

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE: : MATHÉMATIQUES ET  
INFORMATIQUE  
DEPARTEMENT : INFORMATIQUE  
N° :.....



DOMAINE : mathématiques et  
Informatique  
FILIERE : Informatique  
OPTION : informatique décisionnelle et  
optimisation

Mémoire présenté pour l'obtention

Du diplôme de Master Académique

Par : Laziri Zineb et Taibi Rima

**Un système optimisé pour l'implémentation  
des caméras de surveillance.**

**Cas d'étude : Algérie Telecom de M'sila**

Intitulé

Soutenu devant le jury composé de :

Hemza loucif	Université de M'sila	Président
Tahar Mehenni	Université de M'sila	Encadreur
Aimad -Eddine Debbi	Université de M'sila	Examineur

Année universitaire : 2023/2024

## Dédicaces

Je voudrais consacrer ce travail à :

Ma mère sur le don divin d'amour et d'affection qu'elle a toujours porté pour moi, à savoir

Un soutien sans fin tout au long de ma vie ;

Mon père qui m'a soutenu et encouragé tout au long de mon parcours universitaire.

À tous mes frères, qui n'ont jamais cessé d'être les miens

Mohamed, Amina, Mounir, Assil

À toute ma famille.

À mes chers amis.

À toutes les personnes qui m'ont soutenu et qui ont participé à me soutenir

**Zineb**

# Dédicaces

Avec un énorme plaisir que je dédie ce travail

À mes très chers parents :

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma  
Considération pour les sacrifices que vous avez consenti pour mon instruction et

Mon bien être.

Je vous remercie pour tout le soutien, la patience et l'amour que vous me portez

Depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.

Aucun mot ne saurait exprimer ma reconnaissance et ma gratitude à votre égard.

Vous êtes les meilleurs parents du monde. Que ALLAH Tout Puissant vous garde et

Vous procure santé et bonheur

Mes très chères sœurs :

ROKIA, RACHA

Les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection

Que je porte pour vous. Mes fidèles accompagnantes dans les moments les plus

Déliçats. Vous avez toujours été présents pour les bons conseils. Votre affection

Et votre soutien m'ont été d'un grand secours. Veuillez trouver dans ce modeste

Travail ma reconnaissance pour tous vos efforts.

A toutes les personnes qui m'ont aidé de près ou de loin à réaliser ce travail.

**RIMA**

## **Remerciements**

Nous remercions ALLAH Seigneur du monde de nous avoir donné l'inspiration et la patience

Pour achever ce chapitre de notre vie.

Nous remercions notre encadreur Dr. Mehenni Tahar qui a suivi ce travail avec grand intérêt. Son regard critique et constructif, ses orientations avisées et pertinentes, sa rigueur ainsi que sa patience et son compréhension, nous le remercions du fond du coeur et nous lui exprimons notre respect.

# Table des matières

<b>Dédicaces</b> .....	2
<b>Dédicaces</b> .....	3
<b>Remerciements</b> .....	4
<b>Table des matières</b> .....	5
<b>Liste des figures</b> .....	8
<b>Liste des tables</b> .....	8
<b>Introduction générale</b> .....	1
<b>Chapitre 1</b> .....	3
<b>Le problème de placement des caméras de surveillance</b> .....	3
<b>1.1. Introduction :</b> .....	3
<b>1.2. Le système de la vidéo-surveillance :</b> .....	4
<b>1.3. Problème D'emplacement des caméras de surveillance :</b> .....	4
<b>1.4. Définition du Problème :</b> .....	5
<b>1.4.1 Optimisation de la Couverture de Surveillance (1) :</b> .....	5
<b>1.4.2 Défis Associés :</b> .....	7
<b>1.4.3 Considérations Financières :</b> .....	9
<b>1.5. Approches pour Résoudre le Problème</b> .....	9
<b>1.5.1 Analyse de Risques et Besoins :</b> .....	9
<b>1.5.2 Technologies de Modélisation et Simulation :</b> .....	10
<b>1.5.3 Considérations Pratiques :</b> .....	10
<b>1.5.4 Collaboration et Consultation :</b> .....	10
<b>1.6. Études de Cas et Applications Pratiques :</b> .....	10
<b>1.7. Description du Placement des Caméras de Surveillance</b> .....	11
<b>1.7.1 Analyse Préliminaire Objectifs de Surveillance :</b> .....	11
<b>1.7.2 Identification des Zones Critiques :</b> .....	11
<b>1.8. Considérations Techniques Type de Caméras :</b> .....	11
<b>1.9. Placement Stratégique Hauteur et Angle de Vue</b> .....	12
<b>1.10. Considérations Légales et Éthiques et Respect de la Vie Privée :</b> .....	14
<b>Conclusion</b> .....	15

<b>Chapitre 02.....</b>	<b>16</b>
<b>Les Méthodes d'optimisation .....</b>	<b>16</b>
<b>2.1 Introduction :.....</b>	<b>17</b>
<b>2.2 Optimisation Combinatoire : .....</b>	<b>17</b>
<b>2.2.1 Définition : .....</b>	<b>17</b>
<b>2.3. Classification des méthode d'optimisation : .....</b>	<b>18</b>
<b>2.3.1 Méthodes exactes :.....</b>	<b>18</b>
<b>2.3.2 Méthodes approchées :.....</b>	<b>21</b>
<b>2.4. Classification des métaheuristiques :.....</b>	<b>23</b>
<b>2.4.1 Métaheuristiques à solution unique :.....</b>	<b>23</b>
<b>2.4.2 Métaheuristique à populations : .....</b>	<b>29</b>
<b>2.6 Les opérateurs des algorithmes génétiques : .....</b>	<b>31</b>
<b>2.6.1 Population initiale : .....</b>	<b>31</b>
<b>2.6.2 Le codage : .....</b>	<b>32</b>
<b>2.6.3 L'évaluation : .....</b>	<b>33</b>
<b>2.6.4 La sélection : .....</b>	<b>34</b>
<b>2.6.5 Le croisement :.....</b>	<b>35</b>
<b>2.6.6 La mutation.....</b>	<b>36</b>
<b>2.7 Fonctionnement d'un Algorithme Génétique: .....</b>	<b>36</b>
<b>Conclusion : .....</b>	<b>38</b>
<b>Chapitre 3.....</b>	<b>39</b>
<b>Proposition d'une méthode de résolution d'emplacement des caméras par les algorithmes génétiques .....</b>	<b>39</b>
<b>3.1 Introduction :.....</b>	<b>40</b>
<b>3.2 État de l'art:.....</b>	<b>40</b>
<b>3.3 Méthode proposée : .....</b>	<b>42</b>
<b>3.3.1 Algorithme Génétique :.....</b>	<b>42</b>
<b>3.3.2 Application dans l'Optimisation d'Emplacement des Caméras de Surveillance :.....</b>	<b>43</b>
<b>3.3.3 L'utilisation des algorithmes génétiques pour l'optimisation des emplacements des caméras de surveillance: .....</b>	<b>44</b>
<b>3.3.4 Les étapes d'algorithme génétiques:.....</b>	<b>45</b>
<b>3.4 Langage et environnement .....</b>	<b>46</b>
<b>3.4.1 Python : .....</b>	<b>46</b>
<b>3.4.2 ANACONDA.....</b>	<b>48</b>

<b>3.4.3. Espace de travail :</b> .....	49
<b>Conclusion :</b> .....	51
<b>Chapitre 4</b> .....	52
<b>Implémentation de la méthode et résultats</b> .....	52
<b>4.1 Introduction :</b> .....	53
<b>4.2 Implémentation :</b> .....	53
<b>4.2.1 Cas d'étude :</b> .....	53
<b>4.2.2 Explication du code:</b> .....	53
<b>4.2.3 Les Résultats:</b> .....	59
<b>4.2.4 Discussion des résultats :</b> .....	65
<b>Conclusion :</b> .....	66
<b>Conclusion générale</b> .....	68
<b>Bibliographie :</b> .....	70
<b>Résumé :</b> .....	72

## Liste des figures

<b>Figure 1. 1</b> Cartographie 2D sur un Modèle Urbain avec Attribution Manuelle des Zones de Couverture CCTV .....	4
<b>Figure 1. 2</b> La ville urbaine est divisée en zones plus petites. Chaque zone est classée selon son niveau de risque respectif. ....	9
<b>Figure 1. 3</b> angle de vue et la distance de vision .....	12
<b>Figure 2. 1</b> Classification des méthodes d'optimisation 18	
<b>Figure 2. 2:</b> Principe des la programmation dynamique.....	20
<b>Figure 2. 3</b> Processus de diversification d'une solution .....	22
<b>Figure 2. 4</b> Processus d'intensification de plusieurs solutions .....	22
<b>Figure 2. 5</b> Principe des métaheuristiques à solution unique .....	23
<b>Figure 2. 6:</b> Pseudo-code de la recherche taboue .....	25
<b>Figure 2. 7 :</b> Principe de recuit simulé.....	27
<b>Figure 2. 8:</b> pseudo-code du recuit simulé.....	28
<b>Figure 2. 9:</b> Principe des Métaheuristique à populations.....	30
<b>Figure 2. 10:</b> Principe le codage arborescent .....	33
<b>Figure 2. 11:</b> Principe Croisement en un point .....	35
<b>Figure 2. 12:</b> Principe Croisement en deux points .....	36
<b>Figure 2. 13 :</b> Principe de mutation .....	36
<b>Figure 2. 14:</b> les etaps fonctionnement d'un Algorithme Génétique.....	37
<b>Figure 3. 1:</b> Logo Python.....	48
<b>Figure 3. 2 :</b> Logo ANACONDA .....	49
<b>Figure 3. 3 :</b> Logo Pycharm.....	50
<b>Figure 4. 1</b> l'enplacement des cameras dans RDC .....	61
<b>Figure 4. 2 :</b> l'enplacement des cameras dans 1etage.....	62
<b>Figure 4. 3 :</b> l'enplacement des cameras dans 2etage.....	64
<b>Figure 4. 4:</b> l'enplacement des cameras dans 3etage.....	65

## Liste des tables

<b>Table 4. 1 :</b> Résultat placement des cameras dans RDC.....	60
<b>Table 4. 2 :</b> Résultat placement des cameras dans 1etage.....	62
<b>Table 4. 3 :</b> Résultats placement des cameras dans 2etage .....	63
<b>Table 4. 4 :</b> Résultat placement des cameras dans 3etage.....	64



## Introduction générale

L'optimisation de l'emplacement des caméras de surveillance est devenue un enjeu majeur dans le domaine de la sécurité publique et privée, notamment avec l'augmentation des préoccupations relatives à la criminalité et à la protection des biens et des personnes. Dans un contexte où les ressources sont souvent limitées et les besoins en sécurité croissants, il est essentiel de disposer d'outils et de méthodes permettant de déployer des systèmes de surveillance de manière efficace et efficiente. L'objectif principal est de maximiser la couverture des zones critiques tout en minimisant les coûts et en respectant les contraintes spécifiques de l'environnement surveillé.

Les méthodes traditionnelles d'installation de caméras reposent souvent sur l'expertise humaine et des règles empiriques, qui peuvent manquer de rigueur scientifique et d'optimisation systématique. C'est dans ce cadre que les algorithmes d'optimisation, et plus particulièrement les algorithmes génétiques, trouvent leur pertinence. Inspirés par le processus de sélection naturelle, ces algorithmes offrent une approche puissante pour explorer de vastes espaces de solutions et trouver des configurations optimales pour divers problèmes complexes.

L'utilisation des algorithmes génétiques pour l'optimisation de l'emplacement des caméras de surveillance est une approche innovante qui combine la flexibilité et l'efficacité des méthodes computationnelles avancées. Ces algorithmes permettent de modéliser de manière précise les contraintes spécifiques d'un environnement, telles que la présence d'obstacles, la nécessité de couvrir certaines zones plus intensivement que d'autres, et les limitations budgétaires.

L'intérêt croissant pour cette méthode s'explique par sa capacité à produire des solutions robustes et adaptatives. Contrairement aux méthodes heuristiques traditionnelles, les algorithmes génétiques ne se contentent pas de chercher une solution locale mais explorent de manière systématique l'ensemble de l'espace de recherche pour identifier des solutions globalement optimales. Ils sont particulièrement adaptés aux problèmes de grande taille et aux environnements dynamiques, où les conditions peuvent changer et nécessiter une réadaptation rapide des stratégies de surveillance.

Dans ce mémoire, nous explorons en détail comment les algorithmes génétiques peuvent être appliqués à l'optimisation de l'emplacement des caméras de surveillance. Nous présentons les fondements théoriques de ces algorithmes, leur mise en œuvre pratique, et analysons les résultats obtenus à travers diverses études de cas. Notre objectif est de démontrer que les algorithmes génétiques constituent non seulement une solution viable mais également une avancée significative dans la planification stratégique des systèmes de surveillance. En combinant la puissance des algorithmes d'optimisation avec une approche rigoureuse de la modélisation des contraintes spécifiques, nous visons à fournir des outils pratiques et efficaces pour améliorer la sécurité dans divers contextes.

## **Chapitre 1**

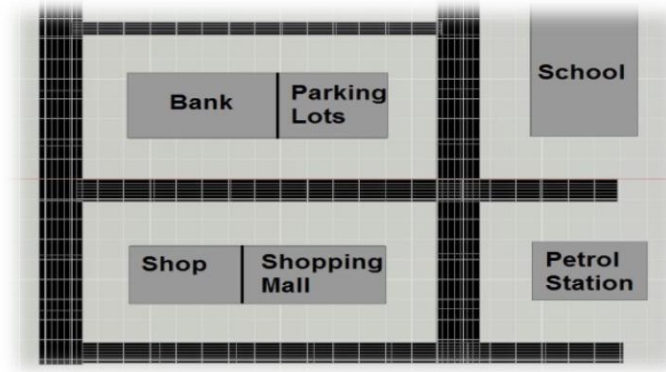
### **Le problème de placement des caméras de surveillance**

## 1.1. Introduction :

La vidéo-surveillance connaît aujourd'hui une forte expansion, tant sur le plan technologique qu'économique. Elle est devenue un élément crucial des politiques de sécurité des gouvernements. Cette évolution répond au besoin de sécurité des citoyens face à l'augmentation de la délinquance et de la criminalité. Dans ce chapitre, nous définirons le système de vidéo-surveillance, expliquerons certains concepts utilisés dans ce domaine, et présenterons les avantages de ce système dans notre vie quotidienne.

La vidéosurveillance est utilisée pour surveiller de près et enregistrer des images de tout individu ou groupe suspect. Ceci peut être réalisé en utilisant des caméras telles que des caméras de télévision en circuit fermé (En anglais : Closed Circuit Television Cameras (CCTV)). Aujourd'hui, la vidéosurveillance est largement installée dans de nombreux lieux publics et a considérablement réduit les taux de criminalité au fil des années. La plupart des caméras de vidéosurveillance ne sont pas situées dans des zones considérées comme des « points chauds » publics. Bien qu'ils soient installés dans des zones à haut risque, ils peuvent ne pas couvrir adéquatement certaines zones dangereuses. En outre, il manque des indications permettant d'évaluer si le nombre et l'emplacement des caméras de vidéosurveillance installées offrent une couverture adéquate et optimale. Actuellement, la couverture CCTV est principalement déterminée par les images capturées sur les moniteurs.

L'emplacement et l'orientation de la plupart des systèmes de vidéosurveillance sont déterminés sans planification. Les zones dangereuses peuvent ne pas avoir une couverture adéquate. D'un autre côté, une couverture CCTV inutile dans des zones moins dangereuses entraînera des dépenses inutiles. Une fois qu'une caméra de vidéosurveillance est installée dans un espace public, son déplacement peut s'avérer coûteux. Grâce à des cartes de risques 2D, nous sommes en mesure d'évaluer le développement depuis la conception initiale jusqu'à la construction urbaine. L'évaluation vise à optimiser l'utilisation des caméras de vidéosurveillance dans les espaces publics. La portée de cet article est l'attribution manuelle des zones de couverture CCTV pour la cartographie 2D d'un modèle de ville à l'aide du traitement d'image. Cet article se concentre sur le placement de caméras fixes sur la base de cartes de risques générées à partir d'un modèle de ville typique, comme le montre la figure 1.1 [11].



**Figure 1. 1** Cartographie 2D sur un Modèle Urbain avec Attribution Manuelle des Zones de Couverture CCTV

## **1.2. Le système de la vidéo-surveillance :**

La vidéo-surveillance consiste en un réseau de caméras installées dans un espace à surveiller. Ces caméras sont connectées à un système informatique permettant le traitement et l'analyse des données recueillies. Le premier système de vidéo-surveillance, conçu par Siemens AG en Allemagne en 1942, était destiné à l'observation des fusées. Depuis lors, ces systèmes ont considérablement évolué. L'analyse et l'intégration des données sont de plus en plus automatisées, nécessitant ainsi moins d'intervention humaine. [12].

