SPIM

Thèse de Doctorat

école doctorale sciences pour l'ingénieur et microtechniques

UNIVERSITÉ DE BOURGOGNE

This is the one

is the one

David Strub

UNIVERSITI TEKNOLOGI PETRONAS

SPIM

Thèse de Doctorat

école doctorale sciences pour l'ingénieur et microtechniques

UNIVERSITÉ DE BOURGOGNE

N° 4 2

THÈSE présentée par

DAVID STRUB

pour obtenir le

Grade de Docteur de
l'Université de Bourgogne

Spécialité : **Informatique**

This is the one
is the one

Unité de Recherche :
Laboratoire Électronique, Informatique et Image

Soutenue publiquement le 17 septembre 2017 devant le Jury composé de :

INCROYABLE HULK	Rapporteur	Professeur à la Desert State University
SUPER MAN	Examinateur	Professeur à l'Université de Smallville
BAT FOFI	Directeur de thèse	Professeur à l'Université de Gotham City

Remerciements

ma famille et plume toufia ✡✌✑✖✔ ✡ ❤

Sommaire

I

Contexte et Problématiques

1

CONTENT/INTRODUCTION/

1.1/ CONTEXT AND MOTIVATION

Ce squelette d'écrit quelques éléments pouvant vous aider pour écrire votre ouvrage de thèse. Un plan typique d'une thèse scientifique est également proposé. ss[**?**] 122* teste cite [**?**]

1.2/ SCOPE AND CHALLENGES

quelle son les point bloquant en qlq ligne

1.3/ CONTRIBUTION

postionement de camera : problem formulation GA results CPPP : decoupe du problem par plan result and solution

1.4/ ORGANISATION

do some things !

2

STATE OF ART

Sommaire

2.1/ CAMERAS POSITIONING

An efficient cameras positioning is a bottleneck in many application, as for example in the video surveillance field [31, 68, 17, 88, 83]. Where an efficient cameras positing is essential to monitor correctly an area. In the following section the question of : What is a good position and orientation for a camera inside the video surveillance network ? What are the objectives of the cameras positioning ? Is investigated and some answer to this questions is proposed by an overview of the literatures, the different formulations and solutions.

2.1.1/ What is an efficient pose in a camera network

The pose of the cameras, is composed by the position inside the area to monitor and the orientations of it. The orientation is also called looking direction.
To evaluate if a camera pose is efficient, it is primordial to identify the objectives. The objective is different from the finality. The objectives are the most important elements to take in consideration in order to place the set of cameras. When the finality is the global application, as for example video surveillance is the finality but the coverage of an area and a target tracking is required to have an efficient surveillance. The coverage is one of the objectives the most interesting and the most common has many problems about cameras positioning (It will be studied in details later).

To have a clever and efficient cameras positioning system different aspects must be studied. To pose efficiently a set of cameras it is useful to know what does it mean efficient for the camera pose. To do that the objective of the cameras network have to be defined clearly.

The objective can vary and depend on the objective the camera will be affected. The positioning of the camera is impacted by the final objective as for example ; the camera will be placed differently for tag detection [87], then for monitoring a vast outside area [38]. In Zhao et al [87] the camera are placed at a fixed elevation (at the height of the torso) with a looking direction almost parallel to the ground, instead to can do the tag detection and localization with no too much deformation of the tag. Otherwise in Li et al [38] an UAV is used to monitor a vast area. The camera looking direction is almost perpendicular to the ground. These 2 articles are focused on having the best coverage as possible of an area but the constraint (camera are mounted on UAV, fixed elevation and others), and the secondary objective (coverage of an area for tag detection, or cover a vast area) give to different formulation and pose estimation.
The following section is focused on the cameras positioning for maximizing the viewing areas. The viewing area or the coverage rate of the area is directly linked to the pose estimation of each camera and their orientation. To have the best coverage is primordial find the best position for each cameras, depending on the constraint and the eventual secondary objectives.
Indeed to maximize the coverage rate by optimizing the cameras position has been developed this past decade, using many different approaches.
His approach is applicable depending on the formulation of the constraints and objectives. In numerous cases the maximization of the coverage is only the first part of the problem, hence the importance of secondary objectives.
The following part is focused on what kind of area is covered what is exactly called coverage and with the secondary objectives associated too.

2.1.2/ First objective : Coverage

The coverage is the main objective but is not the finality. An efficient coverage is an requirement for many application (video surveillance for example)

The focus of this study is to find the best position of each camera in order to maximize the covered area. To do that, it is important to define what is exactly mean ”covered”. Based on the literature the covered area can be varied, depending on the finality.

— Object coverage :

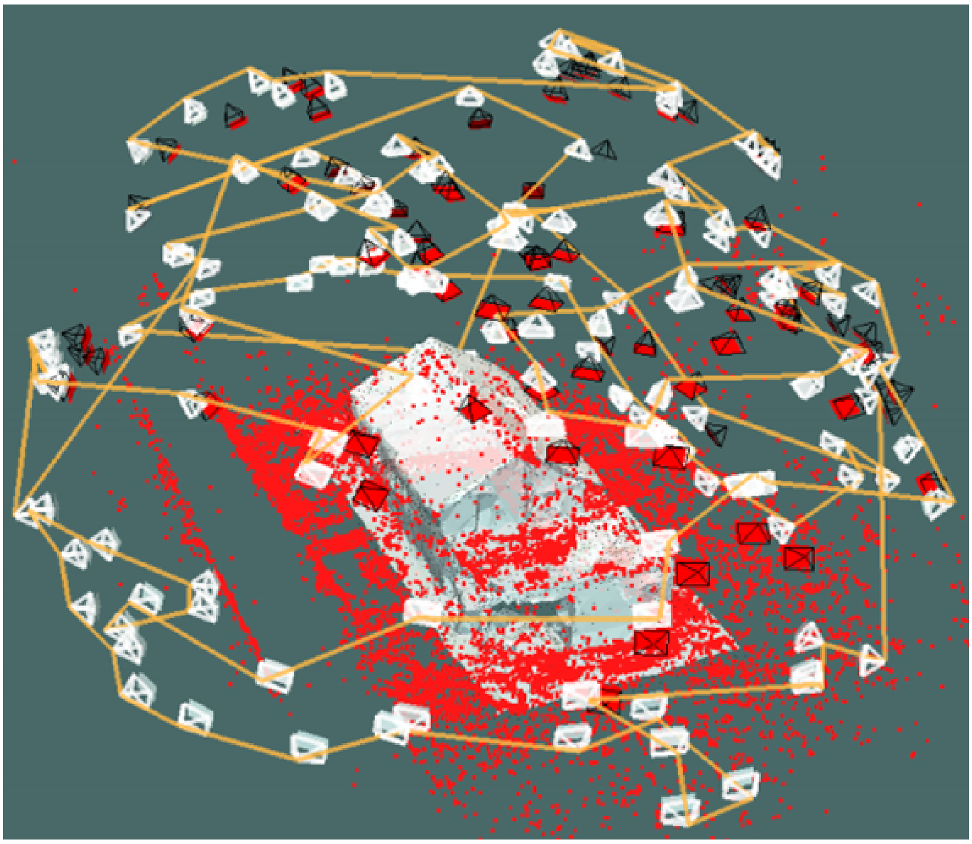


FIGURE 2.1 – Computed camera network pose estimation with the corresponding flight path (yellow) for an 3D object full coverage. This result is from Hoppe et al [33]. .

In Hoppe et al [33] a good coverage is defined by the ability to have full 3D reconstruction of an object (in their 3 dimension) with no occlusion. This definition of the coverage is exploits the prior knowledge of the object to cover. In this case the camera position will made a sphere around the object to cover as in the Figure 2.1. This definition of coverage is not the more helpful due to this restricted application (focused only on 3D reconstruction). This formulation and solution applied is also not applicable for many object in the area.

— Path to cover :
The coverage problem may be reduced as a series of path commonly borrowed by the users (car, pedestrian, . . .). When the area to cover is a well-known place the path of the users can be deduced [5] or if the area to cover is road the trajectory of the driver is knew [41]. In this condition the aim is to cover the common path trajectory of the user as presented in [41, 5, 4, 53]. The path coverage is interesting due to this numerous restriction of the area the coverage can became easier. Otherwise the path coverage introduce an important element is the zone with has a priority. This restricted zone in the area have to be cover in priority or only the path is taking in account in the area to cover.

— Coverage priority :
A natural way of defining the coverage in a context of insufficient number of

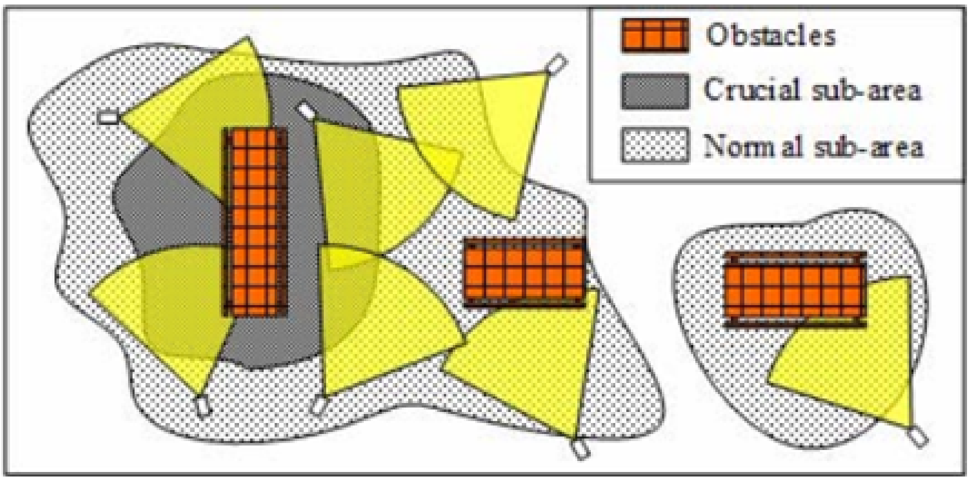


FIGURE 2.2 – Map of an area to cover with crucial sub area (region of interest) the normal sub-area and obstacle. This map is an example of area coverage introduce in Jiang et al [35]. .

camera, is to define in priority some zone of interest to cover. In [83, 35, 34] , their proposed to focus on priority on some predefined region, respectively called region of interest, curial sub-area (see Figure 2.2) and importance space weighting. In the solutions proposed by [83, 35, 34], the camera poses are in priority affected to this specific restricted region and neglected the other part of the area.
Logically if the environment is described with some region of interest some neutral zone must exist. The neutral zone (or normal sub-area) must be cover but their is not the priority. Furthermore some region can be describe as no interest region. In [35, 34] for example, the obstacle are designed as non-interested region and also this region have as consequences to occlude the vision of the camera. The main interest of this design is to keep a maximum of the freedom in the cameras network positioning and see if the camera position can manage with the different local priority and constraints.

— Inside or outside area :
Finally one of the most simplest view of the coverage concern the outside and inside area coverage. The area to cover can be typically a room with walls. Each walls can be a considered as an obstacle and can occlude the camera field of view. In this case the main objective is to cover the surface in totality or at least maximize the coverage. This formulations is also workable for the vast outside area, but in addition it is necessary to take into account the size of the environment and the cameras limitation(as the depth of field), which can make the solution even longer and complex.

The common point in all this coverage definition is the importance to maximize it, despite the other objectives. In the examples presented the coverage was always the first and for some of them the only objective. Despite the interest for maximize the coverage some other element have to be taken in account to have a useful cameras position depending on the finality. The secondary objective can have a not negligible impact on the camera pose. Depending on the camera pose will be greatly affected. The secondary objective are numerous and are closely related then the finality. The most interesting of them are listed :

— The numbers of cameras :
In numerous situation the secondary objective is to limit the number of camera affected to the area as [88, 34, 87]. Limit the number of cameras is primordial to reduce the time computation for the final application and in some case the network traffic. Also reduce the number of camera to cover the full area mean reduce the cost of the video surveillance installation as [12]. In the case minimizing the number of camera is a secondary objective not in contradiction with the camera pose optimization for coverage. The two objective can became complementary. But at some point a to strong trade-off in favour of the minimizing the numbers of cameras may reduce the area cover by accepting some small area to be non covered (also called black hole).

— Object tracking :
The secondary objective can be after the area coverage to detect and localize targets as for example in [17, 68, 39, 82, 65, 87]. In this case the secondary objective is to find a camera pose or dynamic adaptation of the pose for follow one or numerous targets. Keeping a full area covered and at same time the secondary objective is to track efficiently one or more target can became contradictory. The solution must be a trade-off between the coverage maximum coverage and the tracked following like in [17] and [40]. In Liu et al [40] the tracking of a target in a wide area is decomposed in phases, detection and location phase. For the detection phase, the area coverage is important but not any more for the location phase. These phases, although distinct for one target can intervene at the same time if there is more than one. Obviously when the secondary objective is to do tracking the cameras position will be less efficient for cover the area depending on the numbers of targets.

— Luminosity and environmental setup :
Also the control of the image quality in terms of visibility can be an objective. Reddy et al [58] are focus first on coverage of a complex area and in second time localize targets. In order to manage what target must be follow the visibility is taking into account to avoid the too dark area, where the target is not enough visible at all.

— Energetic cost :
To estimate the camera positioning another secondary element can be the energies cost in [40, 9]. Lui et al [40] is a good example, the first objective is to cover most of the area to be able to detect if a target is enter in the area. In second time the target is follow by the smart and autonomous cameras of the network. The set of cameras were randomly dispersed in the area and the coverage of the area will be to select the cameras useful for detect the targets intrusion in the area. The selection of the camera will be to manage between the maximum area overage and the minimum cost consumption. Also the energy cost can be considered not in term of number of cameras turn on, but also in term of distance between each cameras. Notably in the case of each camera position will be a waypoint for doing a path planning [16, 48].

