

2.1. Introduction

L'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation [03] et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation [2]. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données. [15]

Etant donnée l'importance de ces problèmes, nombreuses méthodes de résolution ont été développées en recherche opérationnelle (RO) et en intelligence artificielle (IA). Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) qui garantissent la complétude de la résolution et les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité, donc dans ce chapitre on parle plus détaillé sur les techniques et les méthodes d'optimisation utilisées pour résoudre notre problématique.

2.2. Notions de base en optimisation

Deux types de problèmes d'optimisation sont distingués: des problèmes de minimisation et des problèmes de maximisation. Un problème d'optimisation (de minimisation ou de maximisation) est défini par un ensemble de données et un ensemble de contraintes. Un ensemble de solutions S est associé au problème d'optimisation. Parmi les solutions S , un sous-ensemble $X \subseteq S$ représente des solutions réalisables respectant les contraintes C du problème, à chaque solution s est associée une valeur $f(s)$ qui représente sa qualité. La résolution du problème d'optimisation consiste à trouver une solution $s^* \in X$ qui minimise ou maximise la valeur $f(s)$.

Quelque soit le type du problème d'optimisation, ce dernier est défini par le 6-uplet $\langle D, C, S, X, f, mode \rangle$. Où D représente les données du problème, C les contraintes que doit satisfaire une solution afin d'être admissible, S l'ensemble des solutions possibles du

problème traité, X un sous-ensemble de S représentant les solutions réalisables (admissibles), f une fonction du coût (aussi appelée fonction « objectif » ou fonction fitness) qui associe à chaque solution s une valeur numérique $f(s)$ (nombre réel ou entier) représentant la qualité de s , $mode$ indique le type du problème, il permet de savoir est ce qu’on doit minimiser ou maximiser les valeurs des solutions de X . Dans ce qui suit nous présentons quelques définitions tirées de la littérature liées aux problèmes d’optimisation en général.

Une instance d’un problème de minimisation (maximisation) : est un couple (X, f) Où $X \subseteq S$ est un ensemble fini de solutions potentielles admissibles et f est une fonction du coût (fonction objectif) à minimiser (à maximiser), $f: X \rightarrow \mathfrak{R}$. L’objectif est de trouver $s^* \in X$ tel que $f(s^*) \leq f(s)$ (Au cas de minimisation) et $f(s^*) \geq f(s)$ (Au cas de maximisation) pour n’importe quelle solution $s \in X$. [3]

Un problème d’optimisation : est défini par l’ensemble S des solutions possibles d’un problème. $X \subseteq S$ est l’ensemble des solutions réalisables. $f: X \rightarrow \mathfrak{R}$, une fonction que l’on nomme fonction «objectif». La résolution du problème consiste à minimiser (maximiser) la valeur $f(s)$, où $s \in X$

Un problème multi-objectif : est décrit par la formule 2.1

$$Mode_x \vec{f}(s) \quad (2.1)$$

$$\text{Tel que} \left\{ \begin{array}{l} \vec{C}_i(s) \Delta 0, i = 1 \dots m \\ \vec{H}_j(s) = 0, j = 1 \dots p \\ s \in S \subset \mathfrak{R}^n \end{array} \right.$$

$$\text{Où} \quad \text{Mode} = \left\{ \begin{array}{l} \text{Min au cas de problème de minimisation} \\ \text{Max au cas de problème de maximisation} \end{array} \right.$$

$$\Delta \text{ est remplacé par } \left\{ \begin{array}{l} \leq \text{ au cas de problème de maximisation} \\ \leq \text{ au cas de problème de maximisation} \end{array} \right.$$

Dans le cas de problèmes d’optimisation multi-objectifs, la fonction «objectif» est représentée par un vecteur \vec{f} regroupant N fonctions «objectif». Aussi les contraintes sont représentées par des vecteurs regroupant les contraintes de chaque fonction «objectif».