## **1.3. Problème D'emplacement des caméras de surveillance :**

L'emplacement des caméras est essentiel pour l'efficacité globale des systèmes de surveillance. Un mauvais placement des caméras peut entraîner des angles morts, des zones non couvertes et une qualité d'image médiocre, compromettant ainsi la capacité du système à détecter et à répondre aux menaces potentielles. En revanche, un placement stratégique des caméras permet une couverture optimale de la zone surveillée, une meilleure identification des individus et une réponse plus rapide aux incidents. Par conséquent, l'emplacement des caméras doit être soigneusement planifié et optimisé pour garantir une surveillance efficace et une sécurité maximale.

L'essor exponentiel de la technologie de surveillance par caméra a profondément remodelé notre façon de sécuriser et de protéger nos espaces publics et privés. Du simple commerce de détail aux vastes infrastructures urbaines, la surveillance par caméra est devenue un pilier

fondamental de la sécurité moderne. Cependant, derrière chaque installation de caméra se cache une complexité souvent négligée : le placement stratégique des caméras de surveillance.

Chaque décision de placement de caméra doit prendre en compte une multitude de facteurs, allant de la topographie du terrain et de la disposition architecturale aux modèles de comportement humain et aux risques potentiels. Les exigences légales et les contraintes réglementaires ajoutent une autre couche de complexité, dictant où les caméras peuvent être installées et dans quelles conditions elles peuvent être utilisées. De plus, l'avènement de nouvelles technologies telles que l'intelligence artificielle et l'analyse vidéo avancée a ouvert de nouvelles perspectives dans le domaine du placement des caméras.

Des algorithmes sophistiqués peuvent maintenant être utilisés pour optimiser la couverture de la caméra, minimiser les angles morts et améliorer la précision de la surveillance, ouvrant ainsi la voie à des niveaux de sécurité et de protection plus élevés.

L'emplacement des caméras de surveillance est une question complexe qui implique de multiples considérations techniques, éthiques et légales. Voici une présentation structurée du problème, des défis associés, et des éléments à prendre en compte pour résoudre cette problématique :

## **1.4. Définition du Problème :**

### **1.4.1 Optimisation de la Couverture de Surveillance (1) :**

Le problème consiste à déterminer le placement optimal d'un nombre limité de caméras dans un environnement donné, afin de maximiser la couverture de la zone sous surveillance tout en minimisant l'erreur de mesure 3D [4]. Cela implique de résoudre un problème d'optimisation combinatoire avec les contraintes suivantes :

- Minimiser le nombre de caméras nécessaires pour couvrir l'environnement, tout en assurant une couverture suffisante [2].
- Minimiser l'erreur de mesure 3D en optimisant la configuration géométrique du réseau de caméras [4].
- Satisfaire les contraintes liées à la distribution des caméras (champ de vision, occultations, etc.) [2][4].

La méthode proposée pour résoudre ce problème se décompose en deux parties :

1. Une partie analytique qui définit un critère d'optimisation basé sur une modélisation géométrique permettant d'estimer l'erreur 3D en fonction de la configuration des caméras
2. Un processus d'optimisation combinatoire, comme les algorithmes génétiques, qui minimise ce critère pour produire une solution optimale

Donc il s'agit de déterminer le placement et les paramètres (orientation, zoom, etc.) d'un nombre limité de caméras haute performance (haute définition et/ou haute vitesse) afin de maximiser la couverture d'un environnement complexe tout en garantissant une précision de mesure 3D optimale [1][2][3][4].

Pour procéder au développement de la méthode proposée, il convient de définir certaines limitations :

- Le placement des caméras CCTV est un système autonome et ne devrait pas modifier les structures des bâtiments ou l'urbanisme.
- Seules des caméras CCTV fixes (sans inclinaison ni rotation) sont utilisées. L'angle de placement des caméras CCTV est fixe.
- La vue de la caméra est représentée par un champ de vision en forme de cône vue depuis le dessus.
- L'angle de champ de vision et le placement de la caméra CCTV ne sont pas fixes et peuvent être définis par l'utilisateur.
- Le nombre maximum de caméras CCTV pouvant être attribuées est de quatre.
- La caméra CCTV a ses propres spécifications, mais pour cet chapitre, seule l'angle de vue est prise en compte, ce qui déterminera la sélection de sa taille de format et de sa longueur focale [11].

## **1.4.2 Défis Associés :**

### **1.4.2.1 Complexité de l'Environnement :**

#### **Variabilité des Lieux :**

Les environnements urbains et ruraux présentent des défis différents. Les espaces publics comme les parcs, les centres commerciaux, les intersections de rues et les stations de transport nécessitent des stratégies spécifiques.

#### **Obstacles physiques :**

Lorsqu'on décide des emplacements pour les caméras de surveillance, tenir compte des obstacles est crucial pour garantir une surveillance efficace. Les obstacles physiques comme les bâtiments, les arbres, les poteaux et les véhicules peuvent obstruer la vue des caméras et compromettre la qualité de la surveillance. Il est essentiel d'identifier ces obstacles lors de l'analyse de la zone à surveiller. Une évaluation minutieuse de la ligne de vue permet de déterminer comment chaque obstacle affecte le champ de vision des caméras, en repérant les angles morts ou les zones d'ombre potentielles.

Pour surmonter ces défis, un positionnement stratégique des caméras est nécessaire. Il est recommandé de choisir des emplacements qui contournent autant que possible les obstacles, garantissant ainsi une vue dégagée de la zone surveillée. Dans certains cas, l'utilisation de caméras à angle large ou panoramique peut être envisagée pour compenser les obstacles et offrir une couverture étendue de la zone [10].

### **1.4.2.2 Considérations Techniques :**

**Types des Caméras :** Différents types de caméras (fixes, PTZ, infrarouges, etc.) ont des capacités et des limitations distinctes. Le choix et le positionnement doivent tenir compte des caractéristiques techniques [8].

**Eclairages et considérations Météorologiques :** L'éclairage est un facteur important à considérer lors du choix des emplacements pour les caméras de surveillance. Un éclairage adéquat est essentiel pour garantir une qualité d'image optimale, permettant ainsi une surveillance efficace des zones cibles. Voici quelques éléments à prendre en compte lors de la planification de

l'emplacement des caméras en relation avec l'éclairage. Évaluer le niveau d'éclairage naturel ou artificiel présent dans la zone à surveiller.

Un éclairage adéquat contribue à une meilleure qualité d'image et facilite la détection des détails importants, même dans des conditions de faible luminosité et identifier les sources potentielles de lumière dans la zone, telles que les lampadaires, les lampes extérieures, les enseignes lumineuses ou les fenêtres éclairées. Les caméras doivent être placées de manière à bénéficier de l'éclairage direct lorsque cela est possible. Éviter les emplacements où l'éclairage provient de l'arrière de la caméra, ce qui pourrait entraîner des images surexposées ou des silhouettes sombres et pour les zones surveillées la nuit, envisager l'installation de caméras dotées de capteurs infrarouges ou de vision nocturne. Ajuster les paramètres de la caméra en fonction des conditions d'éclairage pour maintenir un bon contraste et une balance des blancs appropriée.

Assurer que les sources d'éclairage fonctionnent correctement et sont bien entretenues pour éviter les pannes imprévues. Un éclairage défaillant peut compromettre la qualité de l'image et réduire l'efficacité de la vidéosurveillance. En tenant compte de ces facteurs liés à l'éclairage lors du choix des emplacements pour les caméras de surveillance, il est possible d'optimiser la qualité des images capturées et d'assurer une surveillance efficace des zones surveillées, contribuant ainsi à renforcer la sécurité des espaces et les conditions météorologiques affectent la qualité des images capturées.

#### **1.4.2.3 Aspect Légaux et éthiques :**

##### **Vie Privé :**

Il est crucial de respecter la vie privée des individus. Les caméras ne doivent pas surveiller des zones privées telles que les résidences, les toilettes publiques, ou les espaces de détente privée.

**Réglementations :** Conformité aux lois et réglementations locales, nationales et internationales sur la surveillance et la protection des données.



### **1.5.2 Technologies de Modélisation et Simulation :**

**Simulations Informatiques :** Utiliser des logiciels de simulation pour modéliser différents scénarios et déterminer les emplacements optimaux.

**Algorithmes d'Optimisation :** Appliquer des algorithmes (comme les modèles de couverture ou les algorithmes génétiques) pour optimiser la répartition des caméras.

### **1.5.3 Considérations Pratiques :**

**Hauteur et Angle de Vue :** Positionner les caméras à une hauteur et avec un angle qui minimisent les zones d'ombre et maximisent la couverture. La zone de couverture par la caméra CCTV, représentée dans un cône, est classée en trois parties pour déterminer les niveaux de couverture élevés, modérés et faibles, comme illustré dans la Figure 3 . Le niveau de couverture indique l'intensité de la mise au point et la netteté de l'image produite. La couverture est générée par un algorithme, ainsi que le niveau de couverture qui lui est attribué, avec sa valeur représentée en niveaux de gris.

**Discrétion vs. Dissuasion :** Décider si les caméras doivent être visibles pour dissuader les comportements criminels ou discrètes pour une surveillance plus efficace.

### **1.5.4 Collaboration et Consultation :**

**Engagement des Parties Prenantes de planification :** Impliquer les autorités locales, les experts en sécurité, et la communauté dans le processus.

**Retour d'Expérience :** Utiliser les retours d'expérience des installations existantes pour améliorer les décisions de positionnement.

## **1.6. Études de Cas et Applications Pratiques :**

**Études de Cas Réels :** Analyser des exemples concrets où des systèmes de caméras de surveillance ont été installés avec succès. Étudier les améliorations apportées par les ajustements d'emplacement et les technologies utilisées.

**Innovations Technologiques :** Intégrer des technologies émergentes telles que les drones, les caméras mobiles et les systèmes de surveillance intelligents pour une flexibilité accrue. Les

caméras de surveillance modernes sont dotées de fonctionnalités avancées telles que la détection de mouvement, la vision nocturne, la résolution HD, la connectivité Wi-Fi, etc. Utiliser ces fonctionnalités pour améliorer la surveillance et faciliter l'accès aux enregistrements [8]. Utiliser également l'intelligence artificielle pour l'analyse en temps réel des images et l'optimisation continue des positions des caméras.

## **1.7. Description du Placement des Caméras de Surveillance**

### **1.7.1 Analyse Préliminaire Objectifs de Surveillance :**

**Sécurité Publique** : Prévenir les crimes, surveiller les espaces publics.

**Protection des Biens** : Protéger les installations sensibles, les entrepôts, les magasins.

**Contrôle d'Accès** : Surveiller les entrées et sorties des bâtiments, les zones de stationnement.

### **1.7.2 Identification des Zones Critiques :**

**Points d'Entrée et de Sortie** : Portes principales, portails, fenêtres accessibles.

**Zones à Risque Élevé** : Caisse, entrepôts, zones de stationnement.

**Espaces Publics** : la gare, parcs, rues fréquentées.

## **1.8. Considérations Techniques Type de Caméras :**

**Caméras Fixes** : Pour surveiller une zone spécifique sans mouvement.

**Caméras PTZ (\*) (Pan-Tilt-Zoom)** : Pour couvrir de grandes zones avec la possibilité de pivoter et zoomer.

**Caméras Infrarouges** : Pour une surveillance nocturne ou dans des conditions de faible éclairage.

**Caméras IP** : Pour une intégration facile aux systèmes de surveillance en réseau.

Caractéristiques Techniques :

**Résolution** : Une haute résolution pour une identification claire.

**Angle de Vue** : Un angle large pour couvrir plus d'espace ou un angle étroit pour des détails spécifiques.

**Distance Focale** : Ajustée en fonction de la distance entre la caméra et la zone surveillée.

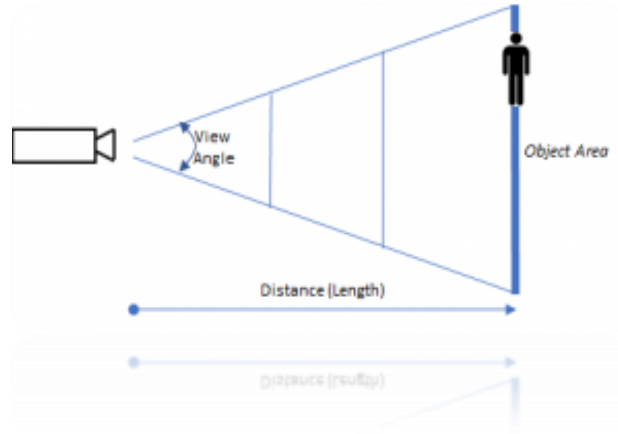


Figure 1. 3 Angle de vue et distance de vision

### 1.9. Placement Stratégique Hauteur et Angle de Vue

Le placement des caméras de surveillance dépend de plusieurs facteurs, tels que l'objectif de surveillance, la zone à couvrir et les contraintes environnementales. Assurez-vous que l'angle de vue couvre toute la zone souhaitée sans aucun angle mort. Généralement, une hauteur de 2,5 à 3 mètres est recommandée. Assurez-vous que les caméras sont positionnées de manière à éviter les reflets, les contre-jours ou les zones d'ombre.

Il est recommandé de consulter un professionnel de la sécurité ou un installateur de systèmes de surveillance pour obtenir des conseils spécifiques en fonction de vos besoins et de votre environnement [8].

Le positionnement stratégique des caméras de surveillance constitue un pilier fondamental de tout système de sécurité efficace. En choisissant avec soin les emplacements des caméras, il est possible d'optimiser la surveillance de manière à obtenir une couverture maximale de la zone tout en minimisant les angles morts et les zones non surveillées. Cette approche présente plusieurs avantages significatifs :

1. **Dissuasion des comportements indésirables** : Les caméras placées de manière visible dissuadent souvent les comportements criminels et non désirés. La simple présence de caméras peut décourager les individus d'agir de manière illicite, contribuant ainsi à prévenir les incidents.
2. **Détection précoce des incidents** : En surveillant stratégiquement les points d'accès, les zones sensibles et les zones à risque, il est possible de détecter rapidement les comportements suspects ou les situations d'urgence. Cela permet une réponse rapide et appropriée aux incidents potentiels, réduisant ainsi les dommages et les risques pour la sécurité.
3. **Collecte de preuves** : Les caméras positionnées de manière stratégique fournissent des enregistrements vidéo précieux pouvant être utilisés comme preuves dans le cadre d'enquêtes criminelles ou de litiges. Ces enregistrements peuvent aider à identifier les auteurs, documenter les événements et soutenir les poursuites judiciaires.
4. **Optimisation des ressources de sécurité** : En concentrant la surveillance sur les zones à risque élevé, l'emplacement stratégique des caméras permet une utilisation plus efficace des ressources de sécurité. Cela permet de maximiser l'efficacité opérationnelle et de garantir une réponse rapide et coordonnée aux menaces éventuelles. le positionnement stratégique des caméras de surveillance est essentiel pour assurer une protection efficace des espaces surveillés. En exploitant ces avantages, il est possible de renforcer la sécurité, de dissuader les comportements criminels et de fournir des preuves cruciales en cas d'incident.
5. **Dissuasion de la criminalité** : Placer des caméras dans des endroits visibles peut réduire les activités criminelles. Les potentiels délinquants sont moins enclins à commettre des infractions lorsqu'ils savent qu'ils sont surveillés. Une étude par la University of North Carolina at Charlotte a révélé que 60 % des cambrioleurs évitent les cibles avec des systèmes de sécurité visibles. [3]
6. **Appui aux enquêtes criminelles** : Des caméras bien positionnées peuvent fournir des preuves cruciales pour identifier et appréhender les suspects après un crime. Selon une analyse du Bureau of Justice Assistance, les images de vidéosurveillance ont été déterminantes dans 93 % des affaires résolues par les forces de l'ordre. [6]

7. **Amélioration de la sécurité publique** : Les caméras placées dans des lieux publics stratégiques, comme les parcs ou les stations de transport en commun, augmentent la sécurité des citoyens. Elles permettent une surveillance constante et une intervention rapide en cas d'incidents. [2]
8. **Productivité au travail** : Dans un environnement professionnel, les caméras stratégiquement placées peuvent dissuader les comportements inappropriés et améliorer la productivité. Une étude de la Harvard Business Review a démontré que la surveillance vidéo peut réduire le temps improductif des employés de 22 %.[4]
9. **Réduction du harcèlement au travail** : Les caméras peuvent jouer un rôle crucial dans la prévention et le suivi des cas de harcèlement au travail. La présence de surveillance vidéo et audio contribue à documenter et dissuader de tels comportements. [8]

#### **Positionnement dans l'Environnement:**

- Entrées et Sorties : Caméras placées au-dessus ou à côté des portes pour capturer les visages.
- Couloirs et Escaliers : Caméras à l'entrée des couloirs et en haut des escaliers pour surveiller le flux de personnes.
- Parkings : Caméras placées pour couvrir les entrées, sorties et allées principales des parkings.