— Multi coverage :
Among the numerous secondary objectives possible the multi coverage is interesting (as for example in [46, 88, 76, 86, 47]). The multi coverage or also called K-coverage where k represent the number of cameras useful to cover some region of the area. The multi coverage or k coverage can be at some point confused with the region of interest and coverage priority. Mostly due to their importance given at some restricted part of area. The difference between them is the impact of the multi coverage on the finale coverage. The multi coverage involved one or few specific zone of the area to be covered by min k cameras at same time in order to be considered as cover. This secondary objective in the case of limit will generate a conflict between the full coverage of the area and the k-coverage requirement even more with a restricted number of cameras.

— Resolution :
The objective of resolution is to keep the quality of the image acceptable [5, 58, 34, 76, 19]. Most of his article are used the distance along the optical axis as a parameters in order to evaluate the resolution constraint. The lens, the sensor and the distance between the camera and a target (or surface) is used to estimate the resolution. In many application is essayer to adapt the distance between the camera and the target or the zoom (when the lens is not a fix focal) then to modifies the other the sensor or the lens.
This secondary constraint affect mostly the system with positioning a set of cameras comprising a zooming lens. If the constraint of resolution is not placed the interest to optimize the camera position for maximum coverage will be to zoom out (or place the camera highest) in order to cover the wider region as possible. When the constraint of resolution is added a trade-off must be done between the zoom out or the camera distance to the target four maximize the coverage, and the zoom in to reduce the area cover for a better resolution.
In [58] the resolution is assimilated to be part of the image quality. In order to keep enough quality on the followed target, the distance of the target is used to keep an acceptable resolution. The problem has been designed by using a Gaussian function in order to define the proper distance between the cameras and the target to keep an acceptable resolution for the application.
Also the depth of view have the same consequence on the camera positioning. Due to the focus point and the aperture of the camera (associate to the resolution), an object to close or to far became blurred. an ideal distance to the between the target and the camera is defined with some boundary as in [24].

The secondary objectives can be numerous and varied, where just few of them has been introduce (the more common and interesting). Among it, some of them are closely related and can be interconnected. The secondary objectives can be associate as in [58] for example where the coverage target, luminosity, the resolution, and obstacle constraint are associate to find the best cameras position with maximize the coverage of the area and the target with good visibility condition.
The real interesting element about the secondary objectives is the impact in the cameras positioning for the global coverage of the area. Then the finality and the problem formulation will have a considerable impact of the obtained solution. Obviously the secondary objective have to be chosen carefully and there are related then the finality. To have an efficient cameras positioning system a trade-off between the different objective and their importance have to be done in order to know what is the priority. In this case the problem

became a multi objective problems.

2.2/ Art gallery problem

The problem of camera positioning is a tricky problem and depending on the finality of the camera networks and the formulation the camera pose will be affected.
Once the objectives defined (see section 2.1) the next step, is to know how to represent the problems. To do that manly 2 different paradigms have to be studied. The first is from the geometrical problem called the Art Gallery Problem (AGP) formulation is commonly and historically borrowed as is presents in the following section with a fast definition of the problems, the solution used and the limit of this paradigm.

2.2.1/ Definition of the paradigm

The art gallery problem is a geometrical problem introduced by Victor Klee in 1973. The problem was to estimate the number (and the position) of useful guard to cover an art gallery. The particularity of a art gallery is the complexity of the room shape, with many wall to dispose the painting. The shape complexity of the room make the estimation of guards number even more difficult.
In order to formulate properly the problem, the room is assimilate at a polygon P, composed by n vertices $(v_1; v_2; \dots v_n)$. The vertices are linked by n edges $(v_1 v_2; \dots; v_{n-1} v_n)$ to make the shape of the Polygon P (or room).

A guard x is inside the room $x \in P$. A guard x can cover or see any point $y \in P$ if the segment xy is not intersect by any boundary of the polygon P (wall), in order to have $xy \subseteq P$. The polygon P is considered as fully cover when for any position of the point y in the polygon at least one guard can see him.
A guard x can cover at 360° all around him, with no depth of field limitation (except the wall obstacle). Clearly that mean the guard can see and monitor the entire length of the room one side to another side if no obstacle around to occlude. For example, if the shape of the room is a triangle, quadrilateral or another convex simple polygon, at any position taken by one guard, this guard can monitor all the area despite the size of the room.

The minimum number of guards X useful to fully cover the polygons P is $G(P)$ with k is the number of guard in order to have a set of points $X = \{x_1 \dots x_i \dots .x_k\}$ so that every $y \in P$ are cover by at least on point of the subset of X.
The AGP in addition to estimating the numbers of guard also are interested on finding the optimal position of this restricted number of guard. This 2 questions can be solve in the same time by using one of the solution proposed.

2.2.2/ Solution

The main advances on the AGP since this formulation in 1973 are numerous. The following paragraphs present the major advance on it.
The first and one of the more important is the proof given by Chvàtal in 1975 [13]. The polygon must have to be covered by a minimum of guard, the proof of Chvàtal propose to link the minimum number of guards to the number of vertices n. A polygon composed by

n vertices need in the worst case a minimum number of guard equal at $n/3$ The Chvàtal proof is based on the triangulation of the polygon. The Triangulation is made based on the vertices of the polygon.
The proof given by Chvàtal is also confirmed by the work of Fisk few years later (1978). The work of Fisk is also based on triangulation and colouring node. It is probably the easiest to understand and also give a solution to estimate the pose of each guard (it is recommended to begin by the Fisk proof before the Chvàtal despite the chronologic order as it is preconized in [54]). The book of O'ROURKE et al [54] is an early work about the AGP with the formulation, proof and advancement of the field clearly explained.
Once the proof of the minimum number in the worst case found the objective became to find an optimal solution in reasonable time.
For that the work of Toussaint and Avis (in 1981) is the reference and propose a solution working in $O(nlogn)$. This work has been follow and upgrade until the solution of Couto, Resend and Souza 2011 [14] a solution is finally proposed in $O(n^3)$ in the worst case.

It is a short overview of the AGP solution but also the solution proposed are very specific to the AGP and cannot be re-use for problem little different.

2.2.3/ LIMIT OF AGP AND CAMERA COVERAGE RELATION

The problem of positioning camera for coverage estimation is quite close then the AGP. In many ways, the AGP is a reduction of the camera positioning for a total coverage of a complex areas. The camera positioning is the logical continuation of AGP and once some AGP is knew, the problem can be extended as is for example show in [20, 58, 19].

The algorithm developed for AGP cannot be applied directly on the problem of cameras positioning for maximum coverage. The main reason is the cameras limitation field and depth of view ([12, 84] which makes is unreadable the solution proposed of solve AGP, where AGP considering the guard with no limitation for the depth of field and field of view. Also another reason make the AGP solution not applicable for the camera positioning is the diversity of cameras. In the same system the AGP may have many guard, they are all interchangeable because it has all the same ability (or skill) to monitor the area. This weakness in the AGP formulation associate to the limited field of view make the solution form AGP note adapted as is showed in [53, 34].
Due to this limitation, the solution developed for AGP are not applicable. However the formulation and some proof has their importance. Despite that, numerous article are based on the AGP to formulate the problem as in [19, 56]. For example in [19] the similar approaches then the AGP is used in order to estimate the occluded region. Also in [56] (page 105)to estimate the area covered a grid of $y = (y_1...y_i)$ is used in order to discretise the area. this method is directly based on the AGP formulation.
Also one important impact of AGP is the shape of the room. In [84, 34, 58, 19] the shape of the room are close then the definition of an art gallery. This phenomena can be imputed to the link did between the number of vertices and the useful number of guards (or cameras) to cover it.
The room composed with numerous wall inside create few occlusion. The occlusion mean the segment $xy \not\subset P$ where $x \in P$ is the camera position $y \in P$ is a points in the room P. The occlusion is one of the element make the camera positioning really complex. The use of room inspired by AGP is therefore a good choice in order to verify the effectiveness of the system in a complex environment.

The complexity of this problem is also an important factor which makes the relation of camera positioning and AGP.
The AGP is NP-hard problems [54]). The NP-hard proof is available on the book of O'Rourke section 9.2 of the book [54]).
To proof the AGP is NP-hard, the first part is to reduce the problem to an other problem welle known for this complexity. The relation is made by reducing the AGP with a polygon composed by holes to another standard problem (in the demonstration the is used 3SAT).
Once the AGP is reduced to 3SAT and because 3SAT is an NP-complete problem the AGP is also considered as NP-complete or NP-hard but only when the room is composed with holes. Also another work of Lee and Lin 1986 proof the complexity of AGP without hole also by reducing the AGP ton an other well known problem (for more explication see the book of O'Rourke section 9.3 [54])).

As we explained earlier the camera positioning can be reduced as an AGP [56] notably by removing most of the constraint due to the camera properties.
In the literature numerous article us the AGP reference to explain the complexity of the problem as for example [Moeini et al., 13, 46, 88] assumed the problem is at least NP-complete or NP-hard. The complexity of the problem will have an impact on the solution used to try to solve it and optimize it.

The AGP can be at same time for part the historical source of the cameras positioning and give some answer about the problem this formulation and this complexity. Despite that, the AGP is not the only source to refer about the cameras positioning for maximize the coverage. Some clue and solution can be find in other related field.

2.3/ WIRELESS SENSOR NETWORK

The wireless sensor network (WSN) can be as AGP considered as inspiration for the cameras positioning problem. The WSN is an active field of research and in many aspect related to the cameras positioning. This parties is focused on the WSN and this relation with the camera positioning. But first allow :

What is the Wireless Sensor Network ?

The WSN is a distributed network of sensor or in some case actuator, in this case is also called WSAN. Each sensor of the network is a relay for the information and commend, to the rest of the network. The sensor are at same time the node for the network. The node have to objective to transmit by relaying the information to the other node or to a centralized agent. Furthermore the node have to be the sensor for collect information and decided to communique with the other node. The information collected by the sensor are vast (extensive) depending on the final application and the capacities of the sensors ability. The WSN is used in different field for various application as for telecom and antenna positioning [78] military surveillance field [40, 72], airport surveillance [42], video surveillance and tracking [40], environmental monitoring [9]... Logically numerous sensor and information can be collected as temperature, movements, images, song and also some actuator can be used as radio frequency for example.
The application of WSN are wide, especially since the WSN has more than one discipline. The WSN try to optimize a network of sensor in different aspect as for example [82] focus on an architecture adapted and efficient enough for data transfer (image) or like in [65]

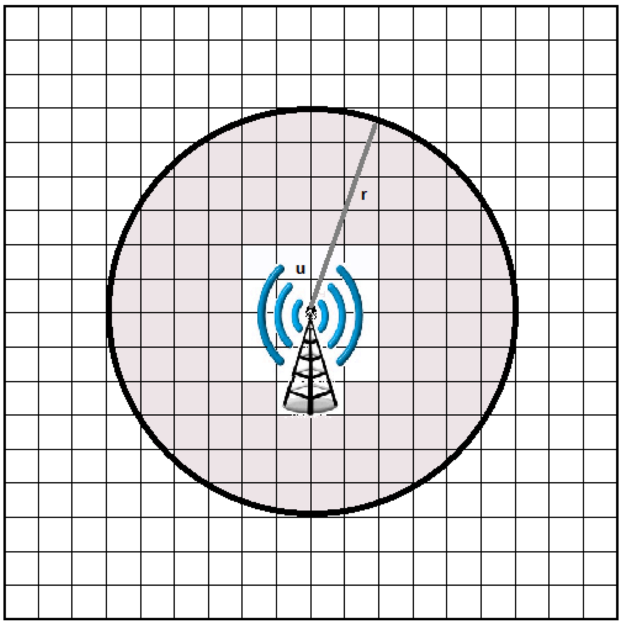


FIGURE 2.3 – one omnidirectional sensor centred on u, with a radius r for the range.

the WSN are dedicate to adapt the network around static node and energetic resources in order to keep the network connected.
For our case the discipline the more interesting is the coverage of an area with his specific constraint of the WSN for maximum coverage.
The other discipline of the WSN as the network optimisation will not be addressed in the following section. Only the problem of coverage is studied the other discipline are not considered as the first or main objective but can be some secondary objectives after the problem of coverage.

2.3.1/ SENSOR AT 360

The wireless sensor network (WSN) refer commonly to sensor or actuator with no restriction in the view angle, it is considering to have a $360°$ of field of view as in [36, 86, 11] (see Figure 2.3) and in some case a spherical as example in [47, 78].

Each sensors have a position u in the area and a power. From the sensor power the radius r of the circle is deduced from u as center (show 2.3). This circle give the area cover by a sensor in the simplest case. The simplest case correspond to a flat area without any obstacle, or it is negligible and do not impact the covered area as in [36, 86]. Others more complex solutions can be used. Its are more complex but also more realistic as in [78] where are taking in account the relief and obstacle. In [86] more complex model have been developed, where despite of a flat ground without obstacle each sensor is composed by a perception radius and a communication. The communication radius a bit bigger then the perception radius. This 2 radius correspond to area covered for one antenna (sensor) and the distance of emission/reception of the data. In order to have an efficient coverage of the area, the antenna must be place in order to have connection with other antenna but without too much overlap of the perception sensor.

The solution proposed in order to optimize the positioning of the WSN for a circular sensor can be varied. Mostly 2 different ways are applied for the sensor with have circular angle of view or spherical.

— The first solution use an heuristic based on geometry construction as in [47]. This approaches give a good coverage solution but is usually greedy and can be quickly limited in term of number of sensors, moreover if some external constraint are added. For example in [86] the greedy solution was tested and optimized by using a ”partition and shifting” strategy in order to upgrade the result. The limit of this solution despite the greedy consumption resources is also not applicable to the problem of camera positioning due to the reduced field of views of a cameras.

