564             }while(TC[k]!=0 && indv[i][k]!=(-1));
565             if(m!=matiere && somme_classe_specia(P_C,m,indv[i][k_depart])<1)
566             {
567                 if(M_C[indv[i][k_depart]][m]!=0 && somme_1_sp(indv,m,k)!=0)
568                 {
569                     do{
570                         i=rand()%(G[m])+dep_g[m];
571                     }while(indv[i][k]==-1);
572                     P_C[i][indv[i_depart][k_depart]]=1;
573                     indv[i][k]=indv[i_depart][k_depart];
574                     ssss=1;
575                     do{
576                         srand(time(NULL));
577                         kkk=rand()%(NCR);
578                         if(indv[i][kkk]==-1)
579                         {
580                             indv[i][kkk]=indv[i][k];
581                             x=indv[i][kkk];
582                             ssss++;
583                         }
584                     }while(ssss<M_C[x][valeur_matiere] && creneau_classe(indv[i][kkk],kkk)>1);
585                     valeur_matiere=chercher_P_M(i);

```

Après avoir compiler ce programme nous avons obtenu l'individu suivant :

### ➤ *Un extrait d'individu obtenu*

```

0 0 0 0 0 0 0 0 28 21 22 21 13 22 13 22 0 0 0 0 22 13 21 28 0 0 0 0 22 22 22 0 21 14 0 21 0 0 0 0 0 0 0
5 0 3 0 11 11 3 0 3 0 0 11 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
27 0 0 0 0 0 0 0 26 26 24 0 26 0 0 0 0 0 0 24 0 0 24 0 0 0 0 0 0 0 0 0 0 1 24 0 0 0 0 0 0 0 24 12 0
0 0 0 0 0 0 0 0 29 29 7 4 2 7 0 2 0 0 0 0 0 7 7 0 0 0 0 0 0 0 0 0 2 2 7 4 0 0 0 0 2 7 4 2
0 0 0 0 0 0 0 0 6 0 16 0 0 0 16 6 0 0 0 0 16 0 6 16 0 0 0 0 6 0 16 0 6 0 0 0 0 0 0 0 0 0 0 0
8 23 0 15 23 8 15 23 23 0 15 23 15 0 23 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 15 15 8 23 8 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 20 19 9 0 20 9 20 20 0 0 0 0 20 0 20 9 0 0 0 0 0 9 20 9 9 20 0 20 0 0 0 0 0 0 0 0
0 10 10 0 18 10 10 10 10 10 10 0 10 0 0 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 10 0 10 0 0 0 0 0 0 0 0
9 15 0 0 0 0 0 0 7 15 7 15 0 15 0 7 0 0 0 0 0 7 7 0 0 0 0 0 0 0 0 7 0 15 15 0 0 0 0 0 7 15 0
0 0 0 0 0 0 0 0 24 0 0 0 26 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 26 0
0 0 0 0 0 0 0 0 27 5 17 10 17 5 10 17 0 0 0 0 0 10 17 0 0 0 0 0 0 17 0 10 17 17 10 10 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 1 16 0 1 16 0 0 16 0 0 0 0 0 16 0 0 0 0 0 0 16 16 0 16 0 0 0 0 29 0 1 29
0 0 0 0 0 0 0 0 18 0 9 18 0 9 18 0 0 0 0 18 0 0 9 0 0 0 0 0 0 0 9 0 0 0 0 0 0 0 0 9 0
0 0 0 0 0 0 0 0 2 23 16 16 23 0 16 0 0 0 0 16 0 23 0 0 0 0 19 16 2 0 16 16 2 16 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 12 29 0 0 6 0 0 6 0 0 0 0 6 0 0 0 0 0 29 0 6 0 6 12 0 0 0 0 0 0 0 0 0
3 0 3 15 0 0 15 28 15 15 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 28 3 0 0 3 0 0 0 0 0 0 0 0

```

Figure 18 : un extrait d'individu obtenu

➤ *explication*

après le décodage d'individu obtenu nous pouvons poser un tableau de service contient les emplois du temps de tous les enseignants, par exemple les emplois du temps d'enseignants 1 et 4

Enseignant 1								
	8h – 9h	9h-10h	10h-11h	11h-12h	14h-15h	15h-16h	16h-17h	17h-18h
<b>Lundi</b>								
<b>Mardi</b>	2 <sup>ème</sup> BAC, lettres (c12)	2 <sup>ème</sup> année BAC, SVT (c3)			1 <sup>ème</sup> année BAC, S.ECO (c3)		2 <sup>ème</sup> année BAC, S.ECO (c1)	
<b>Mercredi</b>						1 <sup>ème</sup> BAC,S.E CO (c3)	2 <sup>ème</sup> BAC, SVT (c3)	2 <sup>ème</sup> BAC, lettres (c12)
<b>Jeudi</b>					2 <sup>ème</sup> BAC,S.ECO (c1)			
<b>Vendredi</b>	2 <sup>ème</sup> BAC, SVT (c3)	1 <sup>ème</sup> BAC,S.ECO (c3)		2 <sup>ème</sup> année BAC,SVT (c3)				
<b>Samedi</b>								

Enseignant 4								
	8h – 9h	9h-10h	10h-11h	11h-12h	14h-15h	15h-16h	16h-17h	17h-18h
<b>Lundi</b>								
<b>Mardi</b>	Tronc. C science (c2)	1 <sup>ème</sup> année BAC, S.Ex (c3)	Tronc. C science (c2)	1 <sup>ème</sup> année BAC, S.Ex (c3)	1 <sup>ème</sup> année BAC, S.Ex (c1)		Tronc.C science (c4)	
<b>Mercredi</b>					Tronc.C science (c2)		1 <sup>ème</sup> année BAC, S.Ex (c1)	
<b>Jeudi</b>								
<b>Vendredi</b>	Tronc.C science (c2)		1 <sup>ème</sup> année BAC, S.Ex (c1)	Tronc.C science(c4)	1 <sup>ème</sup> année BAC, S.Ex (c3)			
<b>Samedi</b>	1 <sup>ème</sup> année BAC.S.Ex (c3)		Tronc.C science (c4)					

## 4.2. fonction d'adaptation

Pour programmer la fonction d'adaptation nous avons besoin d'une fonction qui retourne la valeur absolue d'un nombre donné :