#### **1.10. Considérations Légales et Éthiques et Respect de la Vie Privée :**

Au cœur de la lutte contre l'insécurité, la vidéosurveillance constitue une solution ultime à bien des égards. Cependant, comme le professe John Stuart Mill dans son célèbre opus sur la Liberté : «la liberté des uns s'arrête là où commence celle des autres » , chacun est libre de surveiller à sa guise tout en veillant à respecter le droit fondamental d'autrui : celui de l'intimité de la vie privée [7].

**Zones Non Surveillées** : Éviter les zones privées comme les toilettes, les vestiaires, et les résidences privées.

**Signalisation** : Informer le public de la présence de caméras de surveillance via des panneaux clairement visibles.

**Conformité Réglementaire** : Réglementations Locales : Se conformer aux lois et règlements locaux concernant la surveillance vidéo.

**Protection des Données** : Garantir que les données collectées sont sécurisées et utilisées conformément aux lois sur la protection des données.

**Optimisation et Plan de Maintenance :**

- **Inspections Régulières** : Vérifier le bon fonctionnement des caméras et des systèmes de stockage.
- **Nettoyage et Réparation** : Assurer le nettoyage des lentilles et la réparation des équipements défectueux.
- **Évaluation Continue** : Analyser les images pour identifier les zones non couvertes ou les problèmes de qualité d'image.
- **Ajustements Nécessaires** : Modifier l'emplacement ou l'angle des caméras en fonction des besoins évolutifs.

**Conclusion**

Le placement des caméras de surveillance doit être soigneusement planifié pour maximiser la couverture et l'efficacité tout en respectant les contraintes éthiques et légales. Une analyse approfondie des besoins, une compréhension des caractéristiques techniques des caméras, et une attention aux détails pratiques et légaux sont essentiels pour concevoir un système de surveillance performant et respectueux des droits des individus.

## **Chapitre 02**

### **Les Méthodes d'optimisation**

## **2.1 Introduction :**

Les problèmes d'optimisation occupent actuellement une place importante dans la communauté scientifique où il est nécessaire de trouver des solutions optimales pour des problèmes très compliqués et difficiles à résoudre. Parmi les méthodes les plus utilisées pour la résolution de tels problèmes, on trouve les métaheuristiques qui englobent un ensemble d'algorithmes capables de résoudre des problèmes assez compliqués en s'inspirant d'analogies avec la physique (recuit simulé), avec la biologie (algorithmes évolutionnaires) ou encore l'éthologie (colonies de fourmis, essaims particulaires).

Dans ce chapitre, nous allons présenter le concept d'optimisation combinatoire et ses différentes techniques, où nous allons nous concentrer sur les métaheuristiques vu que c'est la méthode d'optimisation que nous allons utiliser pour générer notre modèle de détection d'intrusions. Plus particulièrement, les algorithmes génétiques qui sont utilisés dans ce travail.

## **2.2 Optimisation Combinatoire :**

### **2.2.1 Définition :**

L'optimisation combinatoire est minimiser ou maximiser une fonction souvent appelée fonction coût, d'une ou plusieurs variables soumises à des contraintes. 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 et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire [13].

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 [12].

### 2.3. Classification des méthode d'optimisation :

Les méthodes de l'optimisation combinatoire peuvent être classées en deux grandes familles de classes : les méthodes exactes et les méthodes approchées (Figure 2.1).

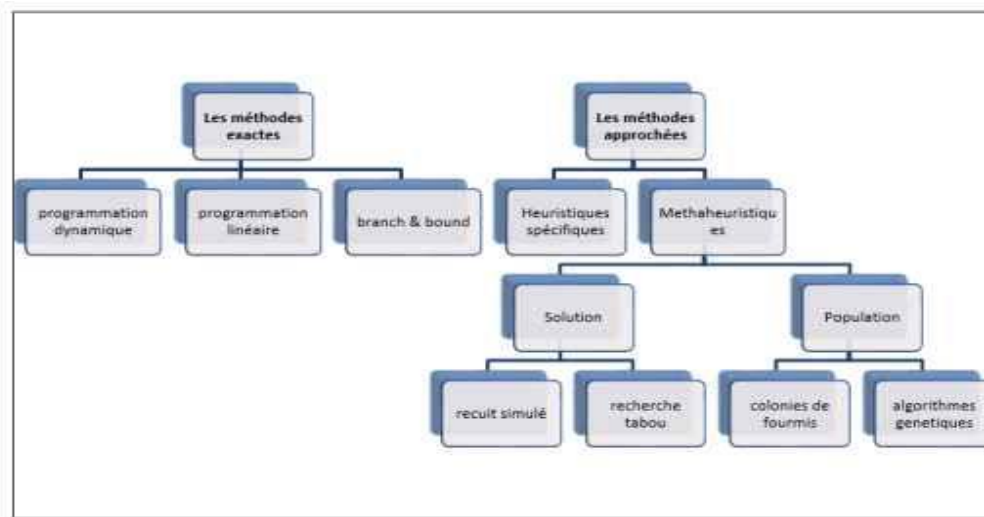


Figure 2. 1 Classification des méthodes d'optimisation

#### 2.3.1 Méthodes exactes :

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible les échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles les techniques de séparation et évaluation (Branch and Bound) ou la programmation dynamique. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. [14]

##### a) Branch and Bound:

La méthode Branch and Bound (procédure par évaluation et séparation progressive) consiste à énumérer ces solutions d'une manière intelligente en ce sens que, en utilisant certaines propriétés du problème en question, cette technique arrive à éliminer des solutions partielles. De ce fait, on

arrive souvent à obtenir la solution recherchée à réduire le temps de résolution en des temps raisonnables. Bien entendu, dans le pire cas, on retombe toujours sur l'élimination explicite de toutes les solutions du problème [15]. Pour ce faire, cette méthode se dote d'une fonction qui permet de mettre une borne sur certaines solutions pour soit les exclure soit les maintenir comme des solutions potentielles. Bien entendu, La performance d'une méthode Branch and Bound dépend, entre autres, de la qualité de cette fonction (de sa capacité d'exclure des solutions partielles têt).

**Algorithme général:** [15]

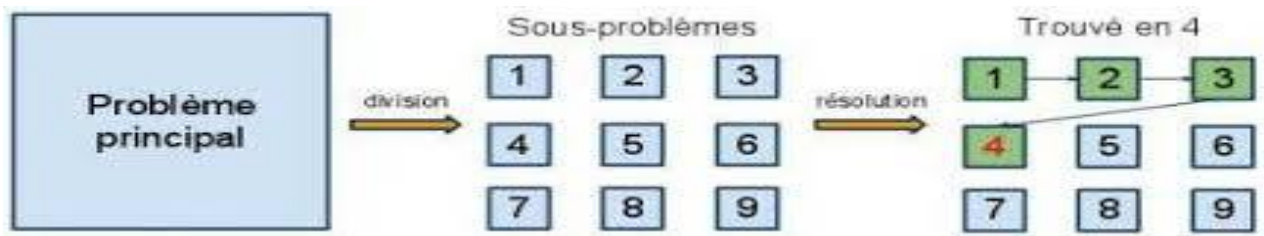
**Début**

- 1 Définir  $L = X$  et initialiser  $\hat{w}$ .
- 2 Tant que  $L \neq \emptyset$ .
- 3 Sélectionner un sous-problème  $S$  de  $L$  à explorer.
- 4 Si une solution  $\hat{w}_0 \in w \in S \mid f(w) < f(\hat{w})$  peut être Trouvée  $\hat{w} = \hat{w}_0$ .
- 5 Si  $S$  ne peut pas être élagué :
  - 6 Partitionner  $S$  en  $S1, S2, \dots, Sr$ .
  - 7 Insérer  $S1, S2, \dots, Sr$  dans  $L$ .
  - 8 Retirer  $S$  de  $L$ .
- 9 Fin Si
- 10 Fin Si
- 11 Fin Tant que
- 12 Retour  $\hat{w}$ .

**Fin**

**b) La programmation dynamique:**

La programmation dynamique a été introduite depuis 1940 par Richard Bellman. Le concept de base est simple : une solution optimale est le résultat obtenue à partir des solutions des sous-problèmes résolus de façon optimale. Il faut donc diviser un problème donné en sous-problèmes et les résoudre un par un. [15 (Figure 2.2)]



**Figure 2. 2:** Principe des la programmation dynamique

**Algorithme général de programmation dynamique :**

**Début**

- 1 Caractériser la structure d'une solution optimale.
- 2 Définir récursivement la valeur d'une solution optimale.
- 3 Calculer la valeur d'une solution optimale en remontant progressivement jusqu'à l'énoncé du problème initial.
- 4 Construire une solution optimale pour les informations calculées.

**Fin**

**c) Algorithme A\*:**

Stewart et White proposèrent une version multi objectif de l'algorithme A\* (A\*MO). L'algorithme A\* est maintenu dans sa forme originale. Le poids d'un arc  $(u, v)$  dans le graphe de recherche, correspond à un tuple  $(c1, c2, \dots, cn)$  où  $ci$  désigne le coût relatif à l'objectif  $fi$  induit par le passage de  $u$  à  $v$ . le coût d'un chemin, correspond à la somme vectorielle des poids des arcs qui le composent. Les fonctions  $k^*$ ,  $g^*$ ,  $h^*$  prennent une forme non scalaire où  $k^*(u, v)$  désigne l'ensemble des coûts non dominés des chemins reliant  $u$  à  $v$ ,  $g^*(u)$  représente le coût de la recherche ayant abouti à la solution intermédiaire  $u$ ,  $h^*(u)$  désigne l'ensemble des vecteurs

coût non dominés parmi l'ensemble  $k * (u, v)/v \in T$  où  $T$  représente l'ensemble des états terminaux. Les nœuds du graphe sont triés et parcourus comme pour l'algorithme A\* classique. L'ordre de tri est décrit par les relations de dominance entre les vecteurs coût  $f$  [16].

### **2.3.2 Méthodes approchées :**

Lorsque l'on dispose d'un temps de calcul limité ou lorsqu'on est confronté à des problèmes difficiles ou de taille importante, on peut avoir recours aux méthodes approchées, en se contentant de rechercher une solution de bonne qualité. Dans ce cas le choix est parfois possible entre une heuristique spécialisée et une métaheuristique :

#### **2.3.2.1 Heuristique :**

Le mot heuristique vient du grec « eusko » qui signifie « je trouve » d'où la célèbre Eureka d'Archimède. Une heuristique, ou méthode approximative, est un algorithme qui fournit rapidement (en temps polynomial) une solution réalisable, pas nécessairement optimale, pour un problème d'optimisation difficile. Une méthode heuristique est généralement conçue pour un problème particulier, en s'appuyant sur sa structure propre.

#### **2.3.2.1 Métaheuristicques :**

Une métaheuristique est un algorithme d'optimisation visant à résoudre des problèmes d'optimisation difficiles (souvent issus des domaines de la recherche opérationnelle, de l'ingénierie ou de l'intelligence artificielle) pour lesquels on ne connaît pas de méthode classique plus efficace.

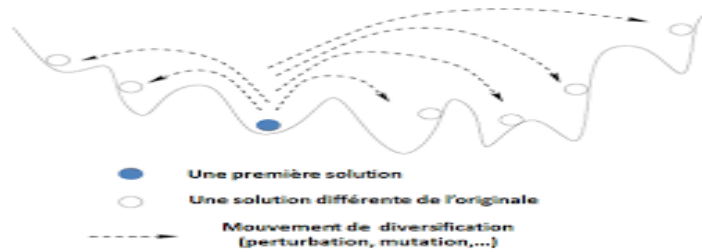
Les métaheuristicques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global (c'est-à-dire l'extremum global d'une fonction), par échantillonnage d'une fonction objectif. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution (d'une manière proche des algorithmes d'approximation).

Il existe un grand nombre de métaheuristicques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes utilisent cependant un haut niveau d'abstraction, leur permettant d'être adaptées à une large gamme de problèmes différents.

Concepts fondamentaux des métaheuristiques : Les métaheuristiques ne nécessitent pas une connaissance particulière sur les problèmes d'optimisation à résoudre. Il suffit d'associer une ou plusieurs variables à une ou plusieurs solutions (optimum). Il existe deux points critiques pour toute métaheuristique :

**Diversification :**

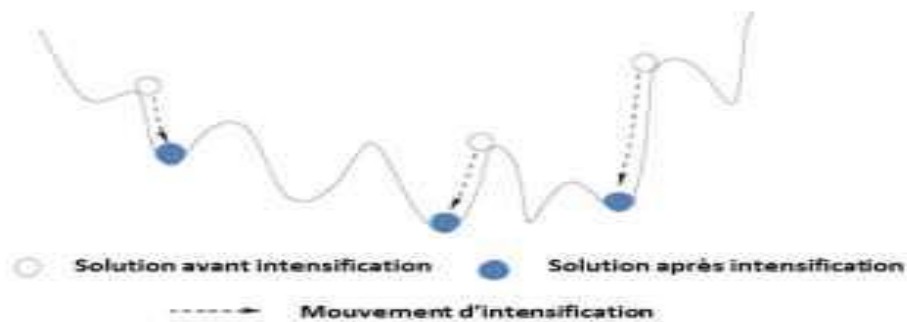
Le principe de diversification d'une méthode d'optimisation donnée correspond à sa capacité de parcourir aisément l'espace de recherche pour obtenir des solutions très différentes les unes des autres. Autrement dit, c'est un mécanisme pour une exploration assez large de l'espace de recherche.



**Figure 2. 3** Processus de diversification d'une solution

**Intensification :**

Autant le principe de diversification essaye de déplacer les solutions dans d'autres zones de l'espace de recherche, autant le processus d'intensification vise à forcer une solution donnée à tendre vers l'optimum local de la zone à laquelle elle est attachée. En effet, Elle permet une exploitation de l'information accumulée durant la recherche.



**Figure 2. 4** Processus d'intensification de plusieurs solutions

## 2.4. Classification des métaheuristiques :

Une manière de classer les métaheuristiques est de distinguer celles qui travaillent avec une population de solutions de celles qui ne manipulent qu'une seule solution à la fois (méthodes de trajectoire). Voici une figure qui montre cette classification avec des exemples typiques pour chaque catégorie

### 2.4.1 Métaheuristiques à solution unique :

Les méthodes itératives à solution unique sont toutes basées sur un algorithme de recherche de voisinage qui commence avec une solution initiale, puis l'améliore pas à pas en choisissant une nouvelle solution dans son voisinage

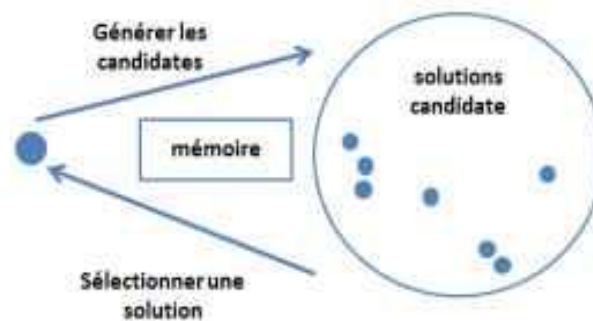


Figure 2. 5 Principe des métaheuristiques à solution unique

Il existe plusieurs méthodes pour cette classification, les plus connues sont :

#### 1) La recherche locale:

La recherche locale itérative est un modèle de recherche locale qui améliore le principe de recherche locale multi départ (« multiple start local search ») dans lequel des méthodes de descente sont lancées successivement sur des solutions initiales générées aléatoirement pour contrer l'aspect déterministe de la descente. Le principe est simple, une recherche par descente est appliquée sur une solution initiale pour générer une meilleure solution. On applique ensuite une nouvelle recherche par descente sur cette nouvelle solution après l'avoir « perturbée ». La

solution obtenue est comparée avec la solution initiale pour savoir si elle la remplace ou non. Tout cela représente une itération de la recherche locale itérative exemple :

- Le problème du voyageur de commerce où une solution est un cycle contenant tous les nœuds du graphe et l'objectif est de minimiser la longueur totale du cycle
- Le problème SAT où une solution est une association et l'objectif est de maximiser le nombre de clauses vérifiées par l'association dans ce cas, la solution finale est celle qui satisfait toutes les clauses.

## 2) La recherche taboue:

La méthode de recherche tabou, ou simplement méthode taboue, a été formalisée en 1986 par F. Glover [17]. Sa principale particularité tient dans la mise en Suivre de mécanismes inspirés de la mémoire humaine. L'idée consiste à garder la trace du cheminement passé dans une mémoire et de s'y référer pour guider la recherche.