— The second solution try to find an efficient and quick solution to optimize random position for each sensor of the network. This solution include many different family of algorithms focused on optimization. Among the family of algorithms, the evolutionary algorithms (disused in detailed in section 3) is commonly used as in [36, 78], and [11].
These solutions propose to optimize the position in order to maximize the coverage depending on different constraint or the secondary objectives. The method of optimization have to be adapted to the problem. In [11] the integer linear programming with an ”LPsovlver” are chosen in order to maximize the coverage with 2 type of sensors. One standard with a smaller area coverage but with a smaller cost (can have a $100m$ radius for 150$) and the other sensor can cover a wider region (can have a $200m$ radius for 200$) and the objective is to cover the region but also to reduce the cost. The solution proposed is to use the integer linear programming adapted to the problem of coverage optimization with the economic cost in constraint.
In [78] and [36] the solution proposed is based on 2 different evolutionary algorithm in order to optimize the sensor positioning. In [36] the camera positing with a multi coverage is solved with using an evolutionary algorithm called particle swarm optimization (PSO). The objective is to optimize the position of the sensor in order to have an efficient coverage of the area and also enough redundancy to keep the network workable if one or few sensor fail. In [78] one evolutionary algorithms is also used to optimize the potion of antenna. The objective in this paper is to give the best coverage of an area with taking in count the relief of the area. The relief make the coverage estimation of each sensor even more complex and costly in terms of time computation. The Genetic algorithm are used in order to find quickly a position for each antenna of the network.

Among the examples seen the second solution based on optimization is the more interesting and also the more flexible to new constraints and secondary objective. The aim is to see up to what point and if it is applicable to the problem of cameras network positioning.

2.3.2/ Visual sensor network

Logically, the result obtained with the Omni-directional sensor are interesting and must be applied to the problem with even more constraint and objective as can be the positioning of visual sensor network (VSN). In fact the visual sensor or camera have a limited depth of field but also a limited field of view. This new constraint make the sensor positioning more complex as that was for AGP (see section 2.2) to passes from guard to camera. The advantage in this case, are first the sensor have been designed with a limited depth of field. Also the solution applied previously for circular sensor was not only geometric or based on heuristic but also the problem is formalized as optimization problems with some solution based on optimization and meta-heuristic. This way to present the problem is appear as the more suitable to add constraint and secondary objectives. Also the solution and the formulation for the problem of VSN are mostly the same then the problems of cameras positioning and is discussed in the following parts

2.4/ Solution not based on evolutionary method

The solution used to pose a set of cameras to maximize the coverage are various but among the possible solution manly two way can be describe. The solution proposed they come from numerous sources and paradigm which AGP and WSN.
The first is to construct a solution as an heuristic to have an appropriated cameras position. The second way is to formalize the problem as optimization problem and applied meta-heuristic.

2.4.1/ The constructive solution

The cameras pose can be done by construction. By construction that mean a deterministic method is applied to pose one camera after the other or to adjust their position.
In [40] a constructive solution is applied in order to select the smart cameras of the network. Each smart camera area a node of the network and transmit information, image and are fully autonomous. They work in 3 different modes :

- The first is the sleepy mode. The sleepy mode is used in order to economize a maximum of the energy. To do that the camera is turn off. That mean no computation and just the network is listen at regular intervals to wait the wakeup call.

- The Second is the detection mode the camera is turn on, but with a low frame rate. Just a few computation is done to detect if a target is enter in the field of views. Some information may be transmit by network. This mode consume more energy of the previews but the smart cameras can stay in this mode during a long time.

- The last mode is a tracking mode. This mode is the more energy consumer. The camera is turn on, with a high frame rate and numerous computation have been done to track and localize the target. Also more information have to be transmit by the network. The information are useful to localize the target by communicate with the potential other smart cameras with a view on the target.

The objective in [40] are multiples depending on the state of the cameras. the one the more interesting for us is to keep under control the longest time as possible the area for target detection. Numerous smart camera are randomly dispersed in the area (as an air-drop in a battlefield) and the aim is to select the camera in order to maximize the coverage of the camera in detection mode. The solution proposed by LIU et al in [40]. To do that the solution used is by construction with using a talk between each smart cameras of the network. Also each sensor is capable to estimate precisely this localization, it is an important element to select the cameras. The solution proposed is inspired of the network distributed talk.
To begin all sensor are in sleepy mode. The camera wake up regularly and send a call at the neighbour, if they receive no answer that mean no other camera is awake around and the camera stay in detection mode. If no enough answer have been received, that mean the required density of the camera around are not enough the camera stay turn on detection mode. Otherwise the camera return on sleepy mode until another wake-up later. This procedure is applied in each smart camera with all have their own pace. After a certain time the network is well organize to cover the area in detection mode. The method introduce by LIU et al in [40] work well. the solution proposed have the advantage to can work in wide area and to be dynamic (if one cameras do not have any-more power the network ll re-adapts).
The solution are efficient with a high density of sensor to have a relatively low coverage (just enough to detect target in the sparse sensor placement). Also the method presented is really dependent then the network communication and the capability to localize precisely each sensor. Finally the solution proposed to select a set of sensor among a randomly posed sensor is not enough optimized.

Another cameras pose estimation by construction is proposed by HÖRSTER et al [34]. The solutions proposed is based on a greedy search heuristic. The objective is to find a position and orientation of a set of cameras with a fixe pan, in the environment inspired by the AGP.
In [34] a first greedy search solution have been presented then finally a solution called Dual Sampling has been presented based on it. The dual sampling is an incremental method.
First step is to initialize the position of all the cameras. A random initialization for the position and the orientation must be appropriate.
Second step is to select one point of the area. The area is discretized by numerous points, with each point must be covered by at least one camera. Around the selected point several position and orientation are tested for the cameras at proximity. The possible position are obtained by sampling the area around the point to cover. Finally the best cameras position and orientation is kept. The second step is repeated and the set of uncovered control point is reduced an each iteration. This procedure is applied until the stop criterion is reach that mean enough points of the area are covered.
This constructive solution have some inconvenient notably in terms of efficiency. Indeed this solution is limited by the number of camera and size of the area due to the exponential complexity.

In [53] the camera placement to cover a basic mobile robot trajectory is studied. The trajectory is modelled as the region of interest, with a gradually decreasing interested from the trajectory center.
The solution applied in [53] is to do a local optimization one camera after the other with the “steepest decent method (decent de gradient)”. If this local optimization give a better solution the network of camera is modified otherwise the camera stay at the same place. This operation is repeated until the convergent arrangement is obtained or no more upgrade can be found. This result presented in the experiment done by Nikolaidis in [53] are interesting despite the simplicity of the area and the very small amount of camera used. The principal limitation is due to the number of step required for optimize independently each camera of the network. Also a multitude of local optimisation is not obviously the same then the global optimization.

Ma et al [42] propose a solution for the problem of finding the minimum camera barrier coverage. The objective is to cover only the boundary of an area, to be able to detect target intrusion in the perimeters. The perimeter is relatively wide and can be considering at some point as an area to cover composed by big hole in this center.
The region to cover is cut in numerous sub-region. The region are inter connected, each sub-region have to be “full-view covered”. The full view covered is define if for any target direction there always exist a sensor cover to the face of it.
The solution purposed is based on constructive solution with adapted heuristic (the heuristic used is presented on [42]). The global idea is to cut the perimeter in different sub-region and applies the method presented to full view cover each sub region. The cutting on sub-region offer at the heuristic to work in reasonable time due to this restricted area.
The solution proposed though this efficiency in the case of barrier coverage is not rely appropriate to vast coverage area. The first limitation is the number of cameras use to fully cover the area. The number of cameras is mainly due to this definition of “full view covered” . That definition imply many overlap to have the multi direction coverage.

As show, the previous solution are based on constructive method for the camera placement and their local optimization. Each camera is placed dependently then the network with an iterative process in order to have a position for each camera of the set. This methods have some consequence, nobly the fast increasing number of iteration useful to have a solution good enough. The time complexity is event more problematic with increasing the size of the area. Also these solution are extremely dependent then the formulation and the constraints and cannot be easily adapted to other problems.

2.4.2/ !!! Linear programming optimisation !!! (to verify)

Different method of linear optimization was applied and test. In some case due to a well-adapted formulation ; a specific shape of the area or cameras number The linear optimisation is efficient. The linear optimisation has been commonly used to have initial result in order to have a reference point before to develop other solution more appropriate(as example [1, 88, 12, 58]. . .).
The linear optimisation is finally rejected due to this fast limitation. Indeed the linear optimization can be quickly in difficulty due to the fast increasing complexity of the

problem and in many case be lock in local minima. as presented in the flowing example.

The [19] is based on AGP and the WSN and propose a fusion off their 2 paradigms in order to use there assumptions. Some modification have been done as example to taking in account the field of view limitation. The interesting aspect is how some camera properties have been modeled to fit to the problem of AGP.
The solution proposed is workable with using an omnidirectional or considering a PTZ as omnidirectional cameras by using an efficient angular sweep. The simulated omnidirectional cameras is simulate by PTZ camera with a non-continue zoom or fix focal length as in the experimentation (see fig 7) where the PTZ can have a focal two focal length at 50mm and 35mm.

Finally the solution proposed is to discretize the area to cover (fig 7 left) and also the different possible parameter for the camera (as : localization, orientation, focal...). Thanks to this discretization and a formulation close then the binary integer programming (BIP) and apply a well-known method “Branch and Bound” in order to optimize the camera placement.
This solution propose a good coverage with the minimum of cameras in reasonable time until, the number of location sample (point in the grid on the floor) , the numbers of camera, and the number of parameter possible for the camera stay relatively reduced.

Zhao et al [87] are trying to find the optimal position for a set of camera in order to maximize the Indore coverage (like an AGP room) and also to do tag detection. The solution proposed for the coverage is to adapt the number of point of interest wish must be covered by the camera.
The area is discretize in grid. The grid is composed by point selected smartly depending on the coverage rate. Each point of the grid simulate a potential location of the target.
Also a limited number of possible position for the camera (the boundary of the room) are defined. The adapted grid and the limited camera positioning is used to limit the complexity (as number of possible solution). linear optimization to the Binary integer problem formulation.
Use a Binary integer problem formulation (BIP) in order to apply a linear optimization is popular and other use it as in [87, 5, 19]. In [87] the solution proposed is to use BIP formulation the grid smart sampling and LP_solve to optimize the camera position and orientation.

A Similar method is used in [5] to find the position and orientation of a set of cameras. The goal of this paper is to find the appropriate position with maximize the resolution for a set of cameras dedicate to do tracking. The position is determined depending on a set of standard pedestrian trajectories. The camera pose try to maximize the given trajectory to have the best resolution and the entire coverage.
The problem is also formulate in order to apply a linear well-known algorithm. In this case the branch and bound is applied.

2.4.2.1/ Limitation of linear method

Different method of linear optimization was applied and test to answer, the problem of camera position for maximum coverage. In some case due to a well-adapted formulation or a restricted area and cameras number this solution is efficient enough. In some other case the method was studied but finally rejected due to this fast limitation (as example [1, 88, 12] ...). Indeed the linear optimization can be quickly in difficulty due to the fast increasing complexity of the problem and in many case be lock in local minima.

In WANG et al [77] propose a solution with an atypical problem formulation. The solution proposed in [77] is mainly based on the method of discretize the area. The idea is to have an area discretized with precision with use the minimum of point. To do that the solution proposed is to decompose the area in order to give more point in the grid were the shape of the room need it to be correctly described.
The principal advantage of this solution is to propose an area representation with enough precision and a minimum of point to describe it. Less point to describe the area to cover mean also a winning time efficiency in computation during the camera pose estimation (cost function is faster). Despite this interesting solution the result presented in the experiment does not appear to be conclusive.

2.4.3/ Game theory

Among possible solution an atypical method is to use the game theory [37]. The game theory is use to optimize the looking direction of the cameras as in [68, 17, 37, 66]. These articles are based on game theory to find an equilibrium (also named Nash equilibrium) between two contradictory objectives, the maximize the resolution and the multi target tracking.

Soto et al [68] propose a network of a dozen of PTZ cameras. The PTZ is for pan tilt zoom, that mean the position of the cameras are fix and the solution proposed is to find the best orientation with the appropriate focal lens to track most of the targets.
To do that the camera are smart enough to communicate with the close neighbours and adapt the pan, tilt, zoom depending on the needs.
The need has been defined by an utility function (or local cost function). The goal is to track most of the target as possible with the better resolution. The cameras scores when it obtained a desired resolution image for all the targets visible by the networks.
The trade-off proposed is between the multi tracking and the best coverage resolution for each target. The multi-target problematic can appear far from maximization of the coverage. But the number of targets which may be higher than the number of cameras, push the camera tracking to be an interesting solution to maximize the coverage of an area. In this case the quality of the coverage will also dependent then the number of targets and the importance of the resolution constraint.
In [17] and [66] different experiment have been proposed with a number of target is lightly increased. In this articles the game theory is applied to trade-off between the tracking and the resolutions. The proposed experiment is based on the decentralized method. It is justified by the complexity to dynamically adapt all the camera of the network at same time depending on each targets trajectories.

Furthermore, the decentralized solution is more adapted for the security of the transmission and the risk of interception by a hostile opponent. The security may be an important factor in some applications.
To have the decentralized system the camera must have some autonomy.
In the experiment proposed in [68] and [17, 66] each camera are smart enough to have their own tracking and control module. Also the camera are able to communicate with each other to come to a consensus. The consensus is when a Nash equilibrium is found between the 2 contradictory objectives. In this case it is a win win situation for both objectives.
So the objectives are not independently optimize, moreover the solution proposed is to optimize its simultaneously in order to have a consensus. The consensus is reach when it became impossible to upgrade one of them without downgrade the other one.
In the experiment of [17, 66] the consensus is found by the camera communication in order to have a maximum of the target cover with the higher resolution. The experiment is based on numerous target moving freely in the area. Also in [17] one of the target must be cover in priority with a high resolution. That will have an impact on the other camera position. The result of the experiment as in figure 2.4 form [17] show a really efficient global coverage with almost all the target cover at every time despite the movements of the targets.
The advantage of these solutions is the acceptable result and dynamic reconfiguration of the system for the reasonable size of the area and the decentralized computation.
Otherwise this method have some limit, as show in the experiment the area is relatively restricted and numerous cameras with a fixed position are useful. The consequences is the quantity of overlap relatively important. In this case the number of sensor is not well optimize to cover the area. Also to use properly this method for maximum coverage it requires a large number of simulated target in the region to observe.