Une solution d’un problème d’optimisation : est un ensemble de quantités souvent numériques pour lesquelles des valeurs sont à choisir. L’ensemble de ces valeurs est généralement regroupé dans un vecteur représentant une solution. Supposant un problème de taille n, le vecteur représentant la solution s est représenté par:

$$\vec{S} = [s_1, s_2, s_3, \dots, s_n] \quad (2.2)$$

Les différentes valeurs prises par les variables ($s_1, s_2, s_3, \dots, s_n$) constituent l’ensemble des solutions envisageables. Selon le type des variables ($s_1, s_2, s_3, \dots, s_n$), le type du problème d’optimisation peut être reconnu. Par conséquent, on distingue deux classes de problèmes d’optimisation: des problèmes continus et des problèmes discrets. Dans la première classe (i.e. la classe des problèmes continus), les variables composant une solution donnée sont de type réel. Tandis que, dans les problèmes de la deuxième classe (i.e. la classe des problèmes discrets), les variables composant une solution donnée peuvent être de type entier, naturel ou binaire. [22]

Une contrainte d’un problème : est une restriction imposée par la nature et les caractéristiques du problème sur les solutions proposées.

L’espace de recherche S d’un problème : est composé de l’ensemble de valeurs pouvant être prises par les variables ($s_1, s_2, s_3, \dots, s_n$) qui construisent la solution s.

Le voisinage $V(s)$ d’une solution s : est un sous ensemble de S, dont les membres sont des solutions proches (voisines) de la solution s. En effet, on dit qu’une solution «s'» est une voisine de s, si elle peut être obtenue en modifiant légèrement la solution s.

L’optimum local : est la meilleure solution appartenant à un voisinage $V(s)$ de la solution s. Une solution (appartenant à S) est un optimum local de la structure du voisinage $V(s)$ de la solution s si elle vérifie la condition suivante (2.3):

$$f(s') \leq f(s) \quad \forall s \in V(s) \quad (2.3)$$

Pour un problème de maximisation l’inégalité est inversée, c.-à-d. la condition (2.3) sera remplacée par la condition (2.4):

$$f(s') \geq f(s) \quad \forall s \in V(s) \quad (2.4)$$

Une solution optimale (L’optimum global): Une solution s^* est dite optimale (ou optimum global) si les deux contraintes suivantes sont vérifiées:

1. Elle est réalisable: tirée de l’ensemble de solutions possibles (l’espace de recherche S) et respecte toutes les contraintes du problème posé.

$$2. \left\{ \begin{array}{ll} f(s^*) = \max_{s \in X} \{f(s)\} & \text{En cas de problème de maximisation} \\ f(s^*) = \min_{s \in X} \{f(s)\} & \text{En cas de problème de minimisation} \end{array} \right.$$

Où X est l’ensemble de solutions réalisables.

2.3. Les étapes de la méthode générale de résolution des problèmes

Il y a de nombreuses écoles et les méthodes changent par la nature et le nombre des étapes et des outils utilisés. Nous proposons ici à titre d’information une trame standard de résolution de problèmes, sachant qu’il est toujours possible d’y ajouter des étapes ou des outils. L’outil de résolution de problèmes n’est pas à utiliser dans tous les cas. Et puis, comme tout outil, il y a ceux que nous préférons et ceux que nous n’aimons pas trop.

Un outil n’est jamais indispensable pour bien travailler. Il constitue une aide, ce n’est pas un passage obligé. En conséquent, il faut travailler avec les outils que l’on aime, qui nous sont sympathiques.

Comme nous n’avons pas beaucoup de temps à consacrer à l’amélioration de l’organisation, nous devons travailler efficacement. Afin d’éviter de disperser notre énergie, nous nous concentrerons sur l’un des problèmes les plus importants. Pour cela, nous procéderons à un inventaire de tout ce qui ne marche pas bien autour de nous, puis nous classerons ces problèmes par ordre décroissant d’importance.

Il arrive parfois que certains dysfonctionnements soient difficiles à exprimer. Ce sont des malaises latents comme par exemple : « Nous avons des difficultés à communiquer ». Dans ce cas, il est utile de définir le problème avec précision en réalisant une analyse préalable.

Lorsque le problème à résoudre est choisi et compris par tous, il faut constituer un groupe de travail composé des personnes dont les compétences sont en rapport avec l’affaire à traiter.

Ensuite, nous devons préciser les indicateurs qui nous permettront d’évaluer l’efficacité de notre action.

Nous pouvons alors enfin entrer dans la phase active, c’est-à-dire rechercher les causes potentielles du problème et bien comprendre les relations entre elles. Un problème bien posé et bien compris est à moitié résolu.