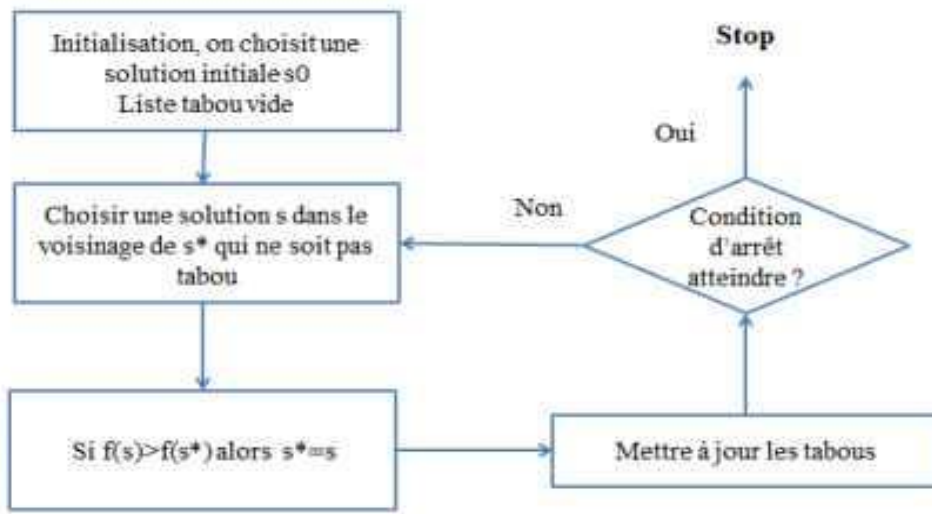
- Notion de base:  
À chaque itération, l'algorithme tabou choisit le meilleur voisin non tabou, même si celui-ci dégrade la fonction de coût. Pour cette raison, on dit de la recherche avec tabou qu'elle est une méthode agressive.
- La liste tabou (mémoire):  
L'idée de base de la liste taboue consiste à mémoriser les configurations ou régions visitées et à introduire des mécanismes permettant d'interdire à la recherche de retourner trop rapidement vers ces configurations

**Algorithme :** L'algorithme de recherche tabou peut être résumé comme suit :

```

engendrer une configuration (solution) initiale  $s_0$  de  $S \leftarrow S_0$ 
Répéter
  Voisinage  $s'$  de  $s$  n'appartient pas à la liste tabou
  Calculer  $\Delta E = f(s) - f(s')$  /*  $f$  fonction objective à maximiser
  Si  $\Delta E < 0$  alors
     $s \leftarrow s'$ 
  Fsi
  mettre à jour la liste tabou
Jusqu'à (critère d'arrêt)
Retourner( $S$ )
    
```

**Pseudo-code :** La figure suivante illustre la structure générale de la méthode recherche tabou :



**Figure 2. 6:** Pseudo-code de la recherche tabou

### Domaines d'application :

La méthode de recherche tabou est utilisée pour résoudre des problèmes complexes et/ou de très grande taille (NP-complet). Vu l'importance de cette technique de recherche, il existe plusieurs domaines qui s'intéressent à celui-ci. Parmi eux, on peut citer les suivants:

- Problèmes de transport.
- Planification et ordonnancement.
- Optimisation de la production et gestion des inventaires.
- Optimisation de graphes.
- Logique et intelligence artificielle.
- Télécommunications.
- Optimisation de structures.
- ...etc

### Avantages et Inconvénients:

La méthode de recherche tabou fournit plusieurs avantages mais elle souffre aussi de beaucoup d'inconvénients. Les avantages et les inconvénients les plus importants sont les suivants :

**Avantages** : On peut résumer les avantages de la technique de recherche tabou dans deux points principaux qui sont : Grande efficacité et Fonctionnement simple à comprendre.

**Inconvénients** : Il existe plusieurs inconvénients pour la méthode RT, parmi eux : Paramètres peu intuitifs, demande en ressources importantes si la liste des tabous est trop imposante.

### 3) Le recuit simulé:

La méthode du recuit simulé est une généralisation de la méthode Monte-Carlo ; son but est de trouver une solution optimale pour un problème donné. Elle a été mise au point par trois chercheurs de la société IBM : S. Kirk Patrick, C.D. Gelatt et M.P. Vecchi en 1983, et indépendamment par V. Cerny en 1985 à partir de l'algorithme de Metropolis ; qui permet de décrire l'évolution d'un système thermodynamique [18].

#### Principe de base :

La méthode de recuit simulé ou « simulated annealing » est un algorithme d'optimisation souvent utilisé lorsque le calcul de la solution optimale exacte demanderait un temps de calcul trop important. Historiquement, cette technique tire son nom et son inspiration de pratiques issues de la thermodynamique et plus particulièrement, de la façon dont les métaux sont chauffés puis refroidis. Ce processus utilisé en métallurgie pour améliorer la qualité d'un solide cherche un état d'énergie minimale qui correspond à une structure stable du solide. En partant d'une haute température à laquelle le solide est devenu liquide, la phase de refroidissement conduit la matière liquide à retrouver sa forme solide par une diminution progressive de la température. Chaque température est maintenue jusqu'à ce que la matière trouve un équilibre thermodynamique. Quand la température tend vers zéro, seules les transitions d'un état à un état d'énergie plus faible sont possibles.

Metropolis et al. furent les premiers à implémenter, dès 1953, ce type de principe dans le calcul numérique. Ils utilisent une méthode stochastique pour générer une suite d'états successifs du système en partant d'un état initial donné. Tout nouvel état est obtenu en faisant subir un déplacement (une perturbation) aléatoire à un atome quelconque. Soit  $\Delta E$  la différence d'énergie occasionnée par une telle perturbation. Le nouvel état est accepté si l'énergie du système diminue ( $\Delta E \leq 0$ ). Sinon, il est accepté avec une probabilité définie par :  $p(\Delta E, T) = e^{-\Delta E/(Cb \cdot T)}$  où T

est la température du système et  $C_b$  une constante physique connue sous le nom de constante de Boltzmann.

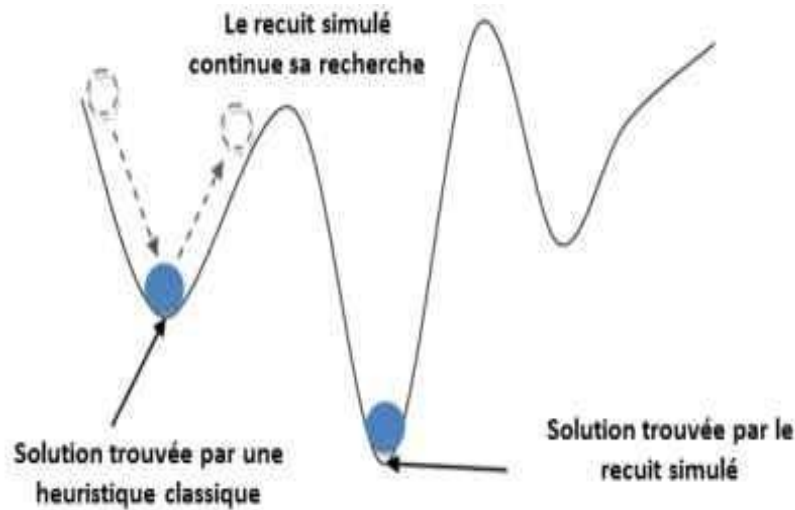


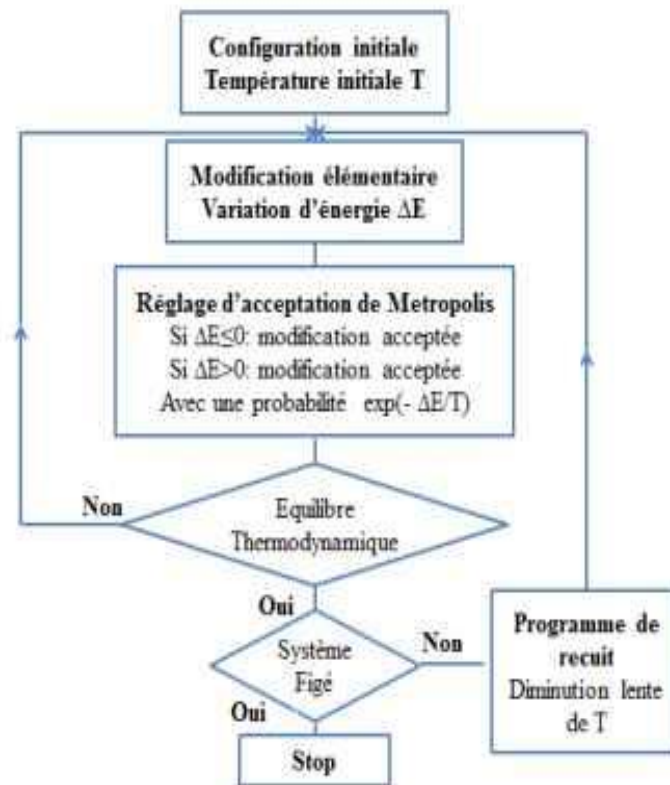
Figure 2. 7 : Principe de recuit simulé

Algorithme :

```

engendrer une configuration (solution) initiale  $s_0$  de  $s$ :  $S \leftarrow S_0$ 
Initialiser la température  $T$ 
Répéter
  Voisinage  $s'$  de  $s$ 
  Calculer  $\Delta E = f(s) - f(s')$  /*  $f$  fonction objective à maximiser
  Si  $\Delta E < 0$  alors
     $s \leftarrow s'$ 
  Sinon
    tirer  $u \in [0, 1]$ 
    si  $u < \exp(-\Delta E/T)$  alors  $s \leftarrow s'$ 
  Fsi
  Réduire la température  $T$  ( $T \leftarrow \alpha T$ ) ( $0.5 < \alpha < 0.9999$ )
Jusqu'à (critère d'arrêt)
Retourner( $S$ )
    
```

**Pseudo-code** : La figure suivante donne un pseudocode du recuit simulé :



**Figure 2. 8:** pseudo-code du recuit simulé

### Domaines d'application :

la méthode du recuit simulé peut être utilisée pour résoudre plusieurs problèmes d'optimisation. En effet, par rapport aux autres algorithmes d'optimisation globale, il est perçu que cette méthode est plus facile à mettre en œuvre et

Moins encline à être piégée dans les optimums locaux. Parmi

Les domaines d'applications les plus connus on peut citer :

- La conception des circuits intégrés (problème de placement et de répartition).
- Le routage des paquets dans les réseaux.
- La segmentation d'images.
- Le problème du voyageur de commerce.

- Le problème du sac à dos.

### **Avantages et Inconvénients :**

Comme pour toute métaheuristique, le recuit simulé offre plusieurs avantages, et au même temps souffre de plusieurs défauts. Dans cette section on cite quelques avantages et inconvénients de cette méthode d'optimisation combinatoire.

Les principaux avantages de l'algorithme de recuit simulé (SA) peuvent être résumés comme suit:

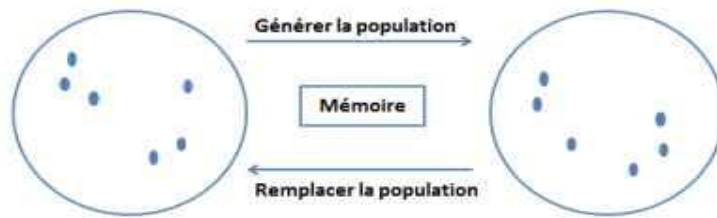
- il peut traiter des systèmes arbitraires et des fonctions objectives, en termes de modèles hautement non linéaires et multimodaux, données bruitées ou soumises à de fortes contraintes.
- il converge vers un optimum global lorsque le nombre d'itérations tend vers l'infini (assez grand).
- il fournit des bonnes solutions pour la majorité des problèmes d'optimisation.
- il est relativement facile à mettre en œuvre et moins sensible à la taille de problème.
- c'est une technique d'optimisation généralisée, car elle ne repose pas sur toutes les propriétés restrictives du modèle.

Il existe plusieurs défauts, parmi eux on peut citer les suivants :

- Nombre important de paramètres (température initiale, taux de décroissance de la température, durée des paliers de la température, critère d'arrêt du programme).
- Réglage souvent empirique des paramètres.
- Temps de calcul excessif dans certaines application (problème de performance).
- L'impossibilité de savoir si la solution trouvée est optimale.

### **2.4.2 Métaheuristique à populations :**

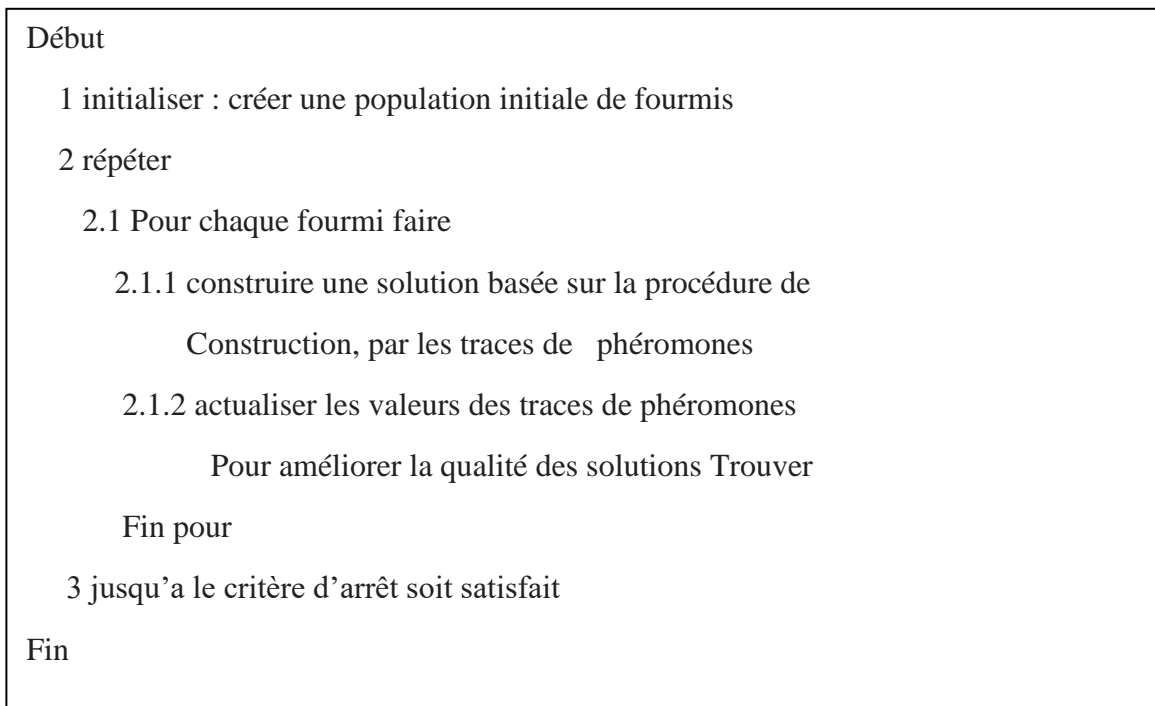
Dans cette classe les métaheuristicues utilisent la notion de population : elles manipulent toutes un échantillonnage de la fonction objectif, via des processus communs. Autrement dit, les méthodes d'optimisation à population de solutions améliorent, au fur et à mesure des itérations, une population de solutions. L'intérêt de ces méthodes est d'utiliser la population comme facteur de diversité



**Figure 2. 9:** Principe des Métaheuristique à populations

**Les colonies des fourmis :** L'algorithme de colonies de fourmis a été à l'origine principalement utilisé pour produire des solutions quasi optimales au problème du voyageur de commerce, puis, plus généralement, aux problèmes d'optimisation combinatoire. On observe, depuis ses débuts, que son emploi se généralise à plusieurs domaines, depuis l'optimisation continue jusqu'à la classification, ou encore le traitement d'image.

Algorithme générale de colonies de fourmis est présenté comme suit [20] :



## 2.5. Les Algorithmes Génétiques (AG) :

Les algorithmes génétiques sont inspirés de la théorie de l'évolution et des processus biologiques qui permettent à des organismes de s'adapter à leur environnement. Ils ont été inventés dans le milieu des années 60 (Holland, 1962 ; Rechenberg, 1965 ; Fogel et al, 1966) [20].

La sélection naturelle, que Darwin appelle l'élément "propulseur" de l'évolution, favorise les individus d'une population qui sont le mieux adaptés à un environnement. La sélection est suivie de croisements et de mutations au niveau des individus, constitué d'un ensemble de gènes. Ainsi deux individus "parents", qui se croisent, transmettent une partie de leur patrimoine génétique à leurs descendants. L'individu enfant fait que celui-ci est plus au. Au fur et à mesure des générations son sélectionne les individus les mieux adaptés, et l'augmentation du nombre des individus bien adaptés fait évoluer la population entière [21].

La mise en œuvre d'un algorithme génétique : est réalisée suivant les étapes suivantes [22] :

1. Création d'une population initiale.
2. Evaluation des individus de la population.
3. Sélection des meilleurs individus.
4. Reproduction (Croisement et mutation).
5. Formation d'une nouvelle génération.

## 2.6 Les opérateurs des algorithmes génétiques :

### 2.6.1 Population initiale :

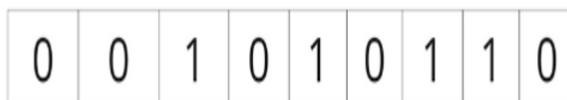
L'AG démarre avec une population composée de  $N$  individus dans le codage retenu. Le choix des individus conditionne fortement la rapidité de l'algorithme. Si la position de l'optimum dans l'espace de recherche est totalement inconnue, il est intéressant que la population soit répartie sur tout l'espace de recherche. Si par contre des informations à priori sur le problème sont disponibles, il paraît évident de générer les individus dans un espace particulier afin d'accélérer la convergence. Disposant d'une population initiale souvent non homogène, la diversité de la population doit être entretenue aux cours des générations afin d'explorer le plus largement possible l'espace de recherche. C'est le rôle des opérateurs de croisement et de mutation.