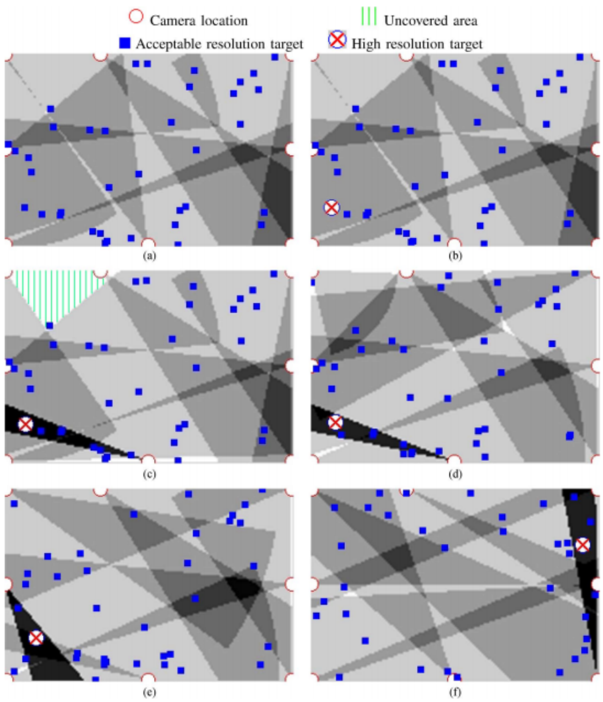


FIGURE 2.4 – result of coverage after have used the game theory with objective to maximize the tracking and the resolution. result obtained in the experiment presented in [17]

TABLE 2.1 – Sum-up ref

ref	**Best solution**	X	Y	Z	Pan	Tilt	Roll	focal lenght	Coverage room	number of cameras	Secondary objectives and constraints		
[42]	heuristic	✔	✔	0	✔	0	0	fix	2D	≈ 10	barrier coverage	conectior depen-dance	
[40]	heuristic	random dispersion (x ;y)			0	0	0	fix	2D	≈ 600	resolution	tracking	
[5]	non-linear	✔	✔	0	✔	0	0	fix	2D		tracking	resolutior	
[19]	game	✔	✔	0	✔	0	0	✔ (discret)	omni di-rectional (by pan)	≈ 10	cost reduction	resolutior	DoF
[68]	game theory	0	0	0	✔	✔	0	✔	2D	≈ 10	resolution	tracking	multi target
[17]	game theory	0	0	0	✔	✔	0	✔	2D	≈ 10	resolution	tracking	multi target
[66]	game theory	0	0	0	✔	0	0	✔	2D	≈ 14	resolution	tracking	multi target
[53]	heurisitic	✔	✔	0	✔	0	0	fix	2D	2 to 4	Region of interst	trajectory coverage	
[87]	LP-solve	✔	✔	0	✔	0	0	fix	2D	≈ 10 (8-11)	tag detection	tag visbility	
[84]	linear programming relaxation	✔	✔	0	✔	0	0	fix	2D discrete square	≈ 20	Region of interst		
[34]	LP-solve	✔	✔	0	✔	0	0	✔ (discret)	2D	≈ 10	cost reduction		
[77]	Multistage Grid Subdivi-sion	✔	✔	0	✔	✔	0	✔	2D	9 to 15	Region of interst	big area	

2.5/ SOLUTION BASED ON EVOLUTIONARY METHOD.

The solution already broach formulate the problem of camera positioning (as in the section 2.4) from various point of views. But the more common have not been approach yet. In numerous work the camera positioning for maximum coverage is assimilated to a problem of optimization. Due to the complexity of the problem (as that was introduce in the AGP section 2.2) the linear optimization may not appropriate. The solution is to apply some stochastic method with an appropriate meta-heuristic. Among the various possibility the algorithm from the Evolutionary algorithm has been used at numerous time.

In [88] different solution have been tested to optimize the position. Among the solution proposed a greedy with a local optimization, a sampling algorithms and the Simulated Annealing (SA) from the EA family has been compare. Also the SA is used to customize the sampling mechanize in order to propose a new customized solution.
In Zhao et al [88], the problem is formalized as a BIP (binary integer programing). The goal is to find among a restricted numbers of possible positions and orientations the best pose for a set of cameras. To cover the area, despite the obstacle and the region of interest.
Among the solution proposed, the greedy algorithm has a good approximation until the number of camera stay relatively small. The problem of the greedy algorithm is to have an important risk to be stock in the local optimum. This risk increase proportionally then the size of the environment. Otherwise the sampling technique (with the SA inspiration) is more adapted to the problem with times constraint and offer well solution.
Despite that the solution proposed have some limitation. The major limitation is due to the experiment proposed. In fact the experiment is made in very small room with only few pose possible for each cameras. this have to effect to limit the complexity and reduce the search space. Consequently the solution proposed can appear better then a real situation and must be widely worst in a bigger area (in terms of time computation and optimisation of the pose).

In [1] the camera positioning for outside, have been designed as an optimization problem with few solutions tested. Among the solution tested the SA and Gentic algotrithme (GA) both form the EA Family. The work of Akbarzadeh et al [1] are interesting at different point.
The problem formulation is one of the interesting point. The goal is to cover an outside area with obstacle (and their potential occlusion) , k-coverage and the relief of the terrain. The cameras are always pose at the same distance then the floor despite the altitude and depending on the position and orientation of the cameras the occlusion generate by the obstacle and the relief has been computed. All these elements combined make the problem formulation relatively complete and realistic. The work of Akbarzadeh et al [1] are interesting at different point. The problem formulation is one of the interesting point. The goal is to cover an outside area with is composed by obstacle and their potential occlusion and the relief of the terrain. Moreover then the relief and the obstacle the area is composed by some region with a k-coverage constraint. Due to the relief of the ground the camera may not be pose always at constant altitude. The solution proposed is to place the camera at fix distance then the ground floor despite the altitude. Like that the altitude of a camera is automatic deduced form his position and the relief associate. All these elements combined make the problem formulation relatively complete and realistic. Among the solution studied in it, the non-linear have been tested with an quasi-Newton

optimization methods (called BFGS see in [1] section C), the SA and the CMA-ES form the EA family with some mechanism close then a Genetic algorithm (see in [1] section D) have been tested too. The result of the comparison give a net advantage of the algorithms from the EA family n term of coverage rate with a fix number of cameras but also in term of time computation. The SA give a close coverage rate (some time slightly better) then the CMA-ES and also the SA have a better time computation. Otherwise the CMA-ES is the more efficient in average with a reasonable time computation close then SA. The the quasi-newton is far away in terms of time and coverage compared to the EA solutions.

In Chrysostomou and Gasteratos [12] optimization of the coverage with several constraint is presented. The goal is to cover an inside area inspired by the AGP with the minimum of cameras. The minimum is fixed by a threshold depending on the coverage. The other goal is to maximize the coverage for a fix number of cameras. This 2 objectives are relatively close and just few element have to be adapted to pass from on to an other problem.
The coverage of the area is related to the camera visibility and few constraint have been proposed to control it, as visibility, viewing angle, field of view, resolution, viewing distance, and occlusion. The solution proposed to optimize the position, orientation and some of the camera parameters (as the focal lens) is to apply an algorithms form the EA family called Bee Colony inspired by the Ant colony algorithms and the bee exploration in the nature. The Bee colony is in this paper the more efficient in the 2 problem formulation (minimum of cameras and maximum of coverage with fix number of cameras) after a brief comparison with a GA and a broach and bound method.
The experiment proposed is relatively limited only one room has been tested with only one test per algorithm with is not relevant for stochastic algorithm (due to the importance of randomness).

In some solutions proposed as in [12, 58, 1] the Genetic algorithms have been used as comparator elements. The GA and solution strongly inspired by the GA are not only used as as a comparative elements but it is also in some case appear more efficient as in [74, 72, 35, 76].

2.5.1/ Solution used Genetic Algorithm or close related

Among the solution applied from the EA to optimise the coverage area, the GA or the algorithms closely related as been use as the following examples.

In [74] the problem of cameras positioning for video surveillance to control building is studied. A building with different floor is considered and must be covered by a set of cameras. To do that the environment have to be considered in the 3 dimension with the floor occlusion. The solution proposed is to fix the altitude of the cameras for each level of the building and considering each level as one independent room to optimize.
The optimization is based on a basic GA with a customized crossover and mutation in order to fit to the problems. For the crossover a swap between two cameras from two set of camera position (chromosomes) and for the mutation a Gaussian is use to perturb some to the camera parameters. Also the GA is used with an elitist selection. All this parameters of the GA are essential to have an appropriate optimization depending on

the problems and to understand the performance of it. Finally the area is covered with 32 cameras (for 3 levels) with relatively good coverage. Few black hole and an acceptable level of overlapping.

In [35] the solution of coverage problem for wide area with region of interest and obstacle have been optimized with using a standard GA. In [35] the GA have been tested for wide area with numerous sensor to position and orient. The experiment have been done in a vast area $400X300m^2$ with 60 cameras with all have the same depth and field of view. The number of camera are not enough to control the full area and some choice must be done between the different region of interest. The GA proposed in this article offer the best result around 47%, with a minimum of overlapping.
The solution proposed is interesting and work well to optimize the pose in big area with lot of cameras. one element is missing is the algorithms behaviour with an number of cameras supposedly enough to fully cover the area.

In [76] a variant of the classical GA have been used. The Multi Agent Genetic Algorithm (MAGA) [89] have been developed by the fusion of 2 algorithms, the GA and the multi agent systems. The principal advantage of this algorithms is this supposed efficiency on the optimization for a huge number of dimension (Zhong et al estimate the efficiency range around 20 to 10 000 dimensions).
MAGA is used in [76] to optimize the area coverage. The area to cover is a 3D area and most the volumetric space have to be covered. Unlike the other article the cameras have to be positioned in the 3 dimensions (in x, y and z) and the orientation too. The constraint of volumetric space associate to the optimization of the position, orientation and cameras property, increase greatly the size of the search space. Despite this interesting solution the experiment proposed are not enough consistent to properly judge it. But appear promising in term of potential increasing search space and answer quality.

In [72] the solution proposed is an Hybrid Evolutionary Algorithms (HEA). The HEA is based on the GA with different modification, notably on the operators. For example the cross-over is redesigned in order to have two different part.
One of the part is a local cross-over and the other is a more classical cross-over. The HEA proposed in this article is relatively close then another EA algorithms called the Mimetic algorithms. The solution presented in [72] is presented with different experimentation dedicate to maximize the visibility and minimize the cost of the sensor network. The experiments propose to compare an simple random selection to the HEA. The result of this experiment show a real efficiency in terms of minimizing the number of useful sensor and the total utility (the utility are related then the multi objectives.)

In view of these articles the GA appear appropriate for the optimization of numerous cameras in vast area as show in [35]. The GA optimization can be a good starting point to develop an new method by tuning and customizing the numerous parameters as in [72, 76] or [74]. Also despite the example presented the GA is relatively under-exploited in view of this customized ability and efficiency.

2.5.2/ Solution used Particle swarm optimization or close related

To maximize the coverage despite various constraints the EA family are commonly used with different algorithm as showed. But the one the most commonly used is the Particle

Swarm Optimization (PSO) as in [83, 91, 58, 44, 24, 23, 36].

In [83] the PSO is used to optimize the orientation of several camera (around 150). The camera has been randomly posed on the area. Each camera are the same and cannot change its position but may adjust its orientation to any directions. The area is designed in 2D and the looking direction is described with only one rotation in pan. Finally the PSO is used to optimize the looking direction in order to optimize the coverage. The PSO optimize the area to reach a coverage around 65% after 1000 iteration and 20 particles for each. The intimal coverage in the experiment is around 53% after the random dispersion.

In Fu et al [23] the same problem as [83] is discussed. The optimization of the orientation for the important set of cameras (100 cameras) using PSO. The difference is the number of parameters to optimize. In [23] the looking direction is not only optimize on pan but also with the tilt.
The PSO is compared with other algorithms as the SA. Finaly the PSO outperform the other solution experimented in this article.

In Zhou et al in [91] the objective is to cover an indoor area inspired by the AGP to detects targets. The room is represented by a set of possible target position. The coverage of the target is optimize for a fix number of cameras.
The experiment made in [91] show the ability of PSO in term of efficiency and speed compare then a hierarchical method. The proposed hierarchical approached is a greedy constructive heuristic (see more in 8* section III). The experiment is made in a basic scare room described by a grid of 81 possible target position with all have the same importance and have to be covered.
The result of the experiment did it in [91], shown the slightly advantage of the hierarchical method. In fact the hierarchical method give a better solution but with a much more time computation. Otherwise The PSO is the solution that combines efficiency close then the hierarchical method and time efficient between 2 and 6 time faster than the hierarchical method.

As in [91] and the [58] propose a similar solution for a similar problem formulation. The goal is to maximize the coverage depending on some target priority with must maximize the resolution using PSO.
Moreover the solution proposed in [58] is also taking into account the visibility parameter as depth of field and light intensity in a room inspired by the AGP. All this constraint affect greatly the coverage result. Finally the experiment did it shown efficiently of PSO for a small environment with few camera (around 7 cameras) for a multi objective problem (more objective and constraint then in [91]).

In [24] the PSO is use to optimise the problem of maximum coverage as the other. The camera pose and orientation need to covered the totality of a small square room. The orientation is defined by in pan and tilt and all the camera have the same focal lens. It is a relatively basic objective with few common constraint. The interesting part of this article is the use of PI-PSO with is a Probability Inspired binary PSO.
To use this algorithm the problem have to be adapted. This adaptation give an original

formulation. Despite this the solution proposed is greedy in term of camera for cover a small space without obstacle.

[44] the finality is different but the problem can be considering close then the problem of camera positioning. The objective is to position several transistors in a rectangular printed circuit board (PCB). The transistors must have a rectangular shape with different size and ratio.
Despite the apparent simplicity the several rectangular shape for the chips and some potential additional constraints, as the relation dependence between chips, or the non-overlapping of the transistors, make it becomes complex. The solution is to optimize the position and orientation of the chips on the board using Evolutionary algorithm. The PSO is chosen and more precisely the craziness based PSO (CR-PSO). The interest of the CR-PSO is the variety introduce during the optimization compare then a classical PSO. The variety introduced during the optimisation process by the CR-PSO is help full top pass over the local minima.