Nous choisirons alors parmi toutes les causes identifiées celles qui ont une incidence capitale sur le problème et nous établirons un plan d’actions visant à éliminer les causes. Bien entendu, il conviendra de mettre ce plan en œuvre, puis de vérifier régulièrement qu’il produit les effets escomptés à l’aide de l’indicateur choisi en début de phase.

La résolution d’un problème demande un délai qui peut varier d’un mois à une année selon la complexité du thème choisi et aussi parfois de la quantité d’énergie fournie par les participants du groupe de travail. [19]

2.4. Processus d’optimisation

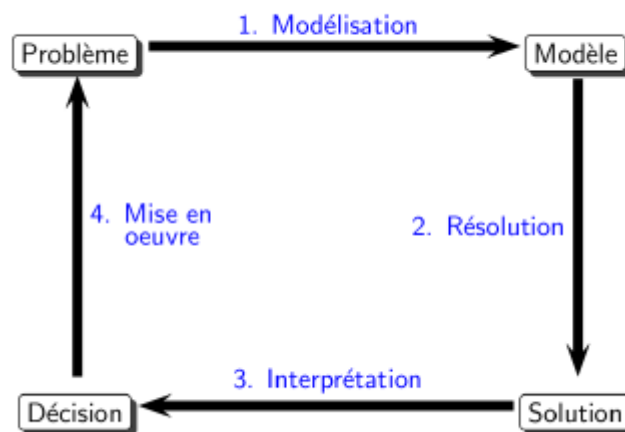


Figure 2.2 : les processus d’optimisation

2.4.1. Analyse du problème

- Compréhension du système : comprendre bien le problème étudié
- Définition des objectifs: Une fois que les phases de diagnostic et de priorisation sont réalisées il est possible de passer à la phase suivante qui est la définition des objectifs. Les objectifs répondent à la question "Pourquoi ce projet ? " et se déclinent à plusieurs niveaux:

un objectif général et des objectifs spécifiques. Les objectifs se formulent en verbe à l’infinitif, et en fonction de leur signification se traduit l’intention recherchée.

- Obtention des données : rechercher et grouper tous les données et les informations nécessaires pour le problème étudié.

2.4.2. Modélisation et choix de méthode

- Choix d'un langage (formel) : Un langage formel est un ensemble des mots. L'alphabet d'un langage formel est l'ensemble des symboles, lettres ou lexèmes qui servent à construire les mots du langage ; souvent, on suppose que cet alphabet est fini.
- Traduction du problème : paramètres, domaines, contraintes, incertitude, . . .
- simplification : résultat d'un consensus entre acteurs

2.4.3. Résolution

- Mise en œuvre algorithmique : concevoir un algorithme de résolution et en proposer une implémentation correcte. Du problème à sa solution, ce cours combine approches pragmatique, pratique et théorique de l'informatique.
- Analyse de la robustesse, pertinence des résultats c’est-à-dire l'évaluation de la performance, de l'efficacité, de la viabilité.

2.4.4. Interprétation des résultats

- Interpréter les résultats dans le monde réel : L’interprétation globale revient à une visualisation schématique ou bien une statistique des résultats.
- Présenter aux acteurs

2.4.5. Mise en œuvre

- Mise en œuvre opérationnelle
- Suivi des impacts, actions correctives
- peut donner lieu à la définition d'un nouveau problème !

2.5. Les méthodes de résolution des problèmes d’optimisation

La résolution de différentes sortes de problèmes rencontrés dans notre vie quotidienne a poussé les chercheurs à proposer des méthodes de résolution et à réaliser de grands efforts pour améliorer leurs performances en termes de temps de calcul nécessaire et/ou de la qualité de la solution proposée. Au fil des années, de nombreuses méthodes de résolution

de problèmes de différentes complexités ont été proposées. Ainsi, une grande variété et des différences remarquables au niveau du principe, de la stratégie et des performances ont été discernées. Cette variété et ces différences ont permis de regrouper les différentes méthodes de résolution de différents problèmes en deux classes principales: la classe de méthodes exactes et la classe de méthodes approchées. L’hybridation des méthodes de ces deux classes a donné naissance à un pseudo classe qui englobe des méthodes dites hybrides. (Voir figure 2.3).