## 2.6.2 Le codage :

Le codage est une modélisation d'une solution d'un problème donné sous forme d'une séquence de caractères appelée chromosome où chaque caractère, dit aussi gène, représente une variable ou une partie du problème. La tâche principale consiste à choisir le contenu des gènes qui facilite la description du problème et respecte ses contraintes [22]. 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  $i$  [14].

### 2.6.2.1 Codage binaire :

Le codage binaire consiste à utiliser des bits (0 ou 1) pour représenter les différentes solutions. Dans le sens où les gènes sont représentés par des bits et les chromosomes sont représentés par des chaînes de bits. Le type de ce codage s'adapte bien aux problèmes de type binaire, comme le problème max SAT ou les problèmes du sac à dos. Prenons l'exemple du problème du sac à dos dans sa forme la plus simple (unidimensionnelle). Supposant que nous devons sélectionner un sous ensemble d'objets parmi un ensemble de 10 objets. Les solutions peuvent être facilement représentées par des chaînes binaires de taille 10, où chaque bit représente l'état d'un objet : le bit 1 pour dire que l'objet est sélectionné, le bit 0 pour dire que l'objet est écarté. La figure suivante représente une des solutions du problème tel que les objets 2, 3, 5, 8, 9, et 10 sont sélectionnés. Tandis que les autres (i.e. 1, 4, 6 et 7) sont écartés [14].



### 2.6.2.2 Le codage à caractère :

Il s'agit d'utiliser une suite de caractères différents pour représenter le chromosome, comme le montre la figure suivante :



### 2.6.2.3 Le codage arborescent :

Ce type de codage utilise une représentation arborescente des individus avec un ensemble de nœuds, où l'ensemble des nœuds représente un individu [14]. Comme le montre la figure suivante :

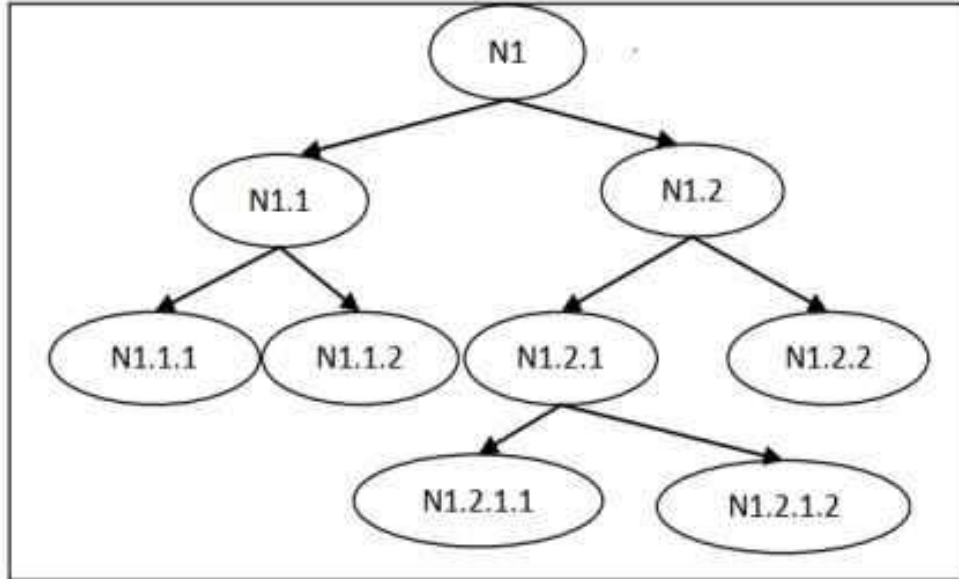


Figure 2. 10: Principe le codage arborescent

### 2.6.3 L'évaluation :

L'évaluation de l'adaptation de chaque individu à l'environnement est réalisée au moyen d'une fonction d'adaptation (fitness). Cette fonction attribue à chaque individu une valeur qui représente son niveau d'adaptation. La fonction d'adaptation peut affecter directement la qualité des résultats obtenus par l'AG ainsi que le temps d'exécution. Elle est fonction du critère à minimiser. En raison de l'analogie avec la théorie de l'évolution (survie des individus les mieux adaptés à leur environnement), l'algorithme génétique est naturellement formulé en termes de maximisation. Etant donné une fonction  $h$  réelle à une ou plusieurs variables [17], le problème d'optimisation sur l'espace de recherche  $S$  s'écrit de la manière suivante :

$$\max_{x \in S} h(x) \quad (1)$$

Dans beaucoup de problèmes, l'objectif est exprimé sous forme de minimisation d'une fonction coût  $f$

$$\max_{x \in S} f(x) \quad (2)$$

Le passage du problème de minimisation à un problème de maximisation est obtenu par transformation de la fonction  $f$  selon la relation suivante :

$$h(x) = \frac{1}{1+f(x)} \quad (3)$$

#### 2.6.4 La sélection :

L'opérateur de sélection est chargé de "favoriser" les meilleurs individus. Plus formellement, l'opérateur de sélection va générer à partir de la population courante une nouvelle population par copie des individus choisis de la population courante. La copie des chaînes s'effectue en fonction des valeurs de la fonction d'adaptation. Ce procédé permet de donner aux meilleures chaînes, une probabilité élevée de contribuer à la génération suivante. Cet opérateur est bien entendu une version artificielle de la sélection naturelle, la survie darwinienne des chaînes les plus adaptées. Plusieurs stratégies sont possibles pour effectuer une telle sélection parmi lesquelles nous abordons [17].

- a) **La sélection par classement** : Elle consiste à ranger les individus de la population dans un ordre croissant (ou décroissant selon l'objectif) et à retenir un nombre fixé de génotypes. Ainsi, seuls les individus les plus forts sont conservés. L'inconvénient majeur de cette méthode est la convergence prématurée de l'algorithme génétique [17].
- b) **La sélection par roulette** : Selon cette méthode, chaque chromosome est copié dans la nouvelle population proportionnellement à son fitness. On effectue en quelque sorte, autant de tirages avec remise que d'éléments existant dans la population [16].

Ainsi pour un chromosome particulier  $ch_i$  de fitness  $f(ch_i)$ , la probabilité de sa sélection dans la nouvelle population de taille  $N$  est :

$$P(ch_i) = \frac{f(ch_i)}{\sum_{j=1}^N f(ch_j)} \quad (4)$$

- c) **La sélection par tournois** : Cette méthode est celle avec laquelle on tire avec remise de deux individus de  $P$  ( $P$  : l'ensemble de la population), et on les fait "combattre". Celui qui a le fitness la plus élevée l'emporte avec une probabilité  $p$  ( $p$  : probabilité comprise entre 0.5 et 1).

On répète ce processus n fois de manière à obtenir les n individus de la population P 0 qui serviront de parents [23].

La variance de cette méthode est élevée et le fait d'augmenter ou de diminuer la valeur de la p permet respectivement de diminuer ou d'augmenter la pression de la sélection.

### 2.6.5 Le croisement :

La naissance d'un nouvel individu, nécessite la prise aléatoire d'une partie des gènes de chacun des deux parents. Ce phénomène, issu de la nature est appelé croisement (crossover). Il s'agit d'un processus essentiel pour explorer l'espace des solutions possibles. Une fois la sélection terminée, les individus sont aléatoirement répartis en couples. Les chromosomes parents sont alors copiés et recombinaison afin de produire chacun deux descendants ayant des caractéristiques issues des deux parents. Dans le but de garder quelques individus parents dans la prochaine population, on associe à l'algorithme génétique une probabilité de croisement, qui permet de décider si les parents seront croisés entre eux ou s'ils seront tout simplement recopiés dans la population suivante. Il existe plusieurs types de croisement parmi lesquels on trouve : le croisement en 1 point, le croisement en deux points et le croisement en N points, ces types sont résumés dans la figure suivante :

- a) **Croisement en un point** : Pour chaque couple, on choisit au hasard un point de croisement (Figure 2. 11). Le croisement s'effectue directement au niveau binaire, et non au niveau des gènes. Un croisement peut être coupé au milieu d'un gène.



Figure 2. 11: Principe Croisement en un point

- b) **Croisement en deux points** : Ont choisi au hasard deux points de croisements successifs. Cet opérateur est généralement considéré comme plus efficace que le précédent

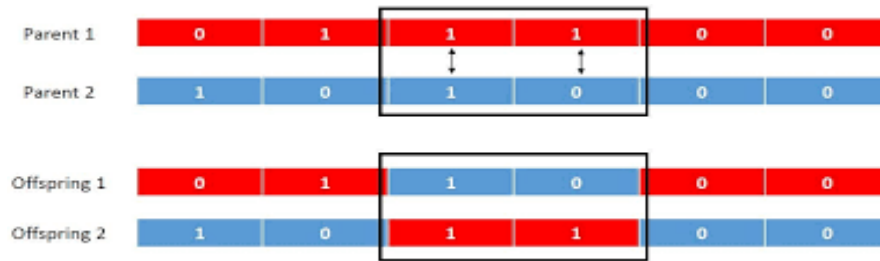


Figure 2. 12: Principe Croisement en deux points

### 2.6.6 La mutation

La mutation est un changement aléatoire selon une certaine règle probabiliste qui doit faire sur les génotypes, avec une faible probabilité  $P_m$  (fixée par l'utilisateur) de la valeur d'un ou plusieurs allèles d'un chromosome. En général, la mutation ne permet pas l'obtention de meilleures solutions, mais elle permet de garder une diversité dans l'évolution des individus et d'éviter les optimums locaux, et se protège contre une perte irrécouvrable dans les caractéristiques des individus. La mutation classique consiste à transformer dans un chromosome binaire un 1 en un 0 ou le contraire

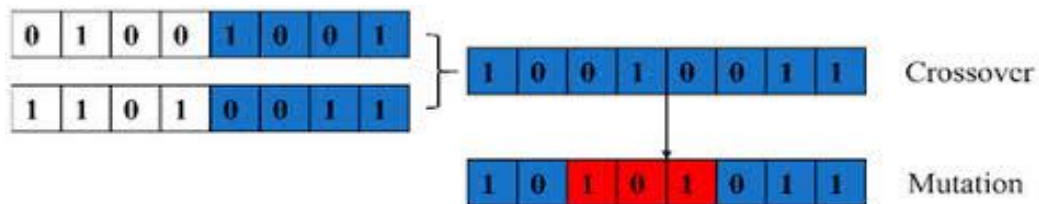


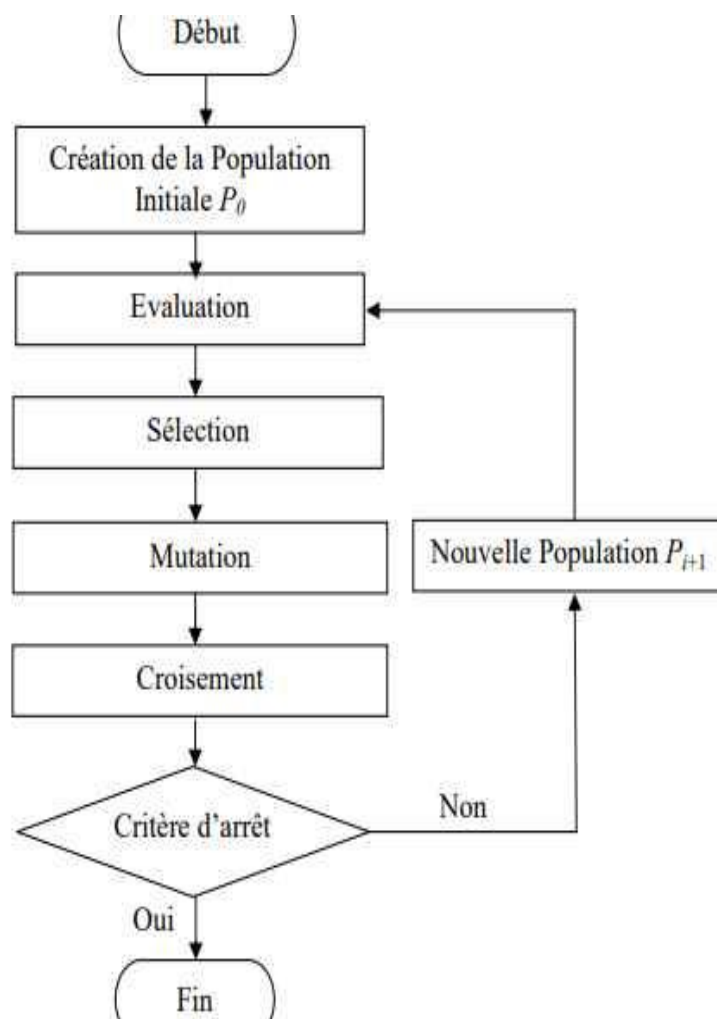
Figure 2. 13 :Principe de mutation

### 2.7 Fonctionnement d'un Algorithme Génétique:

Un Algorithme Génétique (AG) est un processus itératif de recherche exploratoire dont les étapes sont les suivantes :

1. Initialisation de la population initiale  $P_0$  (population de génération 0).
2. Évaluation des chromosomes de la population  $P_i$
3. Sélection des meilleurs chromosomes de  $P_i$  qui vont constituer les « procréateurs » de la génération  $P_{i+1}$ .

4. Croisement deux-à-deux des chromosomes sélectionnés pour générer de nouveaux individus, les « descendants ».
5. Mutation opérée aléatoirement sur un chromosome également choisi au hasard pour apporter de la diversité dans le patrimoine génétique de la population.
6. Remplacement et génération de la nouvelle population  $P_{i+1}$ .
7. Répéter le processus à partir de 2 tant que le critère d'arrêt n'est pas satisfait (i.e. génération N terminale non atteinte, solution optimale non trouvée...)



**Figure 2. 14:** les étapes fonctionnement d'un Algorithme Génétique

## **Conclusion :**

A travers ce chapitre, nous avons abordé les définitions essentielles à la compréhension de travail du point de vue optimisation. Nous avons pu voir qu'un grand nombre de méthodes d'optimisation existe pour résoudre un problème combinatoire. Ces méthodes peuvent être exactes ou bien approchées où on distingue deux grandes familles : les heuristiques et les métaheuristiques qui sont détaillées dans la deuxième partie de ce chapitre. Enfin, nous avons présenté deux techniques largement utilisées pour résoudre les problèmes d'optimisation qui sont le recuit simulé et les algorithmes génétiques. Alors, on a donné pour chacune de ces méthodes sa définition, son algorithme, ses domaines d'application et nous avons terminé par ses avantages et inconvénients. Les algorithmes génétiques seront utilisés dans ce travail pour optimiser l'emplacement des caméras de surveillance dans le but d'augmenter leur performance.

## **Chapitre 3**

### **Proposition d'une méthode de résolution d'emplacement des caméras par les algorithmes génétiques**

### 3.1 Introduction :

L'importance croissante des applications de surveillance dans la recherche se fait sentir de plus en plus. Ces applications peuvent concerner divers domaines tels que la surveillance d'une usine de production, d'une zone à des fins de sécurité, ou de produits industriels, entre autres. Les réseaux de capteurs visuels constituent le fondement de ces systèmes de surveillance. Le positionnement optimal des capteurs visuels (caméras) revêt une importance cruciale car ces systèmes doivent assurer une couverture maximale des zones sensibles tout en minimisant les coûts et en garantissant une qualité de service optimale. La qualité des images dépend étroitement de la position et de l'orientation des caméras. Bien que les besoins en champ de vision puissent varier selon les applications spécifiques, toutes les applications basées sur la vision exigent une disposition des caméras permettant d'obtenir une qualité d'image acceptable.

La principale motivation de ce travail est d'améliorer le placement hors ligne des caméras pour les applications de surveillance. Ce placement dépend de plusieurs facteurs tels que la localisation potentielle des caméras, la présence d'obstacles dans les zones sensibles, et la priorité attribuée à chaque zone. Ainsi, le problème de placement devient un défi d'optimisation avec des contraintes interdépendantes et concurrentes. Étant donné qu'il n'existe pas de solution algorithmique efficace pour les problèmes d'optimisation discrets et contraints, nous avons recours à un algorithme évolutionnaire. Dans ce chapitre, nous présentons un outil de conception pour le placement des caméras de surveillance. Cet outil, basé sur un algorithme génétique, est à la fois simple et efficace. En utilisant cet outil, il est possible de placer les caméras de manière à couvrir au mieux les multiples zones sensibles définies par l'utilisateur. L'outil détermine la position et l'orientation optimales des caméras PTZ (panoramique, inclinaison, zoom) pour assurer une couverture optimale de la zone définie par l'utilisateur.