```
/******val abs*****/  
float val_absolue(float a)  
{  
    if(a<0)  
        return (-1)*a;  
    else  
        return a;  
}
```

D'où le programme de la fonction d'adaptation est comme suite :

```
/******evaluation*****/  
float evalution()  
{  
    int si=0,sk,sj,s,abs;  
    for(int i=0;i<N;i++)  
    {  
        sk=0;  
        k=0;  
        if(k!=4 && k!=8 && k!=12 && k!=16 && k!=20 && k!=24 && k!=28 && k!=32 && k!=36 && k!=40)  
        {  
            sj=0;  
            for(int j=0;j<C;j++)  
            {  
                s=0;  
                for(int k=0;k<C;k++)  
                {  
                    if(indv[i][k+1]!=0)  
                        s++;  
                }  
                if(indv[i][k]!=0)  
                    abs=val_absolue(1/C*s-1);  
                else  
                    abs=val_absolue(1/C*s);  
                sj=sj+abs;  
            }  
            k++;  
            sk=sk+sj;  
        }  
        si=si+sk;  
    }  
}
```

Figure 19 : programme de la fonction d'adaptation (*fitness*)

## Conclusion

Dans ce chapitre nous avons appliqué les différentes étapes des algorithmes génétiques à un problème d'un lycée privé, puis nous avons présenté le programme des deux premières étapes des algorithmes génétiques en utilisant une programmation en langage C, ensuite nous avons décodé l'individu obtenu par ce programme en donnant deux exemples d'emplois du temps de deux enseignants.

## *Conclusion générale et Perspectives*

Dans ce rapport, nous avons essayé de présenter quelques généralités sur la confection des tableaux de service des enseignants (enseignement secondaire), en particulier la séquence historique des études réalisées sur ce sujet, et les termes connus au domaine pédagogique.

Puis nous avons présenté le principe des algorithmes génétiques et plus précisément les différentes étapes suivraient au cours de construction d'une solution réalisable ou meilleure.

Ensuite, nous avons donné une formulation du problème de confection d'un tableau de service d'un lycée privé, en présentant un modèle mathématique convenable.

Finalement, nous avons appliqué les deux premières étapes des algorithmes génétiques à un problème d'un lycée privé en utilisant une programmation en langage C.

Comme perspectives nous allons continuer la programmation de croisement et mutation pour obtenir la meilleure solution, et pourquoi pas faire un interface capable de donner le meilleur tableau de service pour n'importe quel lycée privé.

## ***Bibliographie et web-graphie***

- [1] Burke E., Kingston J., Jackson K., Weare R., “Automated university Timetabling : the state of the art”, the Computer Journal 40 (9) 565-571, (1997)
- [2] Coello Coello, Carlos A., Van Veldhuizen D.A. et Lamont G.B. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York p 576, (2002)
- [3] Deb K.. *Introduction to selection*. Dans : “Evolutionary computation 1: advanced algorithms and operators”. Édité par : Bäck T., Fogel D.B., et Michalewicz Z.; Institute of Physics Publishing, Bristol and Philadelphia, p 331, (2000)
- [4] De Werra D., “An Introduction to Timetabling”, European Journal of Operational research p 19, 151-162, (1985)
- [5] Gagné C. et Parizeau M. *Open BEAGLE, a versatile EC framework (Open BEAGLE, un environnement générique pour effectuer des AE)*, (2004)
- [6] Goldberg D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Longman, p 412, (1989)
- [7] Horn J., Nafpliotis N. et D.E. Goldberg “A Niche Pareto Genetic Algorithm for Multiobjective Optimization”. Dans : Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Vol. 1, pp. 82-87, Piscataway, New Jersey. IEEE Service Center, (1994)
- [8] Meisels, A. Schaerf, « Modelling and solving employee timetabling problems », Applied Intelligence, accepted, (2001)
- [9] Sandhu K., “Automating class schedule generation in the context of university timetabling information system”, School of Management, Nathan Campus, Griffith University, 21 september (2001)
- [10] Schaerf A., et Schaerf M. “Local search techniques for large high school timetabling”, in proceeding of the 1 st international conference on the practice and theory of automated timetabling, pp. 313-323, (1995)
- [11] Thierry Moyaux, Brahim Chaib-draa et Sophie D'amours, « Satisfaction distribuée de contraintes et son application à la génération d'un emploi du temps d'employées », 5ème Congrès international de Génie Industriel - ville de Québec 26 au 29 Octobre (2003)
- [12] <http://dspace.univ-tlemcen.dz/bitstream/112/6385/1/Gestion-Des-Emplois.pdf>
- [13] [http://www.gel.ulaval.ca/~beagle/index\\_f.html](http://www.gel.ulaval.ca/~beagle/index_f.html)
- [14] [http://www.gel.ulaval.ca/~beagle/index\\_f.html](http://www.gel.ulaval.ca/~beagle/index_f.html).