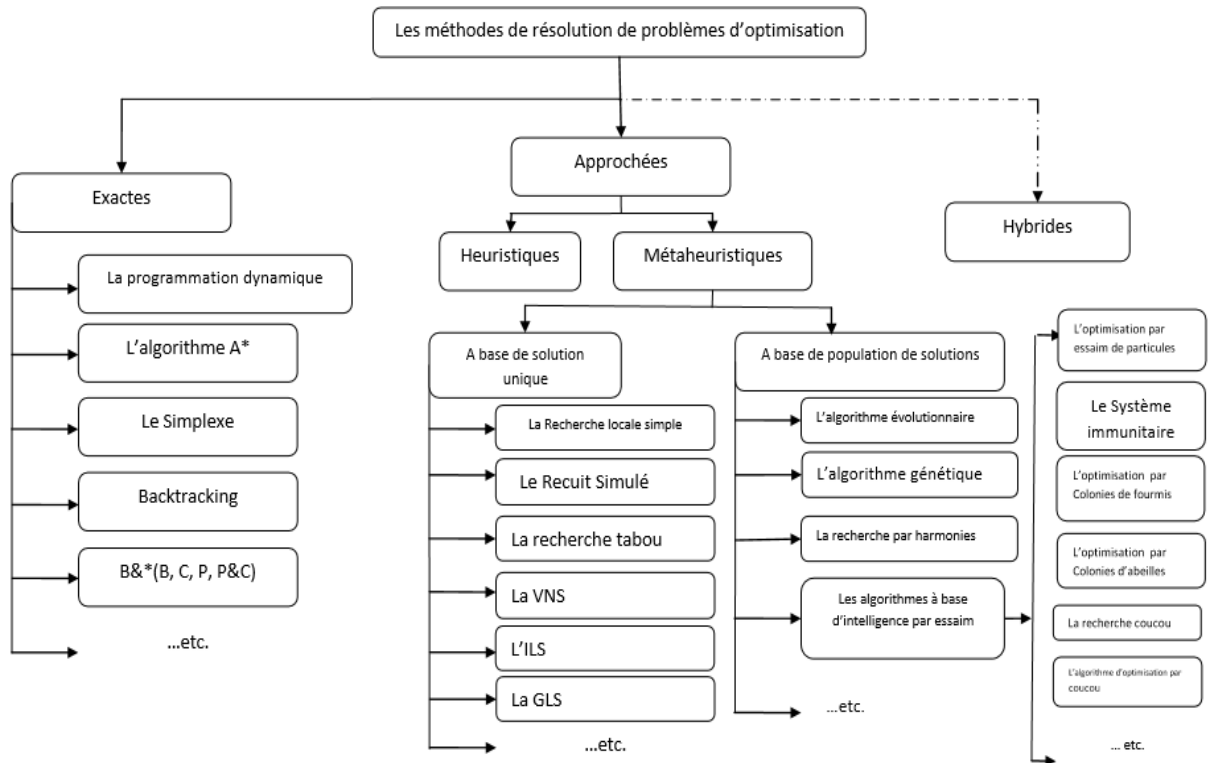


Figure 2.3 Les méthodes de résolution de problème d’optimisation [42]

2.5.1. Les méthodes exactes

L’intérêt des méthodes exactes réside dans le fait qu’elles assurent l’obtention de la solution optimale du problème traité. En fait, elles permettent de parcourir la totalité de l’ensemble de l’espace de recherche de manière à assurer l’obtention de toutes les solutions ayant le potentiel d’être meilleures que la solution optimale trouvée au cours de la recherche. Cependant, les méthodes exactes sont très connues par le fait qu’elles nécessitent un coût de recherche souvent prohibitif en termes de ressources requises. En effet, le temps de recherche et/ou l’espace mémoire nécessaire pour l’obtention de la solution optimale par une méthode exacte sont souvent trop grands, notamment avec des

problèmes de grandes tailles. De ce fait, la complexité de ce type d’algorithme croît exponentiellement avec la taille de l’instance à traiter, elle devient très importante face à des problèmes comprenant plusieurs variables, fonctions «objectif» et/ou critères. [42]

2.5.2. Méthodes approchées

Les méthodes approchées ont pour but de trouver une solution admissible en un temps raisonnable, mais sans garantie l’optimalité de cette solution. L’avantage principal de ces méthodes est qu’elles peuvent s’appliquer à n’importe quelle classe de problèmes, faciles ou très difficiles. De plus, elles ont démontré leurs robustesses et efficacités face à plusieurs problèmes d’optimisation combinatoires.