The solution proposed based on PSO are promising. The PSO or close related have numerous advantage as shown in the previous articles. The main advantages are the efficient and fast optimisation, as well as the fast implementation. In fact the PSO is relatively easy to implement and just few parameters need to be set-up before to have efficient result. The simplicity of the implementation is due to the numerous framework developed in all languages and the low numbers of parameters to set-up as manly number of particles the global inertia.
On another side the PSO is not really efficient to optimize lot of dimension at same time and is quickly limited as the increasing size and complexity of the area.
The popularity of the PSO is mostly due to the association of the quick implementation, fast computation and good optimisation.

ref	Best solution	Other EA solution	X	Y	Z	Pan	Tilt	Roll	Focal length	Coverage represen-tation	Indoor outdoor	Number of cameras	Secondary objectives and contraints		
[72]	HEA GA	✔	✔	x	o	✔	o	o	fix	2D	out	10 to 200	maximizi infor-mation	minimizi total cost	cost reduc-tion
[58]	PSO	o	✔	✔	o	✔	o	o	fix	2D	in	≈ 10	tracking	light intensity	
[91]	PSO	o	✔	✔	o	✔	o	o	fix	2D	in	1 to 20	tracking	resolutic	
[12]	Bee Colony	✔	✔	✔	o	✔	✔	o	(2)	2D	in	≈ 10	cost re-duction	resolutic	
[74]	GA	o	✔	✔	o	✔	o	o	fix	2D+	in	≈ 32	min overlap	multi level	
[83]	PSO	o	random dispersion			✔	o	o	✔	2D	out	50 to 600	region of interest		
[1]	CMA-ES	✔	✔	✔	relief	✔	✔	o	fix	2D + relief	out	10 to 110	relief	region of interest	
[44]	CR-PSO	✔	✔	✔	o	0	o	✔	✔	2D rectangle		≈ 15	Shape propor-tion	depende	rectangula chips shape
[88]	SA*	✔	✔	✔	o	✔	o	o	fix	2D	in	6 to 30	tracking		
[76]	MA-GA*	o	✔	✔	✔	✔	✔	o		3D	in	<30	Visible FoV	cuboid obstacle	
[35]	GA	o	random dispersion			✔	o	o	fix	2D	in	≈ 60	region of interest		
[24]	PSO	o	✔	✔	0	✔ (4)	✔	o	fix	2D	in	15 to 25	resolutio	focus	
[23]	PSO	o	✔	✔	✔	✔	✔	o	fix	2D	out	<150			

2.5.3/ Field of view

optical and the occlusion

cover an area with certain amount of sensors. This number of position can be estimate as follow.
Each camera defined m focusing of optimize the solution and return a acceptable solution.

2.6/ Coverage path planning project CPPP

2.6.1/ AGP to watchmen problem

2.6.2/ linear decomposition and sweep

2.6.3/ Robot application

3

GENETIC ALGORITHMS

Sommaire

3.1/ DARWIN AND THE NATURAL SELECTION

The theory of evolution was introduced by Darwin and inspired the computer science for developing optimization algorithms. To understand the algorithm is important to go back to the origin. The following section is focus on the fundamental theories of the natural selection and this history.

3.1.1/ DARWIN THEROY

Darwin has studied the differences between individuals from the same species and tried to establish a classification of the different sub-species. It appeared some individual from the same species and from different countries had some small differences. These variations were studied and explained by the Darwin theories in The Origin of Species, published in 1859.

The origin of species details what will be called the theory of evolution.

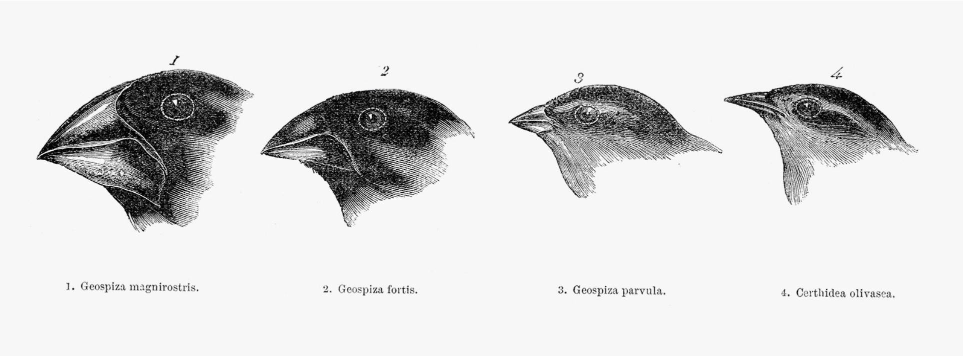


FIGURE 3.1 – Finches from Galapagos archipelago extract od The Origin of species by C.Darwin .

This theory uses the concept of adaptation introduced presciently by J.B. de Lamarck, and deeply studied by Darwin. The adaptation explains the relation between the environment of the individual and the differences generated by natural selection. This observation was first made on bird, called geospiza, or finches, (chaffinch) from Galapagos archipelago. Darwin noticed the difference of their beaks (see the Fig 3.1). The shape of the beak was correlated of the specificity of each island. Finches with the biggest beak correspond to the island with the biggest seed.
This observation was formulated and explained by Darwin by the adaptation of the bird in their environment.
The adaptation is partially due to the natural selection. Indeed the selection is done by the reproduction of the strongest individuals.
The reproduction concerned 2 individuals (one male, one female). Each individual is in competition with the other individuals of the same species. In its condition only the stronger and the more adapted individuals have a chance to have a progeny (an offspring). In fact generation, after generation, the more adapted individual itself reproduced and mute, while the species adapt to their environment. In this case, the strongest finches is the one with an appropriate beak in order to eat more seed.

3.1.2/ Biologic evolution

The Darwin theory of evolution was contested during long time until the confirmation by the progress of biologic sciences, especially with the genetic progress. The progress in this field was used to study the mechanism of the natural selection and evolution. One of the important progress and confirmation of the theory is by using the analyse of DNA code. DNA for DeoxyriboNucleic Acid contain all the useful information to the grow, life and reproduction of life.
The discovery of DNA code permits to confirm the genetic proximity between some species. Moreover the DNA permit to evaluate their evolution and code modification in the same species from different location and several generation gap. Thanks to the genetic and DNA the evolutionary process has been explained deeper and clearly.

In the biologies, every living element is composed by cellular. Inside each cellular the DNA code is stoked. The DNA composes the chromosomes. The chromosomes have a central role in the definition of one individual and in the reproduction process (see the Fig 3.2).

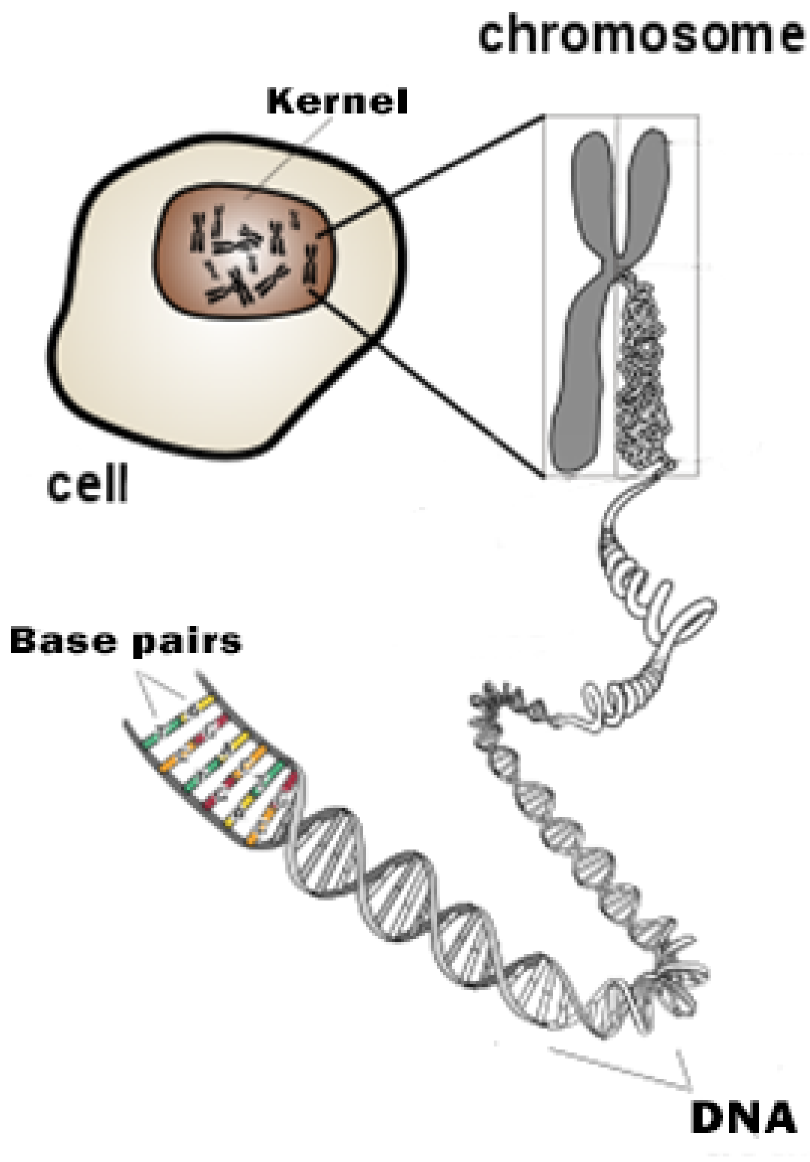


FIGURE 3.2 – Biologic representation, cell to chromosome until DNA.

As has introduced earlier, the evolution is possible by a natural selection. The selection is done by survival and reproduction ability of the better individual. Better mine more adapted to this environment, like for the geospiza the size of the beaks depending than the size of the seed from his island. In the island with the big seed the geopiza with the small beaks was not the more adapted and have more difficulties to find food. This weakness make the geopiza with the bigger beaks in better position to reproduce and the DNA of the individuals with biggest beak is transmit at the next generation.
The understanding of the reproduction mechanism in term of succession and transfer of the natural ability is primordial. It is explained in part by crossover in the cellular state.

Sexual reproduction. The living element for example the geospiza use sexual reproduction. The sexual reproductions assume to merge part of the chromosome form the

two individuals to create their descendants. The selection is essential in order to keep the more adapted individuals of the species.
Among the reproduction mechanism the crossover and the mutation are the one with the bigger impact.

Crossover. The crossover is the action of merging the chromosome of two individuals in order to have a new child.
It is an essential factor to preserve the individual ability of the geospiza. But the natural section and the crossover cannot be considered as the only useful element to evolve.

Genome The genome is a subset of the DNA code. Communally the DNA is cut in many thousand genome where each genome can represent a specified function or ability.

Mutation. Instead of the crossover which allows only the ability preservation from one generation to another, the mutation introduces some “anomalys”. The anomaly can become in some case a biologic advantage. The mutation affects only rare chromosomes of the individual. The chromosomes affected are not fully mutated but only one or two genomes are affected.
The mutation changes the little piece of DNA code to introduce variety in the genetic code by the small “anomaly”.
Most of the time the small mutations are not consequent for the individual but generation after generation the mutation can be preserved and spread in the population.

The giraffe can be taken in example :
In the arid environment, the giraffe with the longest neck has more chance to survive due to this empowered to find food. The giraffe with a neck a bit longer than the other, can became more attractive for the natural selection (in this case more food mean stronger and more attractive). The natural selection pushes the best individual to reproduce together and by the crossover mechanism conserves the small advantage given. The initial mutations give at few giraffe a longest neck and by the process of natural selection associate to the crossover allows this advantage form a small mutation to become the norm. Mutation by mutation and generation after generation the giraffe saw the average length of their neck increased. Finally the actual giraffe is the result of a long and complex evolutionary process.
The mutation can also be the source of degenerate animals but in this case the natural selection by the reproduction (and crossover) will not allow the preservation of the individual and thereby the mutated chromosome will disappear.

3.2/ The evolutionary algorithms

The evolutionary algorithm (EA) is a big family of algorithm and they included many meta-heuristic used in the field of optimization and artificial intelligence.
The evolutionary algorithm are inspired by the biologic mechanism for design meta-heuristics. The origin of the inspiration can be varied as the genetic, insect work, animal

Inspiration or group	Algorithm
Based on memorization.	Neuroevolution
	Learning classifier system
Animal inspired and swarm algorithms	Particle swarm optimization (PSO)
	Bee colony
	Ant colony
	Mimetic algorithm
	Shuffle frog
Swarm algorithms	Addaptatif dimensional search
	Gaussian adaptation
	Genetic Algorithm
	simulated annealing
Combinatorial	Harmony search
Genetic	Genetic programing
	Evolutionary programing
	Evolutionary strategies
	Evolutionary programing
	DarwinTunes
	Genetic Algorithm

TABLE 3.1 – Liste of few basic EA

behavior, . . . (see Table 3.1).
The biologic inspiration are not the only elements use to define the evolutionary algorithms. All the algorithms in this family are dedicate to optimize iteratively a population of solutions. The EA family are not deterministic and use a randomised function in order to evolve.
To summarize the EA have most of this attribute :

— Bio inspired.
— Use random (not fully deterministic).
— Based on population.
— Evolve a set of solutions to optimise a problem.

These characteristics are not the strict definition for all the EA. They are rather the most common element of the major part of the vast EA family. The EA family include several sub-categories as in Table 3.1. In the following part the EA is approach more in details, first a brief historic of the EA.

3.2.1/ HISTORIC

The EA are relatively young and do not have one fix origin. It is the result of more than a decade of research and improvement. The premise of the EA can be the work of Robbins et al [59] in 1951. More commonly, the beginning of the EA are on the late 50s with the works of Bremermann [8], Friedberg [22], Box [7]. They propose different algorithms base on the evolving solutions to optimize a problem.
During almost the three next decade, the research had slowly progressed and they have remained rather unknown. Mostly due to the lower computation power at this time and also to some methodological short comings of those early approaches.