### 3.2 État de l'art:

- L'étude de Fu et al. (2014) explore l'optimisation du placement des caméras dans un réseau afin de maximiser la couverture tout en respectant diverses contraintes, tandis que Malik et Bajcsy (2008) examinent le positionnement des caméras stéréo 3D pour améliorer la résolution spatiale et réduire les erreurs de reconstruction. L'objectif principal est d'optimiser le placement de quatre à dix caméras dans un système de capture de

mouvement (MoCap) pour suivre le mouvement virtuel d'un objet, en garantissant que chaque réflecteur sur l'objet soit visible par au moins trois caméras à tout moment. Les avantages de la méthode PSO (Optimisation par Essaims Particulaires) comprennent sa rapidité et son faible nombre de paramètres à ajuster. Les contributions clés de cette étude incluent l'optimisation des positions et des orientations des caméras en utilisant les coordonnées 2D du mouvement virtuel, la garantie que chaque réflecteur est vu par au moins trois caméras à chaque étape du mouvement, l'évaluation de trois variantes de PSO (SPSO, WPSO, CPSO) en termes de récupération des réflecteurs, de minimisation des erreurs et de stabilité de la fonction de fitness, ainsi que la comparaison des performances de convergence des techniques, montrant que SPSO et WPSO nécessitent plusieurs tests pour une paramétrisation adéquate, tandis que CPSO n'en a pas besoin. En résumé, cette étude propose des techniques d'optimisation pour le placement des caméras dans un système MoCap afin de suivre efficacement le mouvement virtuel, avec des résultats prometteurs en termes de récupération des réflecteurs et de minimisation des erreurs.[24]

- La méthode de placement des caméras proposée par Gupta (2012) appartient à la catégorie CPFC (Placement de Caméras pour Environnements Entièrement Fermés), dans laquelle la forme de la zone de surveillance, le modèle d'imagerie et les obstacles sont pris en compte. Le nombre de caméras utilisées dans le système de surveillance vidéo est minimisé lorsque les caméras couvrent toute la zone de surveillance, et l'algorithme génétique est appliqué pour optimiser les paramètres du placement des caméras. L'algorithme génétique est également utilisé pour optimiser le placement des caméras dans d'autres études. Contrairement à la méthode de [26], la méthode proposée par Van (2009) prend en compte la structure tridimensionnelle de la zone de surveillance, ce qui peut être utilisé dans un environnement complexe. La méthode de placement des caméras proposée par [25] garantit non seulement que toute la zone de surveillance est couverte par les caméras, mais utilise également la qualité de surveillance dans le champ de vision de la caméra pour optimiser le placement des caméras. Dans cette méthode, la qualité de surveillance dans l'axe optique de la caméra obéit à une distribution de Rayleigh, et dans la direction horizontale obéit à une distribution gaussienne. Cela permet aux utilisateurs d'obtenir des vidéos de haute qualité, et convient aux systèmes de surveillance nécessitant des informations détaillées, tels que les systèmes de reconnaissance faciale. La méthode

de placement des caméras dans [27] prend également en compte la qualité de surveillance, dans laquelle le modèle de détection anisotrope de la caméra a été établi en fonction de la résolution de la caméra, de la délocalisation et de la distorsion géométrique. Ce modèle est utilisé pour évaluer la qualité de surveillance des différentes zones dans la couverture des caméras. Dans l'ensemble, dans la méthode de placement des caméras pour les environnements entièrement fermés, il est souvent nécessaire de modéliser la zone de surveillance, et de tenir compte de l'influence des obstacles sur ces caméras et de la qualité de surveillance des différentes zones.

- Challinger (2008) a souligné l'importance pour les techniques de sécurité d'être capables de délimiter les zones à risque et d'établir des normes pour minimiser les incidents indésirables. Malgré les efforts visant à améliorer l'efficacité des systèmes de surveillance pour détecter les mouvements de personnes à l'intérieur des bâtiments, aucun système ne peut garantir une couverture complète. Bigdelil et al. (2007) ont évalué l'efficacité des systèmes de surveillance sur les quais de gare, se concentrant sur la détection et l'identification robustes des individus dans des environnements encombrés. Yabuta et Kitazawa (2008) ont introduit un algorithme pour optimiser le placement des caméras en utilisant une approche 2D pour évaluer leur couverture, mais cette approche présente des limitations en termes de couverture précise, notamment dans les zones masquées derrière des obstacles. OPTICOM (2013) a identifié les vibrations comme un facteur influençant la durée de vie des caméras de surveillance, soulignant ainsi l'importance d'un placement adéquat pour éviter les effets néfastes. Chen et al. (2013) ont noté que la présence de rideaux coupe-feu dans la zone de surveillance peut altérer les performances des caméras, tandis que MoogVideolarm (2012) a souligné que les sources de chaleur telles que les radiateurs peuvent également affecter les caméras de surveillance. [28]

### **3.3 Méthode proposée :**

#### **3.3.1 Algorithme Génétique :**

Un algorithme génétique en informatique est une méthode de résolution de problèmes inspirée par le processus de sélection naturelle et de génétique. Il s'agit d'une technique d'optimisation qui évolue à partir d'une population de solutions potentielles à un problème donné.

L'algorithme génétique fonctionne généralement de la manière suivante :

- 1) **Initialisation** : Une population initiale de solutions potentielles est générée de manière aléatoire.
- 2) **Évaluation** : Chaque solution de la population est évaluée en fonction de son aptitude à résoudre le problème donné.
- 3) **Sélection** : Les individus les mieux adaptés (selon une fonction de fitness) sont sélectionnés pour la reproduction.
- 4) **Croisement (Crossover)** : Les solutions sélectionnées sont croisées pour créer de nouvelles solutions (descendants). Cela peut impliquer l'échange de parties de leurs informations génétiques.
- 5) **Mutation** : De manière aléatoire, certaines des nouvelles solutions peuvent subir des modifications mineures (mutations).
- 6) **Remplacement** : Les nouvelles solutions remplacent les solutions les moins performantes de la population précédente.
- 7) **Répéter** : Les étapes 2 à 6 sont répétées jusqu'à ce qu'une solution satisfaisante soit trouvée ou qu'un certain critère d'arrêt soit atteint (par exemple, un nombre maximal d'itérations).

Les algorithmes génétiques sont souvent utilisés pour résoudre des problèmes d'optimisation difficile où les techniques traditionnelles peuvent être inefficaces. Ils ont été appliqués avec succès dans de nombreux domaines, tels que l'ingénierie, la robotique, la bio-informatique et la finance, l'optimisation de l'emplacement des caméras de surveillance, et autres.

Les algorithmes génétiques sont couramment utilisés pour résoudre des problèmes complexes d'optimisation, y compris l'optimisation de l'emplacement des caméras de surveillance. Voici un aperçu détaillé de leur application dans ce domaine

### 3.3.2 Application dans l'Optimisation d'Emplacement des Caméras de Surveillance :

**Objectif** : Le but est de déterminer l'emplacement optimal des caméras de surveillance pour maximiser la couverture d'une zone donnée tout en minimisant le nombre de caméras utilisées.

- **Représentation** : Chaque individu dans la population représente une configuration possible d'emplacements de caméras. Cela peut être représenté par un vecteur binaire où chaque bit indique la présence ou l'absence d'une caméra à un emplacement spécifique.
- **Fonction de Fitness** : La fonction de fitness doit évaluer la qualité de la couverture fournie par une configuration particulière. Elle peut être définie par :
  - La surface totale couverte par les caméras.
  - La redondance de la couverture (zones surveillées par plusieurs caméras).
  - Les zones critiques couvertes.

Une fonction de fitness typique pourrait être :

$$Fitness = \sqrt{(X2 - X1)^2 + (Y2 - Y1)^2 + (Z2 - Z1)^2} \quad (5)$$

#### ❖ Sélection et Opérateurs Génétiques

- **Sélection** : Des méthodes comme la roulette ou le tournoi peuvent être utilisées pour sélectionner les individus pour le croisement.
- **Croisement** : Des opérateurs de croisement, tels que le croisement à un point ou à deux points, peuvent être utilisés pour combiner les caractéristiques des parents.
- **Mutation** : Des mutations aléatoires peuvent être appliquées pour introduire de nouvelles caractéristiques dans la population.

### 3.3.3 L'utilisation des algorithmes génétiques pour l'optimisation des emplacements des caméras de surveillance:

**Initialisation de la Population** : La première étape consiste à créer une population initiale. Chaque individu de cette population représente une solution possible au problème, c'est-à-dire un ensemble d'emplacements de caméras. Ces emplacements peuvent être codés de différentes manières, mais une représentation binaire est souvent utilisée pour sa simplicité.

#### Exemple :

Une zone de surveillance est divisée en une grille de  $N \times N$ .

Un vecteur binaire de longueur  $N \times N$  représente les emplacements des caméras, où chaque bit correspond à une cellule de la grille (1 indique la présence d'une caméra, 0 son absence).

### ❖ **Évaluation et Fonction de Fitness**

Chaque individu est évalué selon une fonction de fitness qui mesure la qualité de la solution proposée. La fonction de fitness peut inclure plusieurs critères, tels que :

- **Couverture totale** : La proportion de la zone surveillée par les caméras. La redondance de la couverture est définie par la mesure dans laquelle certaines parties de la zone sont surveillées par plusieurs caméras.
- **Coût** : Le nombre total de caméras utilisées.

**Exemple de fonction de fitness :**

$$Fitness = \sqrt{(X2 - X1)^2 + (Y2 - Y1)^2 + (Z2 - Z1)^2} \quad (6)$$

### ❖ **Sélection**

La sélection vise à choisir les individus les plus aptes pour reproduire et générer la prochaine génération. Deux méthodes couramment utilisées sont :

**Tournement Sélection** : Un sous-ensemble d'individus est choisi aléatoirement, et le meilleur individu de ce sous-ensemble est sélectionné.

- **Croisement** : Le croisement combine les caractéristiques de deux parents pour produire de nouveaux individus. Le **Croisement à un point** est un point de croisement choisi aléatoirement, et les segments des deux parents sont échangés pour créer deux enfants.

#### **3.3.4 Les étapes d'algorithme génétiques:**

- 1) Nous commençons par sélectionner une population initiale aléatoire de taille (N) à partir de l'espace de recherche, en veillant à ce qu'elle se situe dans la région réalisable. Chaque individu de cette population est encodé à l'aide de la méthode spécifiée.
- 2) La valeur de forme physique de chaque individu de la population est ensuite calculée à l'aide de matrices représentant la couverture des points prioritaires et non prioritaires. Cela nous permet de comparer et de déterminer l'optimalité de chaque solution.
- 3) Ensuite, nous utilisons la méthode de sélection par tournoi pour choisir une population de "bons" réseaux. Les deux meilleurs individus sont conservés, et le processus passe à l'étape de reproduction.

- 4) Pendant la reproduction, la population sélectionnée subit une recombinaison à travers les opérations suivantes :
  - a. Le croisement est effectué avec une probabilité de 0,8 en utilisant la fonction de croisement éparpillé.
  - b. Pour maintenir la diversité, une mutation est appliquée avec une probabilité de 0,1.
- 5) Ces opérations aboutissent à une nouvelle population, qui remplace celle existante.
- 6) Les étapes 2 à 4 sont répétées de manière itérative jusqu'à ce que le critère d'optimisation se stabilise.

## 3.4 Langage et environnement

### 3.4.1 Python :

Python est un langage de programmation de haut niveau interprété, orienté objet et doté d'une sémantique dynamique.

Ses structures de données intégrées de haut niveau, combinées au typage et à la liaison dynamique, le rendent très attrayant pour le développement rapide d'applications, ainsi que pour une utilisation comme langage de script ou de collage pour connecter des composants existants entre eux.

La syntaxe simple et facile à apprendre de Python met l'accent sur la lisibilité et réduit donc le coût de maintenance du programme.

Python prend en charge les modules et les packages, ce qui encourage la modularité des programmes et la réutilisation du code.

L'interpréteur Python et la vaste bibliothèque standard sont disponibles gratuitement sous forme source ou binaire pour toutes les principales plates-formes et peuvent être distribués gratuitement.

Souvent, les programmeurs tombent amoureux de Python en raison de la productivité accrue qu'il offre. Puisqu'il n'y a pas d'étape de compilation, le cycle édition-test-débogage est incroyablement rapide.

Le débogage des programmes Python est simple : un bug ou une mauvaise entrée ne provoquera jamais d'erreur de segmentation.

Au lieu de cela, lorsque l'interpréteur découvre une erreur, il lève une exception. Lorsque le programme ne détecte pas l'exception, l'interpréteur imprime une trace de pile.

Un débogueur au niveau source permet d'inspecter les variables locales et globales, d'évaluer des expressions arbitraires, de définir des points d'arrêt, de parcourir le code ligne par ligne, etc.

Le débogueur est écrit en Python lui-même, témoignant du pouvoir introspectif de Python. D'un autre côté, le moyen le plus rapide de déboguer un programme consiste souvent à ajouter quelques instructions d'impression à la source : le cycle rapide édition-test-débogage rend cette approche simple très efficace.

### **Cas d'utilisation de Python:**

- Création d'applications web sur un serveur
- Construction de flux de travail pouvant être utilisés en conjonction avec des logiciels
- Connexion à des systèmes de bases de données Lecture et modification de fichiers
- Réalisation de calculs mathématiques complexes
- Traitement des big data
- Prototypage rapide
- Développement de logiciels prêts pour la production

### **Avantages du langage de programmation Python :**

- Les points suivants mentionnent les avantages du langage Python :
- Il est développé sur la base d'une licence open-source approuvée par l'OSI. Il est complètement gratuit et peut également être utilisé à des fins commerciales. Le langage Python peut être librement redistribué ainsi que modifié.
- Python est livré avec une bibliothèque standard étendue que les développeurs peuvent utiliser. Il existe de multiples bibliothèques en Python disponibles pour les tests unitaires, les bases de données, la manipulation d'images, les expressions et d'autres fonctionnalités. Ces bibliothèques standard robustes réduisent les efforts des programmeurs pour réécrire le code à chaque fois. Contrairement à d'autres langages de programmation, il n'est pas nécessaire de réécrire des codes en Python.

- Il est utile pour développer des applications web, des applications de bureau, des applications scientifiques et numériques complexes.
- Python peut exprimer des tâches complexes en utilisant seulement quelques lignes de code.
- C'est un langage portable car le même code peut être réutilisé sur différentes machines sans avoir besoin de faire des modifications. Les codes écrits pour Mac peuvent fonctionner sur Linux ou Windows sans changements. Cela permet d'économiser le temps et les efforts de réécrire le même programme plusieurs fois.
- Python est un langage de haut niveau avec une syntaxe simple, la disponibilité de riches bibliothèques et une grande polyvalence.
- Il est utile pour développer des applications de bureau, des applications web, ainsi que des applications scientifiques et numériques complexes.



**Figure 3. 1:** Logo Python

### 3.4.2 ANACONDA

Anaconda est une distribution open-source des langages de programmation Python et R pour la science des données qui vise à simplifier la gestion des packages et le déploiement. Les versions des packages dans Anaconda sont gérées par le système de gestion des packages, conda, qui analyse l'environnement actuel avant d'exécuter une installation afin d'éviter de perturber d'autres Framework et packages.

La distribution Anaconda est livrée avec plus de 250 packages installés automatiquement. Plus de 7500 packages open-source supplémentaires peuvent être installés à partir de PyPI ainsi que du gestionnaire de packages et d'environnements virtuels conda. Elle inclut également une interface graphique (GUI), Anaconda Navigator, comme alternative graphique à l'interface en

ligne de commande. Anaconda Navigator est inclus dans la distribution Anaconda et permet aux utilisateurs de lancer des applications et de gérer les packages, environnements et canaux conda sans utiliser les commandes en ligne de commande. Navigator peut rechercher des packages, les installer dans un environnement, exécuter les packages et les mettre à jour.

#### 3.4.2.1 Pourquoi utiliser Conda sur les clusters de GRICAD ?

Conda vous permet d'avoir accès aux principaux frameworks en IA : tensorflow, tensorflow-gpu, keras, torch, torchvision, pycuda.....



Figure 3. 2 : Logo ANACONDA

#### 3.4.3. Espace de travail :

**PyCharm** : PyCharm est un environnement de développement intégré (IDE) utilisé pour programmer en Python. Il offre une analyse de code, un débogueur graphique, un testeur unitaire intégré, une intégration avec les systèmes de contrôle de version, et prend en charge le développement web avec Django. PyCharm est développé par la société tchèque JetBrains.