Elles englobent deux classes : Heuristiques & Métaheuristiques

2.5.2.1. les heuristiques

Les méthodes dites heuristiques sont des méthodes spécifiques à un problème particulier. Elles nécessitent des connaissances du domaine du problème traité. En fait, se sont des règles empiriques qui se basent sur l’expérience et les résultats acquis afin d’améliorer les recherches futures. Plusieurs définitions des heuristiques ont été proposées par plusieurs chercheurs dans la littérature, parmi lesquelles:

Définition (A): « Une heuristique (règle heuristique, méthode heuristique) est une règle d’estimation, une stratégie, une astuce, une simplification, ou tout autre type de dispositif qui limite considérablement la recherche de solutions dans des espaces problématiques importants. Les heuristiques ne garantissent pas des solutions optimales. En fait, elles ne garantissent pas une solution du tout. Tout ce qui peut être dit d’une heuristique utile, c’est qu’elle propose des solutions qui sont assez bonnes la plupart du temps. » [08]

Définition (B) : « Les heuristiques sont des critères, des méthodes ou des principes pour décider qui, parmi plusieurs d’autres plans d’action promet d’être le plus efficace pour atteindre un certain but. » [12]

2.5.2.2 les métaheuristiques

Dans la vie pratique, on se retrouve souvent confronté à des problèmes de différentes complexités, pour lesquelles on cherche des solutions qui satisfont deux notions antagonistes: la rapidité et la qualité. Devant le coût de recherche prohibitif des méthodes exactes (particulièrement avec des problèmes de grande taille) et la spécificité des heuristiques au problème donné, les métaheuristiques construisent une solution moins

exigeante. En fait, elles sont applicables sur une grande variété de problèmes d’optimisation de différentes complexités. En outre, elles permettent de fournir des solutions de très bonne qualité (pas nécessairement optimales) en temps de calcul raisonnable.

L’ensemble des métaheuristiques proposées dans la littérature sont partagées en deux classes: des métaheuristiques à base de solution unique et des métaheuristiques à base de population de solutions. Nous présentons quelques métaheuristiques des deux classes dans ce qui suit de cette section :

- **Les métaheuristiques à base de solution unique**

Dans cette section, nous présentons les métaheuristiques à base de solution unique, aussi appelées méthodes de trajectoire. Contrairement aux métaheuristiques à base de population, les métaheuristiques à solution unique commencent avec une seule solution initiale et s’en éloignent progressivement, en construisant une trajectoire englobant essentiellement la méthode de descente, la méthode du recuit simulé, la recherche taboue, la méthode GRASP, la recherche à voisinage variable, la recherche locale itérée, et leurs variantes.

- Méthode de descente : Un algorithme à directions de descente est un algorithme d’optimisation différentiable (l’optimisation dont il est question ici est une branche des mathématiques), destiné à minimiser une fonction réelle différentiable définie sur un espace euclidien ou, plus généralement, sur un espace hilbertien.
- Le recuit simulé : Le recuit simulé adapté aux problèmes continus a été utilisé pour résoudre le problème de segmentation par l’approche paramétrique. le recuit simulé permet d’approche de près la solution optimale du problème plus rapidement qu’une exploration exhaustive dans l’espace de recherche.
- La méthode de recherche avec tabous : La recherche tabou est une métaheuristique d’optimisation présentée par Fred Glover en 1986. On trouve souvent l’appellation recherche avec tabous en français. Cette méthode est une métaheuristique itérative qualifiée de recherche locale au sens large.
- La recherche à voisinage variable : (RVV) est une métaheuristique récente pour la résolution de problème d’optimisation combinatoire et globale, dont l’idée de base est le changement systématique de voisinage au sein d’une recherche locale. [45]

- **Les métaheuristiques à population de solutions**

Contrairement aux algorithmes partant d’une solution singulière, Les métaheuristiques à base de population de solutions débutent la recherche avec une panoplie de solutions. Elles s’appliquent sur un ensemble de solutions afin d’en extraire la meilleure. On distingue dans cette catégorie, les algorithmes évolutionnaires, qui sont une famille d’algorithmes issus de la théorie de l’évolution

par la sélection naturelle, énoncée par Charles Darwin [01], et les algorithmes d’intelligence en essaim qui, de la même manière que les algorithmes évolutionnaires, proviennent d’annalogies avec des phénomènes biologiques naturels.