Instead this difficulty, the fundamental works of Holland [32] and Fogel has been essential to the progress and to popularize the EA.
As of 90s, due to the fast increasing computation power the EA became more popular and numerous new algorithms form EA family have been designed as listed in the Table 3.1. The application of the EA in the engineering field (examples in [2]) and the multiplication of the conference around EA allowed the democratization of these family of algorithms. The EA has been profit of three main and independent methodologies ; evolutionary programming, evolution strategies and genetic algorithms (GA).

- The evolutionary programming, especially the work of Fogel is based on the finite state machine. The goal is to predict events based on the inputs, it is one of the premises of machine learning and classification.

- The evolution strategies, especially the work of Rechenberg ([57]), which propose a strategies based on deterministic selection and random mutation. The goal was to solve difficult experimental problem with discrete or continuous search space.

- The genetic algorithm is the most probably the most polyvalent and diversify of the EA (in terms of algorithm mechanisms). The GA propose an adaptive processes to optimize a solution. The detail of GA mechanism will be explained precisely in the Section 3.1.1.

The GA has more particularly attracts the interest of the research with the work of Holland and Goldberg. The popularity of the GA is most probably due to the increasing power of computation (in 80s, 90s) associate to fundamental progress and the numerous possible applications in the optimization field.

Since the late 90s the research of EA have been focused on the multi objective evolutionary algorithms MOEA [90, 85, 67].

3.2.2/ General formulation

The EA is vast and many type of algorithms exist in this family, despite that a global formulation is proposed.
Most of the EA have to optimize one or several problems using an iterative process to evolve towards a best solution. The EA can be formulated as the optimization of a parametres vector. Where each element of the vector ($\vec{x}$) is one input or a dimension to optimize.

$$\vec{x} = \{x_1, ...x_n\} \in \omega \tag{3.1}$$

Where ω is the search space of the problem, n the number of dimension to optimize(or number of input). Each dimension of $\vec{x}$ must have a limited range (as $\sum_{i=1}^{n} \sup x_i \leq x_i \leq \inf x_i$). The search space represents all the possible solution of a problem.
In the case of the elements in the vector $\vec{x}$ are ordered, the search space even bigger. In this case, it is the product of each range of the vector.Where $\|x_i\| = \sup x_i - \inf x_i$ it is range size of x_i thus the size of the search is define as $\prod_{i=1}^{n} \|x_i\|$.

Thus, the search space of the problem is defined by the boundary of each dimensions and the number of dimensions. Bigger is the search space, more the solution may be long to find in term of computation and power complexity.

The goal of the EA family is to optimize inputs ($\vec{x}$ in order to have the best solution possible for the problem. Where the best solution is defined by a cost function depending than the input a vector $f(\vec{x})$.

$$\max f(\vec{x}) \leq f(\vec{X}) \tag{3.2}$$

Where $\vec{X}$ is the global optimum solution.
The cost function $f(\vec{x})$ is unique for each problem and have to be redesigned for each problematic. Depending on the problem, the global solution is unknown and the best is to tends to this supposed global optimum solution. Optimize a solution $\vec{x}$ is not always enough to solve efficiently the problem. The solution proposed have to respect the constraint linked to the problem. The constraint can be various depending than the problem. As example one naive constraint is the boundary of the search space. To reach a set of m constrain E must be taken in consideration.

$$\begin{gathered} \min(\sum_{j=1}^{m} e_j(\vec{x})) = E \\ \max F(\vec{x}) \forall \vec{x} \in E \end{gathered} \tag{3.3}$$

F is the final cost function, with include the constraints, to evaluate a solution $\vec{x}$ to optimize the problem.
The EA manage the optimization of the problem based on the cost function $F(\vec{x})$ by applying different meta-heuristic. The meta-heuristic use different methodologies to optimize an initial solution. The optimisation is more or less global depending than the method chosen. Mainly the optimization methods are based on the generate new sets of solutions. The new sets is made by evolving the previous sets of solutions. A set of solutions is also called population. Where a population is defined as $pop = \{\vec{x_1}, ..., \vec{x_p}\}$ with p is the number of individual in the population. The solution found by the EA is not mandatory the global optimum and for some type of problem (as Np-hard) it is impossible to confirm it.
The risk in this case is to try to optimize indefinitely. Instead to control the end of the optimization a stopping criteria need to be taken in account.

3.2.3/ Stopping criteria and convergence

The EA does not provide a solution to decide when is necessaries to stop the optimisation. To control the stopping criteria different solutions exist. The solutions proposed in the following part are mostly adapted to the algorithm like GA, PSO, mimetic and other EA working with a population to optimize.

The EA algorithms are mostly efficient in the problems with many local minima. Due to numbers and size of the local minima it can be difficult to assure if a solution is the best or more exactly the global optimum. To know if the global optimal is reached, the method must be sure which no other solution can be the better. Therefore the solution found must be always exactly the same or equivalent in term of cost. The optimisation process must be reproducible (same input gives the same output).
Only the deterministic method can insure to have a global optimum solution as a convex problem. The convex problem has only one global optimal and no local minima.
In this case, the EA looking for the better solution possible (not the optimum). That can

be in some case the same then the global optimum but because of the uncertainty is impossible to call global optimal it is just the better solution founded.

Based on that, how determine when is time to stop the optimisation with EA. At some point is useless to continue the optimization (best solution founded or lock in some deep local optimum) and a stopping criteria should be defined.
Three possible way are communally used :

— The first method (called fix time criteria), is to stop the optimization after a fix numbers of iteration or time limit. The limit is measured in term of time computing, also the numbers of iteration must be fixed by the user. The interest of this method is manly to control the time of computation for the problem requiring a solution in a determined time (as a real time). The risk of this method is to stop the optimization before to have an efficient solution.
  This solution can be mixed with other method to reduce the number of useless iteration.
— The second method (called update criteria), is to stop the optimization (before the convergence) if no better solution is found after a predefined number of iteration. This criteria can be useful for the complex problem or if many solution can have the same quality. The advantage of this solution is to can stop before the convergence with a close solution. The inconvenient is to stop too early, in the beginning of the optimisation due to a good initial solution.
  To use this stopping criteria a correct number of iteration has to be selected. The number of iterations must be sufficient to give time when the meta-heuristic is lock local minima. A long time lock in local minima may append mostly at the early time of the optimisation due to the too good initialisation or in the late optimisation time (when the solution is already well optimized).
  In contrary a stopping criteria with a too big number of iteration became use less. In the worst case, the convergence point will be reached before the too big number of iterations.
— The third method, (convergence criteria), is to stop the optimization by waiting the convergence point. The convergence is reach when the actual population is composed by a set of solutions identical. That means the same solution has been founded by all the individuals of the population.
  The best solution found push the other to evolve in the same direction, by contagion all the individual of the population evolves to reach the convergence point.
  This solution found during the optimisation process is supposed to be the better. In this case, the population has been converge to an optimized solution. At least no better solution can be found during this optimization.

This 3 criteria are the more commune, but depending the problem other criteria can be found with more or less a priori on the problem.

Moreover the stopping criteria presented can be a combined to have an efficient and flexible solution as in the following example :
The mixed stopping criteria is to combine a fix time criteria and the update criteria. The advantage of the fix time criteria is to avoid the case of an almost infinite optimisation loop append, due to an impossible convergence.

Combined with the update criteria the other advantage is to can stop the optimisation before the convergence and before to reach the time criteria limit.
The combination of these stopping criteria assures to have always a solution optimize in a reasonable time (fast, efficient and time predictable for the worst case).

3.3/ GENETIC ALGORITHMS

Among the evolutionary algorithms one of those was very close to the Darwin theory by reusing the operating principle of natural selection and was also based on the genetic with the influence of the crossover and mutation (see section 3.1.2). This algorithm is calling Genetic Algorithm (GA) and was introduced for the first time by Holland in 1962 [**?**].
The genetic algorithm (GA) is from the EA family but is also one of the fundamental algorithms of the EA. The GA became popular at the late 80s and early 90s, particularly with Goldberg works [27]. Many details were redefined and explored in the knowledge of genetics to have a huge set-up and operator available for the GA.

In this section, the GA will be present with more interesting advance and set-up. The following section will try to list the more interesting aspects of the GA mechanisms.

The explanation will be separated in 6 sections with :

1) chromosome representation
2) population size
3) cost function
4) selection mode
5) operator
6) setting

Before beginning, it is important to remember the GA is an algorithm use to optimize a solution while still on a non-deterministic initiative and therefore can not give the certitude to have the optimal solution.

3.3.1/ CHROMOSOMES

As in the Biologic field the chromosomes contain the properties (with the genomes and DNA) of the individual. A primordial issue when you want to optimize a problem involving the GA is to define properly the chromosomes role, for that different aspects must be taken into account.

The first aspect is related to the problem himself. The chromosome is used to design the problem and it has to represent a solution. To do so, it is important to know the problem and identify clearly what parties of the problem need to be optimized and what is the range of the research area.
Depending on that the coding can be direct or indirect.

- The direct coding is code with the genomes corresponding to the elements of the solution. Using the direct coding may simplify the output of the optimization by returning it back to an element directly proper to use.
- In Contrary, the indirect coding, it is not directly proper to use and need conversion to be used. One example of indirect coding is the willingness to introduce

redundant genome inside each chromosomes. The conversion must be done by a heuristic. The interest of the method is to be able to make a strong constraint adapted to the problem as [60, 75] where it is used to solve a complex scheduling problem with many constraints.

Inside this article the interest between the direct and indirect coding are presented.
The diverse aspects of the problem should be clearly defined before the element makes up the chromosome, like the number of dimension to optimize, the boundary of each dimensions and the importance of the dimension order. When the chromosome is defined the second phase is to choose the best coding solution. Many solutions exist to encode the chromosome like presented in [75] and [55] but among the coding solution, 4 main categories can be considered as basic coding type, which are the combinatory coding, binary, the real (also alphabet) and tree coding.

Binary Coding ([81, 49, 26, 55, 75])
The binary coding offers to format the chromosomes as a bit string in which every genome of the chromosome can be covert on pack of bit with can have only 2 value 0 or 1. This coding method was studied since the beginning of the GA by Goldberg et al [26] and also used in other EA like PSO in [51]. The binary coding is more efficient in the small search space or when the size of the chromosome is not too long. The advantage of the binary coding is the possibility to introduce lot of variety during the process of optimization [81]. The variety is traditionally from the mutation but in the case of the binary coding the crossover introduce variety by the potential split of the genome in 2 pack of bit.

The combinatory ([62, 18])
The combinatory coding is commonly applied in some specific problems where the goal is to order all the element. In this case, the position of the genome in the chromosome is primordial as in [18]. It is characteristically used to the problem as TSP [62] (Travelling salesman problem). When the combinatory coding is used for problem the aim is to optimize the order the element of the chromosome. With a combinatory coding the each chromosome already has all genome of the answer. An example, the TSP the aim is to order various cities (in the problem of TSP every genome represents a city) and all the possible cities are included in the initial solution (chromosome). In this case, the problem is to optimize the combination of the element composed by the initial solution. One evolution of the combinatory traditional coding is to can add and remove some genome, that obviously affect the size of the chromosome, for example when the goal is to find the shortest distance in the tree [43].

Real Coding ([81, 55, 75])
Real coding or integer coding is considering every genome of the chromosome as a number to optimize. This number can be a real or an integer and may have an infinity of possibilities in the negative or positive. In fact, the value does not have an infinity of possibilities because the constraint by the computer and limits of the problem itself too (size of the search space). This coding is used when the search space is large and also can be efficient when many dimensions need to be optimized. But most of the time a

special attention should be put to the operator, because in many cases the operator may be adapted or redesigned depending on the problem such as in [52], which the operators are adapted to look for close neighbours.

Tree coding ([43, 55, 75]) The tree coding use the tree representation to take care of the hierarchy but this method is not really popular and not flexible to any case. The advantage of tree coding is this ability to go farer then a combinatory representation. An example in [75], the tree coding is used with the GA to optimize new network telephone or gas/ water pipeline where the relation between the element are primordial. In [75] present the interest of tree coding for the intrusion detection system.
The 4 coding method presented are not the only potential coding, there are the more commune and the roots of other coding and many other have been developed and studied with direct or indirect coding to feet with specific problem. Thereby in the literature a survey are dedicated to the encoding chromosome for the GA [60]. The survey [60] propose to explain and find a robust coding usable depending on the problems.

3.3.2/ Cost function

The cost function or some time called fitness function has an essential role in the optimization process. The aim of this function is tantamount to quantify the quality of one solution. This point is primary to the GA and in most of the optimization process using meta-heuristic. The cost Function is an compass the meta-heuristic during the optimisation. The cost function is dependent then the problem and should be design or redesign depending on each specific problem. Once the cost function is designed for a problem they can be used to test different other algorithms of optimisation with requires also a cost function. Because the cost function is exactly the same that becomes easier to compare the results from different algorithms as is discussed in [21] and also in the chapter 5.2.Once conceived the cost function is considered as a black box by the optimization algorithm and encloses most of the complexity of the problem. Obviously if the cost function is not designed correctly with all the constraints and the objective, the optimization will fail.

Multi objectives

The cost functions are traditionally made-up for problems with only one objective, but in the recent years several solutions have been adapted for multi objective. The goal is to optimize a problem with few sub-objectives included. These sub-objectives can be at some point contradictory and a trade off must be done during the optimisation process, based on the rating made by the cost function.
The cost function for multi objective problem (MOP) is discussed in the survey of Zhou et al [90]. In Zhou et al [90] one of the ways to solve the MOP is to adapt a classic mono objective algorithm in multi by customizing the cost function. The customization of the cost function propose a way to evaluate a solution not depending than one objective but with all of them combined in the same function instead of several cost function. Also other solutions are discussed in [90] like using coefficients for order the objective priority

or by reducing the problem into several sub-problems.

Constraints

The goal is to satisfy the objective(s) while taking into account the constraints. Consequently the cost function has to take care and integrate the constraints. Previously in the chromosome (3.3.1) the strong constraint was established by the coding, especially by imposing limits in the real coding or using indirect coding, but it is feasible to impose some soft constraints in addition to the system by adding the rule in cost function. The rule corresponding to the constraint is helping the optimization to do the good choice not by imposing strong constraint but by affecting ”bad points” or ”good points” depending on the constraints.
The soft constraints can appear a bit useless, but there can be a good trade-off between two contradictory objectives and some other strong constraint, also a soft constraint can be easier to implement and faster in term of time computation compared than a hard constraint.