Il est multiplateforme, fonctionnant sous Microsoft Windows, macOS et Linux. PyCharm propose une édition professionnelle, publiée sous une licence propriétaire, et une édition communautaire publiée sous la licence Apache. PyCharm Community Edition est moins étendue que l'édition professionnelle. [32]

#### Fonctionnalités :

- Assistance et analyse du code, avec autocomplétions, surlignage de la syntaxe et des erreurs, intégration de l'inter, et corrections rapides

- Navigation dans les projets et le code : vues spécialisées des projets, vues de la structure des fichiers et navigation rapide entre les fichiers, classes, méthodes et utilisations
- Refactorisation du code Python : y compris renommer, extraire méthode, introduire variable, introduire constante, remonter, descendre et autres
- Support pour les Framework web: Django, web2py et Flask
- Débogueur Python intégré
- Tests unitaires intégrés, avec couverture ligne par ligne
- Développement Python pour Google App Engine
- Intégration du contrôle de version : interface utilisateur unifiée pour Mercurial, Git, Subversion, Perforce et CVS avec listes de modifications et fusion
- Intégration des outils scientifiques : s'intègre avec IPython Notebook, possède une console Python interactive, et supporte Anaconda ainsi que plusieurs packagent scientifiques, y compris Matplotlib et NumPy.



**Figure 3.3 :** Logo Pycharm

## **Conclusion :**

En conclusion de ce chapitre, nous pouvons constater l'importance significative des applications de surveillance dans divers domaines, ainsi que l'importance croissante d'améliorer la distribution et l'emplacement des caméras dans ces applications. En s'appuyant sur des technologies telles que les algorithmes génétiques, il est possible d'améliorer considérablement la distribution des caméras pour obtenir une couverture efficace et offrir une qualité d'image élevée.

De nombreuses études et techniques utilisées dans le domaine de l'amélioration de la distribution des caméras ont été examinées, allant de l'utilisation des algorithmes d'auto-développement aux modèles analytiques des zones surveillées. L'importance de facteurs tels que la qualité de l'image, la réduction des coûts et la garantie d'une couverture complète a également été soulignée.

Il est clair que l'utilisation des technologies d'intelligence artificielle telles que les algorithmes génétiques peut contribuer de manière significative au développement des systèmes de surveillance modernes, ce qui reflète l'évolution rapide de ce domaine et le besoin croissant de solutions de surveillance et de sécurité efficaces et innovantes.

## **Chapitre 4**

### **Implémentation de la méthode et résultats**

## 4.1 Introduction :

Ce chapitre vise la description et l'application d'algorithme génétique à cas d'étude pour optimiser le placement des caméras dans le système de surveillance. L'objectif principal est d'analyser en profondeur les résultats obtenus, de les comparer aux approches précédentes et d'en discuter les implications. Sur la base de cette analyse, des recommandations seront formulées pour améliorer l'efficacité des systèmes de surveillance et de sécurité.

## 4.2 Implémentation :

### 4.2.1 Cas d'étude :

#### Algérie Télécom :

En Algérie, Algérie Télécom, le principal fournisseur de services de télécommunications, a mis en place un réseau étendu de caméras de surveillance dans ses installations clés, telles que les centres de données et les bureaux régionaux. Le positionnement stratégique de ces caméras a été crucial pour assurer la sécurité des infrastructures critiques et la protection des équipements sensibles contre le vol, le vandalisme ou les intrusions. Dans certains cas, les caméras ont permis d'identifier et de dissuader les tentatives d'intrusion ou de sabotage avant qu'elles ne causent des dommages. De plus, les enregistrements vidéo ont été utilisés comme preuves dans des enquêtes sur des incidents de sécurité et ont aidé à renforcer les mesures de sécurité globales de l'entreprise.

### 4.2.2 Explication du code:

```
import random
```

Le module random en Python définit une série de fonctions pour générer ou manipuler des entiers aléatoires. L'instruction `import random` charge le module random.

```
def generate_camera_placement(num_cameras, x_limit, y_limit, z_limit):  
    return [(random.uniform(a: 0, x_limit), random.uniform(a: 0, y_limit), random.uniform(a: 0, z_limit)) for _ in range(num_cameras)]
```

La fonction `generate_camera_placement` génère les positions de placement des caméras de manière aléatoire dans les limites spécifiées pour les axes x, y et z. L'instruction `import random` charge le module `random`.

```
def fitness_function(camera_placement):  
  
    total_distance = 0  
    num_pairs = 0  
    for i in range(len(camera_placement)):  
        for j in range(i + 1, len(camera_placement)):  
            dist = ((camera_placement[i][0] - camera_placement[j][0]) ** 2 +  
                   (camera_placement[i][1] - camera_placement[j][1]) ** 2 +  
                   (camera_placement[i][2] - camera_placement[j][2]) ** 2) ** 0.5  
            total_distance += dist  
            num_pairs += 1  
    return total_distance / num_pairs if num_pairs > 0 else 0
```

La fonction `fitness_function` calcule la distance moyenne entre toutes les paires de caméras dans une liste de positions de caméras. Si aucune paire n'est présente, la fonction retourne 0.

```
def crossover(parent1, parent2):  
    child = []  
    for p1, p2 in zip(parent1, parent2):  
        child.append(random.choice([p1, p2]))  
    return child
```

La fonction `crossover` prend les gènes des parents et produit un nouvel enfant en mélangeant aléatoirement les gènes des parents à chaque position génétique.

```
def mutate(camera_placement, mutation_rate, x_limit, y_limit, z_limit):
    mutated_placement = []
    for camera in camera_placement:
        if random.random() < mutation_rate:
            mutated_camera = (min(max(camera[0] + random.uniform(-1, 1), 0), x_limit),
                               min(max(camera[1] + random.uniform(-1, 1), 0), y_limit),
                               min(max(camera[2] + random.uniform(-1, 1), 0), z_limit))
            mutated_placement.append(mutated_camera)
        else:
            mutated_placement.append(camera)
    return mutated_placement
```

#### La fonction prend cinq arguments :

- Camera placement : une liste contenant les positions actuelles des caméras, chaque position étant représentée par un triplet (x, y, z).
- Mutation rate : le taux de mutation, c'est-à-dire la probabilité de modifier aléatoirement chaque position de caméra.
- x\_limit, y\_limit, z\_limit : les limites supérieures des coordonnées x, y et z respectivement.

#### Le fonctionnement de la fonction est le suivant :

- Une liste vide, mutated\_placement, est créée pour stocker les positions des caméras après mutation.

#### Pour chaque position de caméra dans la liste camera placement :

- Un nombre aléatoire entre 0 et 1 est généré à l'aide de la fonction random.random(). Si ce nombre est inférieur au mutation rate, une mutation est appliquée à la position de la caméra.
- Les coordonnées x, y et z de la caméra sont modifiées en ajoutant un nombre aléatoire compris entre -1 et 1 en utilisant random.uniform(-1, 1).

- Les nouvelles coordonnées sont ensuite ajustées pour rester dans les limites spécifiées. Les fonctions min et max sont utilisées pour garantir que les coordonnées ne dépassent pas les limites supérieures et ne sont pas inférieures à zéro.
- Si une mutation est appliquée, la position mutée est ajoutée à la liste mutated\_placement. Sinon, la position originale est ajoutée.

Enfin, la liste mutated\_placement, contenant les positions mutées ou originales, est retournée.

```
def genetic_algorithm(population_size, num_cameras, generations, crossover_rate, mutation_rate, x_limit, y_limit, z_limit):
    # توليد المجتمع الابتدائي
    population = [generate_camera_placement(num_cameras, x_limit, y_limit, z_limit) for _ in range(population_size)]

    for _ in range(generations):
        # تقييم المجتمع
        fitness_scores = [fitness_function(camera_placement) for camera_placement in population]

        # اختيار الأفراد الأفضل
        selected_parents = [population[i] for i in
                            sorted(range(len(fitness_scores)), key=lambda k: fitness_scores[k], reverse=True)[:int(0.5 * population_size)]]

        # إنشاء الجيل الجديد
        new_population = []
        while len(new_population) < population_size:
            parent1, parent2 = random.choices(selected_parents, k=2)
            if random.random() < crossover_rate:
                child = crossover(parent1, parent2)
                new_population.append(mutate(child, mutation_rate, x_limit, y_limit, z_limit))
            else:
                new_population.append(mutate(parent1, mutation_rate, x_limit, y_limit, z_limit))

        population = new_population

    # اختيار الحل الأفضل
    best_solution = min(population, key=fitness_function)
    best_fitness = fitness_function(best_solution)
    return best_solution, best_fitness
```

Ce code est une implémentation d'un algorithme génétique.

- La fonction genetic\_algorithm prend en paramètres la taille de la population initiale, le nombre de caméras, le nombre de générations, les taux de crossover et de mutation, ainsi que les limites sur les coordonnées x, y et z.
- La population initiale est générée avec des placements de caméras aléatoires à l'intérieur des limites spécifiées.

- Pour un nombre donné de générations, la population est évaluée en utilisant une fonction de fitness qui évalue chaque placement de caméra.
- Les meilleurs individus sont sélectionnés pour devenir les parents de la nouvelle génération, en fonction de leurs scores de fitness.
- Une nouvelle population est créée en utilisant des opérations de crossover et de mutation, où les individus sélectionnés sont utilisés comme parents pour créer de nouveaux individus.
- La meilleure solution est sélectionnée à partir de la dernière population en utilisant la fonction de fitness, et son fitness est renvoyée.

```
# مدخلات المستخدم
num_cameras = int(input("أدخل عدد الكاميرات: "))
x_limit = float(input("أدخل الحد الأقصى لمحور X: "))
y_limit = float(input("أدخل الحد الأقصى لمحور Y: "))
z_limit = float(input("أدخل الحد الأقصى لمحور Z: "))
population_size = 100
generations = 100
crossover_rate = 0.8
mutation_rate = 0.1
```

- `num_cameras = int(input("Entrez le nombre de caméras : "))`: Cette ligne demande à l'utilisateur le nombre de caméras qu'il souhaite utiliser dans le programme. L'entrée utilisateur sera un nombre entier (integer).
- `x_limit = float(input("Entrez la limite maximale pour l'axe x : "))`: Cette ligne demande à l'utilisateur la limite maximale pour l'axe X. L'entrée utilisateur sera un nombre décimal (float), déterminant la plage maximale des valeurs que l'axe X peut prendre.
- `y_limit = float(input("Entrez la limite maximale pour l'axe y : "))`: Cette ligne demande à l'utilisateur la limite maximale pour l'axe Y de la même manière que précédemment.
- `z_limit = float(input("Entrez la limite maximale pour l'axe z : "))`: Cette ligne demande à l'utilisateur la limite maximale pour l'axe Z de la même manière que précédemment.
- `Population size = 100` : Définit une valeur par défaut pour la taille de la population, dans ce cas 100.

- Générations = 100 : Définit une valeur par défaut pour le nombre de générations que l'algorithme va générer, dans ce cas 100.
- crossover\_rate = 0.8: Définit une valeur par défaut pour le taux de croisement, c'est-à-dire le taux d'échange génétique entre les individus des générations, dans ce cas 0.8.
- Mutation rate = 0.1: Définit une valeur par défaut pour le taux de mutation, c'est-à-dire le taux de survenance de changements aléatoires dans les gènes, dans ce cas 0.1.

```
# تشغيل الخوارزمية الجينية
best_solution, best_fitness = genetic_algorithm(population_size, num_cameras, generations, crossover_rate, mutation_rate, x_limit, y_limit, z_limit)
```

Cette ligne de code exécute l'algorithme génétique et enregistre la meilleure solution ainsi que le meilleur ajustement obtenu lors de l'exécution de l'algorithme.

- Genetic\_algorithm () : Cette fonction représente l'exécution de l'algorithme génétique. Elle prend un ensemble d'entrées telles que la taille de la population, le nombre de caméras, le nombre de générations, le taux de croisement, le taux de mutation, et les limites des axes (X, Y, Z).
- Population size : La taille de la population à utiliser dans l'algorithme.
- Num\_cameras : Le nombre de caméras.
- Générations : Le nombre de générations que l'algorithme va créer.
- Crossover\_rate : Le taux de croisement, ou la probabilité de mélanger les gènes des individus dans la population.
- Mutation rate : Le taux de mutation, ou la probabilité qu'un changement aléatoire se produise dans les gènes des individus.
- X\_limit, y\_limit, z\_limit : Les limites supérieures des axes X, Y et Z respectivement.

Après l'exécution de l'algorithme, les meilleures solutions et le meilleur ajustement sont enregistrés dans les variables best solution et best fitness. Ces variables contiennent respectivement la solution optimale trouvée et la valeur de l'ajustement qui lui est associée

```
# عرض النتيجة
print("أفضل مواقع للكاميرات:", best_solution)
print("قيمة اللياقة لأفضل حل:", best_fitness)
```

Ces lignes utilisent la fonction print () pour afficher les résultats de la solution optimale trouvée par l'algorithme génétique.

La première ligne print ("Meilleurs emplacements de caméra :", best solution) imprime un texte commençant par "Meilleurs emplacements de caméra :" suivi de la valeur de la variable best solution. Cela signifie que les positions des caméras sélectionnées comme meilleure solution seront imprimées.

La deuxième ligne print ("Valeur du fitness pour la meilleure solution :", best fitness) imprime un texte commençant par "Valeur du fitness pour la meilleure solution :" suivi de la valeur de la variable best fitness. Cela signifie que la valeur des fitness pour la meilleure solution sélectionnée sera imprimée.

### 4.2.3 Les Résultats:

RDC:

Numéro de caméra	Anciennes coordonnées	Coordonnées trouve par l'étude	Amélioration
1	x=38.30/y=0/z=3	x=38.30 /y=0/z=3	x=0/y=0/z=0
2	X=4.60/y=10.15/z=3	x=4.60/y=10.15/z=3	x=0/y=0/z=0
3	x=10.80/y=2.5/z=3	x=10.80/y=2.5/z=3	x=0/y=0/z=0
4	x=10.80/y=2.10/z=3	x=10.80/y=2.10/z=2.7 0	<b>x=0/y=0/z=0.30</b>
5	x=21.50/y=2.10/z=2.5	x=21.50/y=2.50/z=2.5	x=0/y=0.40/z=0
6	x=4.60/y=13.75/z=2.5	x=4.93/y=13.75/z=2.5	<b>x=0.33/y=0/z=0</b>

7	$x=21.70/y=4.40/z=2.5$	$x=21.94/y=4.70/z=2.5$	<b><math>x=0.24/y=0.30/z=0</math></b>
8	$x=21.50/y=13/z=2.5$	$x=19.90/y=13/z=2.5$	<b><math>x=1.60/y=13/z=0</math></b>
9	$x=13.45/y=4.60/z=2.5$	$x=13.85/y=4.90/z=2.5$	<b><math>x=0.40/y=0.30/z=0</math></b>
10	$x=15.30/y=4.70/z=2.5$	$x=15.60/y=4.90/z=2.5$	<b><math>x=0.30/y=0.20/z=0</math></b>
11	$x=4/y=6.50/z=2.5$	$x=4.30/y=6.30/z=2.5$	<b><math>x=0.30/y=0.20/z=0</math></b>
12	$x=4.10/y=5.15/z=2.5$	$x=4.35/y=5/z=2.5$	<b><math>x=0.25/y=0.15/z=0</math></b>
13	$x=0/y=27.95/z=3$	$x=0/y=27.95/z=2.85$	<b><math>x=0/y=0/z=0.15</math></b>
14	$x=27.95/y=0/z=3$	$x=27.95/y=0/z=3$	$x=0/y=0/z=0$
15	$x=4.10/y=0/z=3$	$x=2.23/y=0/z=3$	<b><math>x=0.13/y=0/z=0</math></b>
16	$x=0/y=0/z=3$	$x=0/y=0/z=3$	$x=0/y=0/z=0$
17	$x=0/y=4.25/z=2.5$	$x=0.52/y=4.25/z=2.5$	<b><math>x=0.52/y=0/z=0</math></b>
18	$x=11.10/y=0/z=3$	$x=11.10/y=0/z=2.83$	<b><math>x=0/y=0/z=0.17</math></b>
19	$x=38.30/y=0/z=3$	$x=38.30/y=0/z=2.62$	<b><math>x=0/y=0/z=0.38</math></b>

**Table 4. 1** : Résultat placement des caméras dans RDC





**2ème étage:**

<b>Numéro de caméra</b>	<b>Anciennes coordonnées</b>	<b>Coordonnées trouve par l'étude</b>	<b>Amélioration</b>
<b>1</b>	$x=4.10 /y=4.25/z=2.5$	$x=5.18 /y=4.25/z=2.5$	<b><math>x=1.08/y=0/z=0</math></b>
<b>2</b>	$x=3.25 /y=19.50/z=2.5$	$x=3.84$ $/y=19.75/z=2.5$	<b><math>x=0.59 /y=0.25/z=0</math></b>
<b>3</b>	$x=24.4/y=26.2/z=2.5$	$x=24.19$ $/y=27.01/z=2.5$	<b><math>x=0.21/y=0.41/z=0</math></b>
<b>4</b>	$x=33.22/y=13.6/z=2.5$	$x=33.22$ $/y=14.90/z=2.5$	<b><math>x=0 /y=1.30/z=0</math></b>