○ **Les algorithmes évolutionnaires**

L’apparition des algorithmes évolutionnaires revient aux années soixante. Elles sont inspirées du principe de l’évolution naturelle des espèces biologiques. Précisément, de la théorie de l’évolution exposée par Charles Darwin qui se base sur le principe de la sélection naturelle. Le principe de cette dernière stipule que les individus bien adaptés à l’environnement ont plus de chances de se reproduire et de survivre que les autres individus. En fait, la combinaison des caractéristiques des individus peut former au fil des générations de nouveaux individus de plus en plus mieux adaptés à leur environnement et qui peuvent avoir plus de chances de survivre que leurs parents. [01]

Et le principe de cette dernière est de simuler l’évolution d’une population d’individus divers auxquels on applique différents opérateurs génétiques et que l’on soumet à chaque génération à une sélection. Ces algorithmes sont de plus en plus utilisés dans l’industrie car ils sont particulièrement adaptés aux problèmes d’optimisation comportant de nombreux paramètres.

Les algorithmes évolutionnaires s’inspire de l’évolution naturelle des êtres vivants. Ils adoptent une sorte d’évolution artificielle pour améliorer la qualité des individus de la population. En fait, ils font évoluer itérativement une population d’individus. Ces derniers représentent des solutions du problème traité. Les qualités des individus sont mesurées à chaque itération du processus de d’évolution. En fonction de leurs qualités, les meilleurs individus seront sélectionnés pour subir des combinaisons qui permettent la production d’une nouvelle population (dite: population d’enfants). Les individus (ou une partie des individus) de la nouvelle population vont remplacer les individus de la population courante (dite: population de parents) pour construire la nouvelle génération d’individus. Le processus d’évolution d’un algorithme évolutionnaire est basé sur trois opérations principales: la sélection, la reproduction et l’évaluation:

-La Sélection: Cette opération s’intervient dans deux phases. Elle est appliquée au premier lieu pour choisir les meilleurs individus parents qui vont se reproduire pour construire de nouveaux individus enfants. Ensuite, elle est appliquée à la fin de chaque itération pour opter pour les individus qui vont survivre et construire la nouvelle population.

- **La reproduction:** Cette opération est en général composée de deux autres opérations: le croisement et la mutation. Elle permet la génération de nouveaux individus en combinant (phase de croisement) les caractéristiques des individus sélectionnés puis en appliquant quelques modifications de certains individus (phase de mutation) pour améliorer leurs qualités.
- **L’évaluation:** Cette opération consiste à mesurer la qualité de chaque individu (calculer la fitness des individus).

La figure 2.4 représente un diagramme résumant les étapes générales d’un algorithme évolutionnaire.