Optimisation and time computation

The cost function should be designed carefully and have to pay special attention to the time computing. Indeed the hight frequency call of the function may generate some heavy slowdown.
The cost function has a special importance in the optimization process to the rating of all the individuals at every generation. If we consider a GA with 100 individuals (it is a common number of individuals not to much and generally enough) and a convergence after 100 generation (it is good minimum in order to avoid a premature convergence), in this case the cost function will be call at minima 10 000 time. Due to this important factor it is primordial to carefully design the cost function as is specified in [3].
It is common to have several thousands or billion calls of the cost function during the optimisation process. Hence the importance to have a function timeliness and economic in resource.
Even the cost of this function must be the most accurate, in some cases like in [49], a complex calculation or noise can affect the reliability of the cost function which will be impacted on the quality of optimized solution but despite the noise and the weakness of the cost function the GA can optimize and give a solution. But it is advisable to have a function accurate and reliable as possible to have an efficient solution.
To conclude with, the importance of the cost function is a major piece of the GA and the choice of the design of cost function associate to the chromosome representation will affect the result but also the setup of the GA and can generate some lock.

3.3.3/ Population

The population gathers a number N of solutions (or individuals) from the same generation. The individual can be represented into one or more chromosomes(instead to have redundancy as in [60]). Commonly each individual is composed by one chromosome.

Therefore the two terms are regularly inverted in the literature.
At every generation most or all the population is renewed (by using the selection and operator). Indeed the population can have an effect on the convergence and the result of the EA and different strategy or set up exist. About the population 2 main points need to be studied : the first is the initial population and the 2nd is the size of the population.

Initialisation of the population Initialization of the population is one of the fundamental questions may affect the convergence of the problems. There are mainly 2 common proposed solutions with one using the full random generation or using efficient and already approved heuristic.

The method is based on heuristic involved a perfect knowledge of the problem and can not be applied for all the problem. The advantage of using heuristic, it can give very good starting individuals at the beginning the optimizations and also used the heuristic may give a solution more respectful of the constraint of the problem, obtusely if the problem include many constraints hard to satisfy.
Although this advantage of using a heuristic to find the 1st generation of the population, can become a handicap and push the GA in the direction of the potential local minima. Indeed using one heuristic to build the first generation can have a population too similar with not enough variety. The variety is essential to run through all the searched space and allows do not converge too fast in a local minima (see [45]).

If using a heuristic to build the initial population is not always the good solution one other solution more versatile is to use the randomness to find initial population. The randomness generates each individual randomly in the search space. One of the advantages is the individual can be well distributed around the search space to cover most of it if the size of the population is big enough. The random distribution permits the algorithm to cover a wide part of the search space quickly during the first generation and the spreading of the population is a good source of variety. The random initialisation is commonly used and less often the heuristic solution.

To initialise the population, the third method is to combine the full random with the one based on heuristic. In this case, for examples a random solution is applied before, the heuristic is used to refine the initial solution. Otherwise different heuristics are used randomly to generate all individuals of 1st the population. Other combination are possible see [43] for more detailed see.

Population size An other key point is to consider the size of the population. The size of the population and this effect on the convergence are studies in many articles [45, 3, 28, 64, 26, 10]. The first point is to find the appropriated size depending on the problem. Indeed if the size of the population is too small the variety of the population can be too short and the algorithm can converge too fast [3], instead if the population is too large the waiting time may become too long. Like that, chose the appropriate the size of the population is not trivial at all. To find the best size of a static population only one way, is to do several experiments and compare the results, like did it in [28].

The population can also be adapted dynamically or auto-adapt the size of the population during the optimization. Commonly the number individuals in the population need to be important during the first generation but close to the convergence the population can be reduced for win time. In [61], the size is fixed by probability. It can also be fixed by a linear equation or even more complex, in function of the progress of the cost function and the variety of solution see in [3]. The population have an important part in the convergence computation, depending on if the size of the population is static, dynamic or auto adapted the convergence can be faster with more or less quality for the solution. The convergence is studied in the [10]. But the convergence is not only link by the population size but also the selection, the coding and the operator choice are an important aspect of the GA like is showed in [28, 64].

3.3.4/ Selection mode

What is the selection mode ?
The selection mode is the method used to select in a population the individuals the most able to reproduce. It is an important key point corresponding to the natural selection in the Darwin theories.
The choice of the selection mode is primordial and affect greatly the quality and the speed of the optimisation process. This choice needs to be done depending than the problem. A selection mode applied on a specific problem can be efficient (in term of answer quality and time convergence) but the same selection mode applied on a completely different problem can be inefficient for it. The selection mode must be selected or adapted for each kind of problem. To choose the selection mode no magic bullet, the testing of different method has to be done.
One of the objective of the selection mode is to keep enough variety in the population to avoid an untimely convergence. Also too much variety in the population and especially not a the beginning may artificially delaying the convergence. A good selection mode has to trade off between too elitist selection (not enough variety) and a too permissive selection (too much variety).
A multitude of selection mode has been developed during the time (few of them have been listed in [55]). Make a choice among this wide list of possibility is difficult. The selection mode among the most common or representative is presented :

Elitist selection - The elitist selection is more like a subgroup of selection mode. His particularity is to use a deterministic way to selected the best individual depending of the cost function the best or some of the best are selected directly for the next generation like that the best individual are preserved and no risk to lose one good individual during the crossover, mutation and other operation. The few best individuals selected are used to engender a new generation. This subgroup of selection is studies as in [15, 45] to estimate the efficiency of convergence using this selection mode or in [67] applied in the multi-objective problem. It is appeared on this article the elitist selection is efficient, converge quickly and the best individual founded are preserved for the next generation. But unseemly it may sometime converged prematurely most of the time because of this difficulty to keep enough variety wish this selection mode for that some of the elitist selection have been customized to try to preserve the variety.

The roulette wheel - The roulette wheel selection is at the same time one of the older selection mode used since 1989 but also one of most inspiring. The roulette wheel gives many methods inspired by this one like for example remainder stochastic sampling in [80]. The roulette wheel selection had a basic operating. Every chromosome is represented in the wheel and the size of the wedge are depended them the quality of the chromosome. This quality is computed from the fitness function. With this technique, the chromosome with the best fitness function has the most luck (but not necessarily) to be selected. Once the wheel was built by the random sampling can begin to select the new individual of the generation. The wheel turns until all the individuals are selected. This method helps the better chromosomes to be more represented in the next generation but also accepted to have some time more or less bad individual conserve for keeping the variety. More the GA work more the size on the wheel of the best solution will increase and help to converge.

Tournament selection - Tournament selection is one of the most used in this last decade. It is working as a tournament. The first step is to create few pools with all the individuals of the actual generation. The pools are randomly create. When the pools are created, the tournament can begin and the best chromosome of each pool (depending to the cost function) are selected for build the next generation, with the other winner. This selection by randomly build the pool, help some not good individual to continue but the best chromosomes are always used to build the next generation. The tournament selection is very efficient to keep diversity and also give a chance to have a fast convergence as is explain in [45].
Obviously the size of the pool has a really big influence on the convergence as is studied in ,[45, 49]. The conclusion of these papers about the pool size is, bigger is the pool (4 or 5 individuals) less diversity is kipped and the convergence goes faster, and can finish lock in a locale minima due to a premature convergence. In the other side the small pool (2 chromosomes) keeps the variety but the convergence can be slower. Finally the size of the pool is also one other parameter to take care and the size of the pool is also strongly linked to the population size. Among the selection mode presented tournament selection is one of the most efficient for manage the variety and that explain this wide popularity to solve an engineering problem.

The last element to consider about the selection mode is when it is the more appropriate to use it ? Indeed the selection of the individual may intervene at 2 occasions (see figure 3.3). The first the more conventional is to select the parent able to reproduce. The second is to select the children good enough for be part of the next generation. The interest of this is to be able to generate new children until the selection criteria are rich in order to have a population with acceptable children in sufficient quantity (This method is more efficient on the problem with lot of hard constraint).

3.3.5/ Operators

The operator have to aim the design of the next generation by generate the offspring. Once the parent able to the reproduction were selected (using the selection mode saw previously) it is time to engender a new children. The techniques, to create a new po-

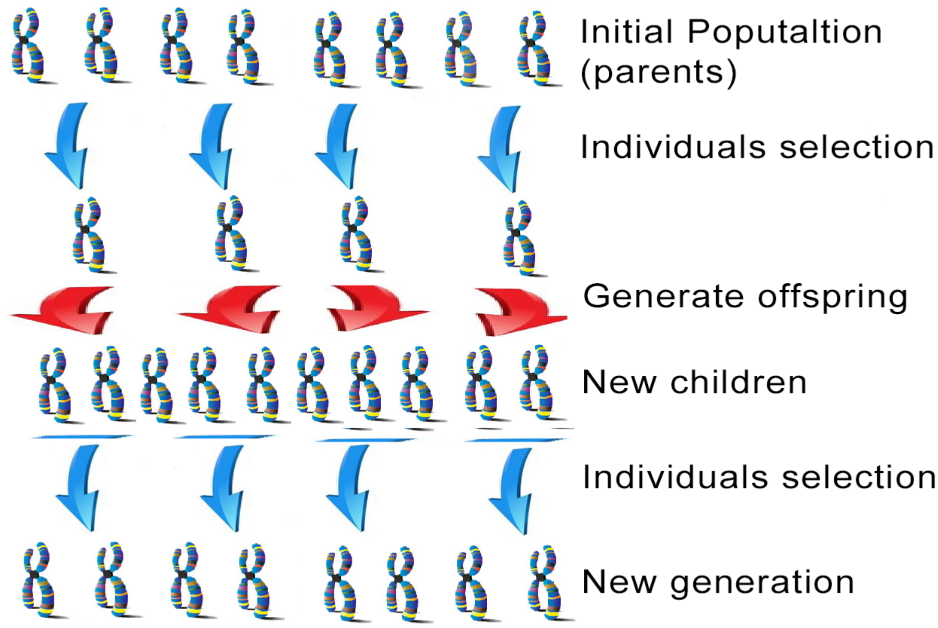


FIGURE 3.3 – The 2 time for the selection mode with the GA mechanisme

pulation are numerous. Among them the more common are the heritage (or selection), random generation, the crossover and mutation.

heritage or selection : The heritage or also called selection is a simple copy of the best parents to the next generation with no modification. This operator is commonly used to keep the best individual in order to do not lose the best solution if no upgrade is made by the other children. Indeed the other operator may propose degenerated children with is some time worst then their parents and the heritage permit to conserve as is the individuals.
The crossover and mutation have a more interesting mechanism. They are also the basis of many other customized operators and the understanding of the basic crossover and mutation required to use it.

Crossover : The crossover operator is directly inspired by the biologies. As 2 mammals reproduce to have progeny. Half of the genetic material of the two parents are used to create a child. The crossover operator is mixing 2 individuals to create a new child. The aim of this is to merge 2 workable solutions to have one other solution potentially a bit better. To merge the 2 individuals it is existing many way like studied in [43].

Mutation The mutation operator have to aim to add diversity by mutate some random allele of the chromosome. This mutation must be randomly choose and are useful to keep the diversity of the population. As the crossover different mutation are existing [43]. The mutation mechanism affect only rare genome in all the population. The genome mutated is randomly selected and the is will be based on random.

Customized operators This 2 principals operators need to be choose carefully and in most of the case redesigned. The redesign of the operator is essential in many occasion. One of the first reason to redesign part of the operator is to fit well to the problem. A example is the mutation redesigned to explore the search space with a logic of close neighbour as in [52].
The second reason is link to the chromosome coding. Depending on the chromosome coding chosen (binary, combinatorics, real, ...) the operator have to be adapted. An example is to modify the mutation and crossover to preserve the genomes in the real coding. The same operator can not be used for combinatoric coding or binary.
The third is more rare but in some case the operator have to be adapted depending on the selection mode chosen. An example is given in [71] with the crossover and the mutation for a elitist selection.
Also one other reason to redesign the operators is to have operator fitting to some of the hard constraint to the problem. In order to have operator able to create children respectful to some hard constraint of the problems.

Operators rate An other important element to take in consideration after the choice of the operator and their implementation is rate of each of then. The rate of an operator correspond to the usage percentage of the operator on the chromosome, higher the rate is, more this operator will be used at each generation generation. In the mutation the rate can be globally understood as a chance to one an genome to be muted. Finding the best rate for every operators became a real challenge. The best solution to find the appropriate operators and their associate rate for a specific problem no other choice to try the couple combination of operators with different rate like in [81, 28, 61].

To conclude on the operators, they are an important factor to evolve generation after generation. The operator have to keep the diversity in order to have an efficient converge (not premature and not to late). The question of the diversity introduce by the operator was been studied in [62, 49] [43]. It is appearing one good static configuration to keep diversity [45] of chromosome is to have crossover mutation with height rate of crossover and small rate of mutation. But the rate of the operator can be adapted depending on the searched space, the convergence and other element. Some research was done on adapted the dynamically the rate of the operator depending too many external factor or using probability like is discussed in [18, 61, 69]

3.3.6/ Setting and set-up

The previous section introduce the different aspect of the GA and give the key to understand the different element and the mechanism of the genetic algorithm. Besides the GA explication is appear many primordial choice to set-up properly the algorithm. Part of their choice are interdependent and the connection between the different parameter can make the set-up tricky. Also the GA has been studied for decades and many variant were developed over the time to make the GA more efficient. That give even more choice but no general set-up have been formulated. To evaluate the performance of a set-up the quality of this answer is used but also the speed of convergence in term of number of generation, and also the variety of the chromosome at each generation until the convergence. The variety is one of the factor useful to explain the convergence speed and the answer

Inspiration or group	Algorithm
Coding chromosome :	what coding choice ? (binary, combinatoric, real,...)
Cost function.	How quantify the answer quality ?
Population :	What size ?
	What initialisation ? (random or heuristic)
Selection mode	what choice of selection mode ?
	And depending on the mode chosen what set-up ?
	As for tournament selection the size pool, the wheel repartition for roulette wheel or the number of parent selected for elitist...
operators	What operators to use ?
	What implementation choice (customized operator or not) ?
	What rate for each operators ?
Stopping criteria	What stopping criteria to use ?
	If is not by convergence what are the boundary ?