**Table 4. 3 : Résultats placement des cameras dans 2etage**

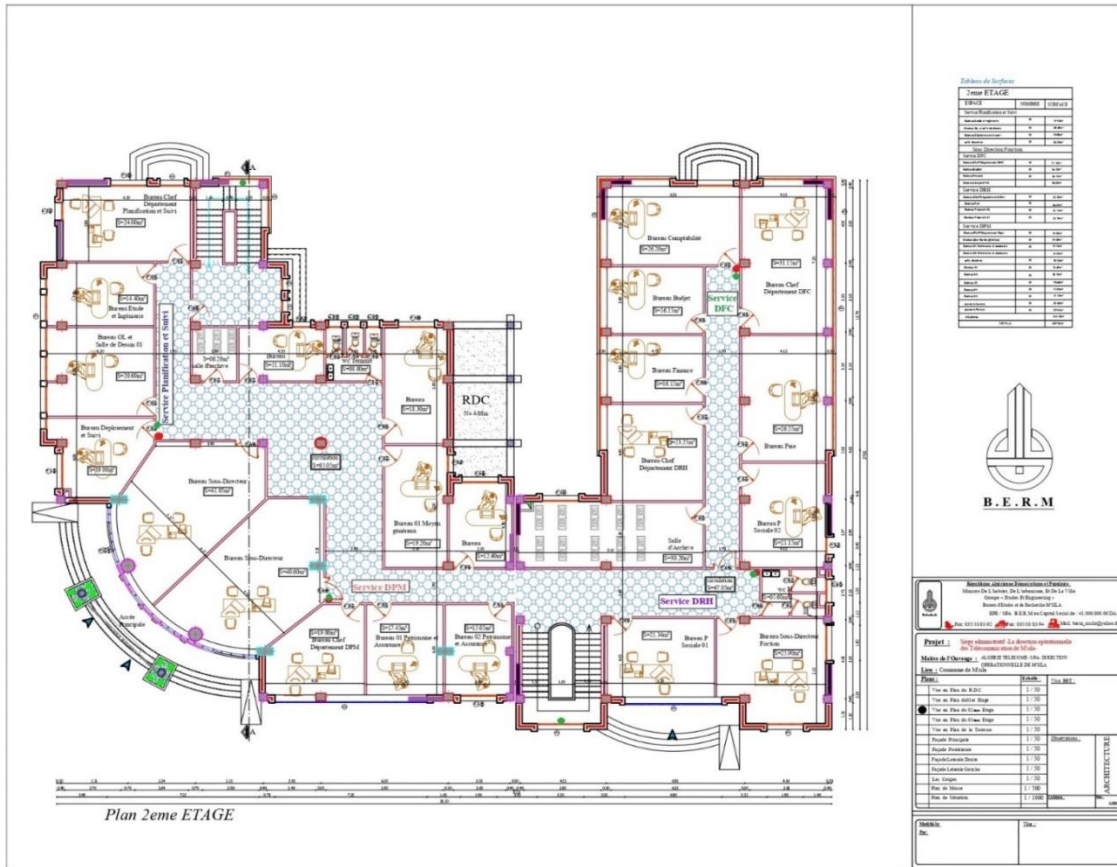


Figure 4.3 : l'emplacement des caméras dans 2etage

3ème étage:

Numéro de caméra	Anciennes coordonnées	Coordonnées trouve par l'étude	Amélioration
1	$x=3 / y=6.80 / z=2.5$	$x=4.25 / y=6.80 / z=2.5$	$x=1.25 / y=0 / z=0$
2	$x=18 / y=6.80 / z=2.5$	$x=21.5 / y=6.80 / z=2.5$	$x=3.5 / y=0 / z=0$
3	$x=22.9 / y=16.20 / z=2.5$	$x=23.52 / y=15.61 / z=2.5$	$x=0.62 / y=0.59 / z=0$
4	$x=26.55 / y=13 / z=2.5$	$x=27 / y=13.48 / z=2.5$	$x=0.45 / y=0.48 / z=0$

Table 4. 4 : Résultat placement des caméras dans 3etage



Grâce à des ajustements continus et à l'adaptation aux changements environnementaux, l'algorithme génétique peut extraire des informations et déterminer les emplacements avec une précision accrue. Cette amélioration notable du processus de placement de la caméra contribue à améliorer la qualité des données collectées et à améliorer les performances globales du système. Par exemple, la caméra numéro 2 au troisième étage présentait un angle d'ombre dans sa position précédente, mais l'algorithme génétique a réussi à améliorer sa position. De même, la caméra numéro 8 au rez-de-chaussée ne voyait pas tous les employés avec précision dans sa position précédente, mais l'algorithme a contribué à améliorer sa position. De plus, certaines zones n'étaient pas sécurisées, comme les cage des escaliers et la salle de conférence dans étage 1 et la salle de conférence et dans l'Accès principale.

Cela souligne l'importance de progrès technologique et du développement d'algorithmes pour fournir des solutions efficaces et avancées dans les domaines de la surveillance et de la sécurité.

## **Conclusion :**

Ce chapitre démontre la faisabilité de l'application d'algorithmes génétiques pour optimiser le positionnement des caméras dans les systèmes de surveillance, conduisant à une amélioration notable de la couverture et de l'efficacité du système. En comparaison avec les approches traditionnelles, les algorithmes génétiques ont prouvé leur efficacité dans l'identification de la disposition optimale des caméras, offrant ainsi des solutions prometteuses pour renforcer la sécurité publique et la sûreté générale.

## **Conclusion générale**

## Conclusion générale

L'emplacement optimal des caméras de surveillance est une problématique cruciale pour assurer la sécurité et la surveillance efficace des espaces publics et privés. Dans ce mémoire, nous avons exploré l'utilisation des algorithmes génétiques comme méthode d'optimisation pour résoudre ce problème complexe. Les algorithmes génétiques, inspirés par le processus de sélection naturelle, offrent une approche robuste pour explorer de vastes espaces de solutions et identifier des configurations optimales pour la disposition des caméras. En simulant des populations de solutions possibles et en appliquant des opérations de mutation, de croisement et de sélection, nous avons pu converger vers des solutions qui maximisent la couverture et minimisent les zones non surveillées tout en tenant compte des contraintes spécifiques du site.

Les principaux résultats obtenus montrent que l'algorithme génétique est capable de maximiser la couverture en trouvant des configurations de caméras qui couvrent le maximum de surface ou de points d'intérêt dans la zone à surveiller, de minimiser les coûts en optimisant non seulement le nombre de caméras nécessaires mais aussi leur placement pour réduire les redondances et les coûts d'installation, et de s'adapter aux contraintes en intégrant facilement diverses contraintes, telles que les obstacles physiques, les zones sensibles à une surveillance accrue, et les préférences spécifiques de surveillance.

Cependant, plusieurs défis et limitations subsistent : la complexité computationnelle, l'algorithme pouvant nécessiter des ressources computationnelles importantes, surtout pour des espaces très vastes ou des problèmes avec de nombreuses contraintes ; la sensibilité aux paramètres, les résultats dépendant des paramètres de l'algorithme génétique (taux de mutation, taille de la population, etc.), nécessitant une calibration minutieuse ; et la scalabilité, bien que l'algorithme soit efficace pour des problèmes de taille modérée, des recherches supplémentaires étant nécessaires pour assurer sa scalabilité à des environnements plus grands et plus complexes. Pour l'avenir, il serait intéressant d'explorer l'amélioration des heuristiques, développer des heuristiques plus avancées pour guider l'algorithme génétique de manière plus efficace ; l'hybridation avec d'autres méthodes, combiner les algorithmes génétiques avec d'autres techniques d'optimisation, telles que les algorithmes de colonies de fourmis ou les algorithmes de recuit simulé, pour améliorer les performances ; et l'application à d'autres domaines, tester cette

approche sur des problèmes similaires dans d'autres domaines, comme l'optimisation de réseaux de capteurs ou la planification de la couverture dans des environnements dynamiques.

En conclusion, ce mémoire démontre que les algorithmes génétiques constituent une méthode prometteuse pour l'optimisation de l'emplacement des caméras de surveillance. Grâce à leur capacité à gérer la complexité et à fournir des solutions optimales dans des conditions variées, ils représentent une avancée significative dans le domaine de la surveillance et de la sécurité. Toutefois, pour maximiser leur potentiel, des recherches continues et des améliorations méthodologiques sont nécessaires.

## Bibliographie :

- [1] <https://www.mesures.com/guides-d-achat/les-cameras-haute-vitesse-et-haute-definition/>
- [2] <https://laas.hal.science/tel-02079969/document>
- [3] <https://www.claravision.com/produit/troublepad-camera-rapide-pour-les-chaines-de-production/>
- [4] <https://theses.hal.science/tel-00004898/document>
- [5] <https://www.mesures.com/guides-d-achat/les-cameras-haute-vitesse-et-haute-definition/>
- [6] <https://laas.hal.science/tel-02079969/document>
- [7] <https://www.rce-sa.com/les-cameras-de-surveillance-et-la-vie-privee/>
- [8] [https://www.kentfaith.fr/blog/article\\_comment-placer-des-cameras-de-surveillance\\_1597](https://www.kentfaith.fr/blog/article_comment-placer-des-cameras-de-surveillance_1597)
- [9] <https://motilde.com/videosurveillance-des-ville-les-composants-essentiels/>
- [10] <https://www.at-electronics.lu/pires-endroits-installation-camera>
- [11] <https://www.researchgate.net/publication/269272124>  
Conference Paper · June 2014  
Bandar Seri Iskandar, 31750 Tronoh, Perak Darul Ridzuan, Malaysia
- [12] Abdesslem LAYEB, Utilisation des Approches d'Optimisation Combinatoire pou La Vérification des Applications Temps Réel. Thèse de Doctorat, Université Mentouri de Constantine 2010
- [13] Mostepha, R : Résolution de problèmes d'optimisation combinatoire par systèmes artificiels auto-organisés. Thèse de magister, Université Mentouri de Constantine ,2008.
- [14] A. Gherboudj. Méthodes de résolution de problèmes difficiles académiques. Thèse de doctorat, Université de Constantine2, 2013.
- [15] h. Hacene. Etude comparative des heuristiques d'optimisation combinatoire. Mémoire de master, Université de Biskra, 2015.
- [16] H. Hamichi. Hybridations d'algorithmes métaheuristiques en optimisation globale et leurs applications. Thèse de doctorat en cotutelle, Ecole Mohammadia D'ingénieurs, Université Mohammed V Agdal, 23 Juin 2013.
- [17] Lambert veller sylvain, le chevalier david, quirico tommy « Problème de ramassage dans une ville virtuelle - Algorithme Tabu Search » Université de Bourgogne 2010- 2011.

- [18] Omessaad, H : Contribution au développement de méthodes d'optimisation Stochastiques application à la conception des dispositifs électrotechniques, Thèse de Doctorat, Université De Lille France 2003.
- [19] M. Sevaux. Métaheuristiques : Stratégies Pour l'Optimisation De La Production De Biens Et De Services. Habilitation À diriger des recherches, Université de Valenciennes et du Hainaut-Cambrésis, Juillet 2004.
- [20] Alain Hertz, L'optimisation Combinatoire, École Polytechnique, Canada, 2006.
- [21] Rachid Chelouah, L'optimisation combinatoire, INA Institut d'informatique appliquée, Suisse, 2003.
- [22] M. Assaad HAMIDA. Introduction Aux Methodes De Controle Intelligent. Faculté des Nouvelles Technologies de l'Information et de la Communications, 2014/2015.
- [23] F. G. Radet A. Souquet. Algorithmes Génétiques
- [24] Solving Optimal Camera Placement Problems inIoT Using LH-RPSO XIAOHUI WANG, HAO ZHANG, AND HAORAN GU School of Aeronautics, Beihang University, Beijing 100191, China
- [25] C. Nicola, L. Leonardo. "Camera placement using particle swarm optimization in visual surveillance applications," 2009 16th IEEE International Conference on Image Processing (ICIP). 2009, pp. 3485-3488.
- [26] A. Gupta, K. A. Pati, V. K. Subramanian, "A NSGA-II based approach for camera placement problem in large scale surveillance applicatio". The 4th International Conference on the Intelligent and Advanced Systems (ICIAS), IEEE. 2012, pp. 347-352.
- [27] C. Wang, F. Qi, G. Shi, "Observation quality guaranteed layout of camera networks via sparse representation," Proceedings of the Visual Communications and Image Processing (VCIP), IEEE. 2011, pp. 1-4.
- [28] A NOVEL METHOD FOR CALCULATING CAMERA COVERAGE IN BUILDINGS USING BIM Ameen Hamza Albahri, PhD Candidate Department of Building, Civil and Environmental Engineering, Concordia University

## Résumé :

Dans ce mémoire, nous présentons un système optimisé pour l'implantation de caméras de surveillance. En utilisant des algorithmes génétiques . Notre méthode vise à placer et orienter de manière optimale plusieurs caméras dans une scène 3D, répondant ainsi à des besoins tels que l'amélioration de la précision de la reconstruction 3D, la maximisation de la couverture de surveillance et le suivi multi-points de vue des piétons.

L'algorithme génétique employé se distingue par sa capacité à explorer efficacement un vaste espace de solutions possibles, en combinant des techniques avancées telles que l'ascension à coordonnées de bloc, l'utilisation de fonctions substituts et des zones d'exclusion. Ces techniques permettent de gérer des fonctions objectives complexes et non différentiables, souvent coûteuses en termes de calcul.

Parmi les caractéristiques clés de notre approche, on note la garantie de convergence globale et la facilité de parallélisation, ce qui accélère considérablement le processus d'optimisation. De plus, en exploitant des propriétés spécifiques de la fonction objective, telles que la symétrie, nous améliorons encore l'efficacité de l'algorithme. Enfin, nous discutons du compromis entre l'atteinte de solutions optimales et la réduction des coûts de calcul lors de l'optimisation successive des points de vue.

Ce système offre une solution robuste et efficace pour les applications de surveillance et de reconstruction 3D, assurant une couverture maximale et des performances optimales dans des environnements complexes.

**Mots-clés :** algorithme génétique , un système optimisé , placement optimale des caméras , caméras de surveillance , Couverture de surveillance , Coût de calcul , dimension 3D.

## **Abstract:**

In this thesis, we present an optimized system for the deployment of surveillance cameras using genetic algorithms. Our method aims to optimally place and orient multiple cameras within a 3D scene, addressing needs such as improving the accuracy of 3D reconstruction, maximizing surveillance coverage, and enabling multi-view tracking of pedestrians.

The employed genetic algorithm stands out for its ability to efficiently explore a vast space of possible solutions by combining advanced techniques such as block coordinate ascent, the use of surrogate functions, and exclusion zones. These techniques allow us to handle complex and non-differentiable objective functions, which are often computationally expensive.

Key features of our approach include the guarantee of global convergence and ease of parallelization, which significantly accelerates the optimization process. Additionally, by exploiting specific properties of the objective function, such as symmetry, we further enhance the algorithm's efficiency. Finally, we discuss the trade-off between achieving optimal solutions and reducing computational costs during the successive optimization of viewpoints.

This system offers a robust and efficient solution for surveillance and 3D reconstruction applications, ensuring maximum coverage and optimal performance in complex environments.

**Keywords** : genetic algorithm , an optimized system, optimal placement of cameras , surveillance cameras , surveillance coverage , computational cost , 3D dimension.

## الملخص:

في هذه الأطروحة، نقدم نظامًا محسنًا لزرع كاميرات المراقبة باستخدام الخوارزميات الجينية. تهدف طريقتنا إلى وضع وتوجيه عدة كاميرات بشكل مثالي داخل مشهد ثلاثي الأبعاد، مما يلبي احتياجات مثل تحسين دقة إعادة البناء ثلاثي الأبعاد، وزيادة تغطية المراقبة، وتتبع متعدد النقاط للمشاة.

تتميز الخوارزمية الجينية المستخدمة بقدرتها على استكشاف مساحة كبيرة من الحلول المحتملة بكفاءة، من خلال دمج تقنيات متقدمة مثل الصعود بالإحداثيات الكتلية، واستخدام الدوال البديلة، ومناطق الاستبعاد. تتيح لنا هذه التقنيات التعامل مع الدوال الهدفية المعقدة وغير القابلة للتفاضل، والتي غالبًا ما تكون مكلفة من حيث الحساب.

تشمل الميزات الرئيسية لطريقتنا ضمان التقارب العالمي وسهولة التوازي، مما يسرع بشكل كبير عملية التحسين. بالإضافة إلى ذلك، من خلال استغلال خصائص محددة للدالة الهدفية، مثل التناظر، نحسن كفاءة الخوارزمية بشكل أكبر. وأخيرًا، ناقش الموازنة بين تحقيق حلول مثالية وتقليل تكاليف الحساب أثناء التحسين المتتابع لنقاط المشاهدة.

يقدم هذا النظام حلاً قويًا وفعالاً لتطبيقات المراقبة وإعادة البناء ثلاثي الأبعاد، مما يضمن تغطية قصوى وأداء مثالي في البيئات المعقدة.

**الكلمات المفتاحية:** خوارزمية جينية ، نظام محسن ، وضع مثالي للكاميرات ، كاميرات المراقبة ، تغطية المراقبة، تكلفة الحوسبة ، البعد الثلاثي.