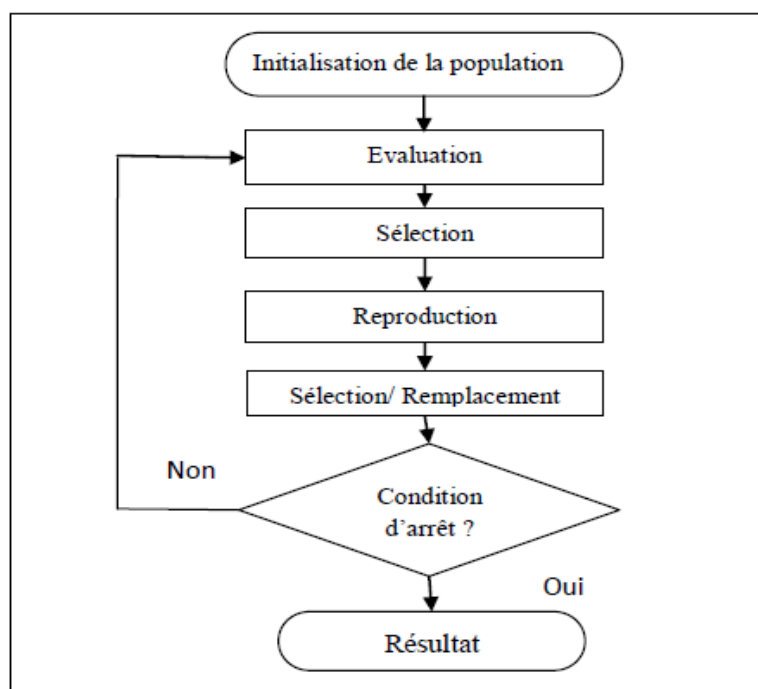


Figure 2.4 Organigramme d’un algorithme évolutionnaire [42]

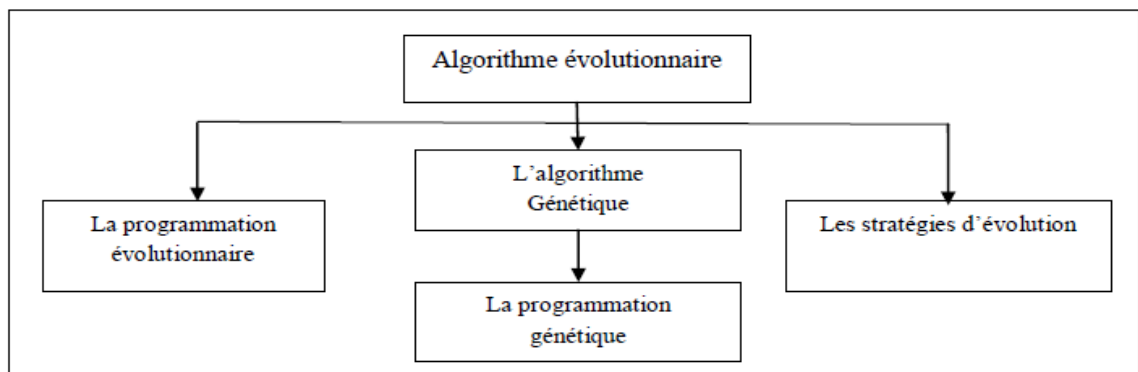


Figure 2.5 Les types des algorithmes évolutionnaires.[42]

Les algorithmes évolutionnaires forment une classe principale de trois sous classes d’algorithmes (voir figure 2.5): les stratégies d’évolution, la programmation évolutionnaire et les algorithmes génétiques.

- **Les stratégies d’évolution:** Ont été conçues pour la résolution des problèmes d’optimisation continus. [29] [10]

- **La programmation évolutionnaire:** Les algorithmes de la classe de la programmation évolutionnaire sont conçus pour faire évoluer des structures d’automates à état fini. [35]

- **L’algorithme génétique** [31] [06]: Il sera bien détaillé dans le chapitre suivant. Il est à noter qu’il existe une sous classe de la classe des algorithmes génétiques appelée « La programmation génétique» [11]. Cette dernière utilise des structures arborescentes pour représenter les individus de la population,

2.6. Conclusion

La programmation mathématique recouvre un ensemble de techniques d’optimisation sous contraintes qui permettent de déterminer dans quelles conditions on peut rendre maximum ou minimum une fonction objective, de nombreux problèmes de l’entreprise peuvent s’exprimer en termes d’optimisation contrainte, aussi rencontre-t-on de multiples applications de la programmation mathématique et ceci dans pratiquement tous les domaines de la gestion.

Dans ce chapitre, nous avons essayé d’aborder quelques notions de base en optimisation. Nous avons présenté des définitions de notions confrontées souvent dans le domaine de l’optimisation en allant de la définition d’un problème d’optimisation, passant par les types des problèmes d’optimisation jusqu’à la définition d’une solution optimale d’un problème donné. Ensuite, nous avons dévoilé les processus d’optimisation et Les méthodes de résolution de problèmes d’optimisation (Exacte et approchée) et nous avons spécifié les méthodes heuristiques et métaheuristiques.

Dans le chapitre qui suit, nous présentons la méthode d’optimisation « les algorithmes génétiques » utilisées pour résoudre notre problématique.