TABLE 3.2 – Sumarizing the question to ask to configure the GA

quality. The variety is almost opposite of the convergence. It is when the chromosome are all very different in the same generation but also generation after generation. A high variety is ideal at the beginning because this allows the optimization browse the search space and potentially help to jump the local minima. As example if the finally solution is lock in a local minima after a to fast convergence that mean not enough variety has been introduce during the optimisation.

During the choice of the ideal parameters for GA the variety is a important element to preserved and more at the beginning of the optimisation to browse the search space.

However the GA stay complex to configure because of all this parameter to have a good answer quality in a reasonable period of convergence. Many aspect need to be adapted depending to the problem, constraint, size of the search space and . . . . Using the simple GA that mean configure a set of parameter. The parameter can be formalized as a vector like in [28]. This formulation is especially efficient test numerous setting. To set-up properly a GA few question need to be posed as in the table 3.2 and few setting must be test as in [81, 28, 61].

3.4/ GA TRENDS

Whether GA is a relatively recent algorithm, it was largely studied during many years and has progressed tremendously. To follow the trends of the GA the survey written in [70] for the Simple Genetic Algorithm (SGA) are a good point to understand the progress before 1994.

In this article [70] the author begins to explain the SGA and the significance of the natural selection with the possible modification to adduce. Also the GA improvement in term of performance is discussed. The SGA is parametrizable depending on the implementation. That give a big importance of the problem formulation and the consequences on the

solution. This survey is relatively hold (from 1994) and other more recent are rather focused on the Multi Objective Evolutionary Algorithms (MOEA), which include many different shapes of Genetic algorithms customized to satisfy the multi objective problems [90].
Although the papers are concerned about the multi objectives and many references are made to highlight the recent advance on this field with different types of adaptation. The evolutionary algorithm, like the multi objectives evolutionary algorithm decomposition (MOEA/D) [**?**]. To decomposed the problem into sub-problems and each sub-problems are weighted by the neighbouring relation between the sub-problems then aggregated.

It exists many other MOEA present in this paper as Non-dominated Norting GA II (NSGA-II) [15] this algorithm are using an elitist selection to optimize efficiently the problem without having to sort the different solution depending to the different objective. Some other MOEA as QGA for Quantum-inspired GA [15, 29, 30], Non-dominated Sorting GA (NSGA) or BMPGA for Bi-objective Multi Population GA [191]@, are examined on this survey[15].
The GA have been studied for different objective and optimization problem. these surveys give a fast view of the GA formulation and specific customization for GA application field, as the following example.

- In [63] are interested on the problem of clustering and use GA for have a non-supervised clustering and also show the different implementation and customization of GA adapted to this problem.
- In [79], the genetic algorithm is applied to the problem of pattern recognition. This problem is a complex multi optimization problem. The GA is used to optimize the classification, the training and the research of a set of efficient features.
- In [55], the GA is applied to security problems to control computer access in the network and prevent attacks. In this case the GA can be used to optimize the classification of the access and this way detects the legal and authorized access then the hacking attempt.

GA have been well studied and the literature about is vast. These last decade the GA has been used for multi objective problem but its popularity has decreased in favour of algorithm which requires less configuration or other algorithm more oriented on learning.

4

PROBLEM MODELISATION

Sommaire

The problem formulation has to translate the objective with all this problematic into a simple formulation usable for different optimization algorithms. The objective here is to find the position for a set of cameras or waypoints. The position of this set of waypoints has to be optimized in order to cover most of the area. The area may be a vast and complex zone. A good formulation is essential to design an efficient cost function. The cost function is used to quantifies the quality of the solutions. It is a crucial element for the optimisation processes.
This chapter present a formal definition of the problem based on the literature and our proposed formulation. The formulation proposed is adapted to optimize the problems with evolutionary algorithm to have an efficient cameras position for maximizing the coverage, depending then many constraint as using a camera mounted on UAV.

The following section is focused on how to estimate the covered area depending on the cameras parameters.

To estimate efficiently the area covered by a given set of cameras some point have to be clearly defined :

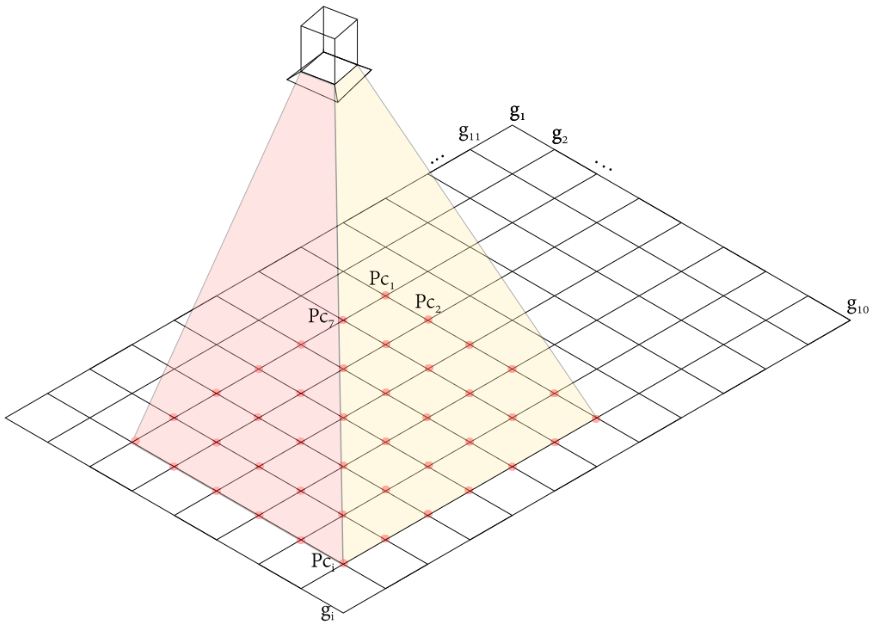


FIGURE 4.1 – Camera projection onto a grid. The grid G is placed on the floor to discretize the area covered with numerous grid point g_i. The point cover by the camera Pc are noted in red

— The area himself. How to represent the area.
— The camera definition.
— The constraints added to the systems.

This 3 parts are discussed in the following sections. The area definition is discussed in the section 4.1 dedicate to the grid design. The camera definition is discussed in the section 4.2 and the third part discuss about the constraints in the section 4.3.1.
Finally when the problem is clearly defined all the different elements are integrated to have an efficient cost function usable for the optimisation process in 4.3.3.

4.1/ THE MAP

The first part is tantamount to estimate properly the area to cover. To do so many methods have been developed most of them are based on an occupation grid G of the area. The occupation grid is a sample discretization of the area with numerous points.

$$G = \left[g_1 ... g_i ... g_m\right], m \in \mathbb{N} \tag{4.1}$$

Where m is equal to the number of points in the grid. The occupation grid is placed on the area to cover. At minima each point g_i of the grid G should be covered by a camera (as in Figure 4.1). Consequently a list of points is set up to enumerate the covered part of G which are noted as Pc.

$$g_i \in Pc \text{ IFF } g_i \text{ is coverd.} \tag{4.2}$$

The design of the grid is an important element of the problem formulation. Different solution has been proposed during the time with different advantage depending on the situation.

4.1.1/ How to design a grid map

The following subsection is focused on the different modelling the grid possibility, based on the literature.

4.1.1.1/ Sampling frequency

The grid map is used to discretize the area to cover. The discretization of the area can vary and the area can get a high level of discretization or low level. The level of sampling frequency has a bearing on the problem formulation and moreover on the optimization.

High sampling frequency

The high level of discretization or high sampling frequency has some advantage and disadvantage.

The high sampling frequency of an area is characterized by a big amount of point g_i for describing the area. The big amount mean to have an important density of point g_i and consequently the value of m is high.

The advantage of it, is to have a better estimation of the coverage. More the area is finely discretize more the estimation of the coverage will be sharp. In [34] an example of high frequency sampling is given in order to have sharp estimation of the area.
The high sampling frequency allows the cameras position to be much more accurate and make a very small adjustment.

On another side, the disadvantage to have a too high sampling frequency is the time computation. Rather to refine the solution the too high level of discretization of the area will make the optimization too long and more complex. Indeed to control the coverage, it is necessary to control if each point of the grid is covered by a camera. That mean the number of the test to estimate the coverage of each point of the grid, for a set of cameras is $m \times n$. Where m is the number of points in the grid and n the number of camera.
More the size of the grid is high more unity test of coverage (see in section 4.2) must be done, and that at each step of the optimization process. Consequently the size of the grid will greatly affect the time computation.
Also the high sampling frequency will affect the positioning of the cameras pose. In fact, more area is finely represented more freedom has to be done to pose and adjust each camera.

Low sampling frequency.

At the opposite a lower sampling frequency can be a good solution to upgrade the convergence speed of the optimization process as that was presented in [91]. In Zhou et

<table>
<tr><th></th><th>Advantage</th><th>Disadvantage</th></tr>
<tr><td rowspan="2">High sampling frequency</td><td>Best estimation of the area to cover</td><td rowspan="2">Time consuming</td></tr>
<tr><td>Give more precision on the cameras poses</td></tr>
<tr><td>Low sampling frequency</td><td>Faster computation</td><td>Bad coverage estimation</td></tr>
</table>

TABLE 4.1 – Sum-up of the low and high sampling frequancy adaventage or disadvantage.

al [91] a small value of m is chosen to have a real time solution for small area and just few cameras (up to 20). In this other side, the low sampling frequency may generate a bad estimation of the area covered due to the too low density of points in the grid. The low density of points may give an approximated view of the area covered and some black hole can appear between the point of the grid. This black hole can be too small to be detected because of the low density of point. In this case, the optimization cannot take in account this black hole and the solution given after an optimisation will be in a real environment not good as aspect.

Low or high sampling frequency

Finally the too low sampling frequency, instead to win computation time, may affect the quality of coverage estimation and consequently the answer. But, as explained the impact of a too high frequency in the solution has also some consequence as is summarized in the table 4.1
The density of the grid has to be adjusted depending than the goal and the precision required. One of the solutions proposed in Zhao et al [87] is to have a progressive refinement by increasing the grid density.

Zhao et al [87] despite a low density of points at the beginning, the number of points are increased slowly to have a better density and refine results. Also the increasing point frequency is applied to refine the solution and add more cameras at each step of the optimisation to avoid the black hole and has a better coverage.

Camera pose precision. The frequency sampling is often linked to the pose precision of the camera as discussed previously. Therefore increase the number of points will also increase in proportion the number of positions possible for each cameras. More the area is finely defined more is necessary to can slightly adjust the cameras positions. Consequently the possible camera position and the search space are increased to finally allow a more refine solution but also more complex to optimize.

4.1.1.2/ DISTRIBUTION

The distribution is an important factor to manage to design an occupation grid. The distribution is how the points of the grid are placed on the area to cover. Different distribution

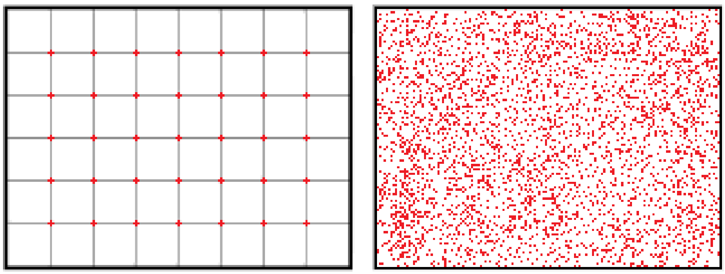
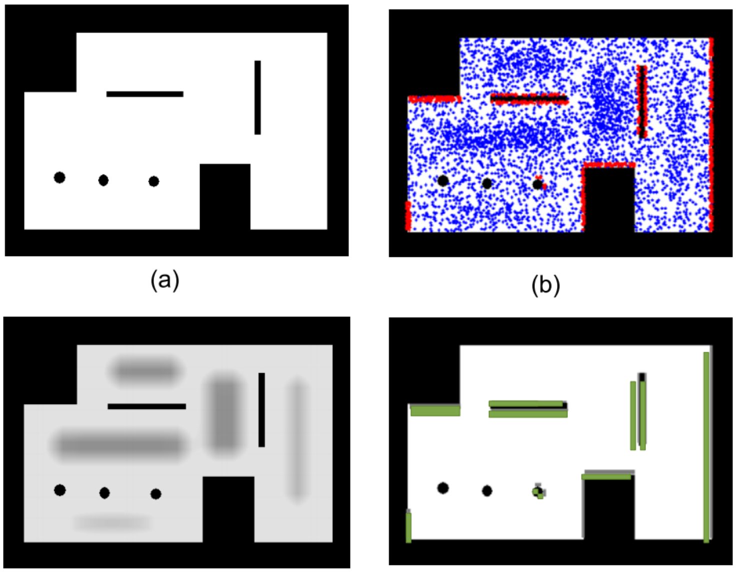


FIGURE 4.2 – (a)Grid with uniforme and regular distribution.
(b) Random distribution.



(a) (b) (c) (d)

FIGURE 4.3 – (a)area to cover.
(c) Importance of the area.
(b) grid representation.

can be used, but commonly in the literature the grid pattern distribution and in a lesser extent the random distribution is applied (see Figure 4.2).

In the [74, 34] the random distribution is used to describe the area to cover.
In Hoster et al [34] the point of the grid are randomly distributed to describe all the area to cover. The advantage of this article is to use the random distribution in order to manage the density of points in some specific region of the area. Especially, by increasing the density of points on some specific zones of the area. The increased density allocates more importance to these zones (see Figure 4.3).
Indeed the higher density will affect the optimization process. The area with more density will be comparatively more profitable to a low density area. In these cases, the zones with

high density are covered in priority. This mechanism is even simplified due to the random distribution.

The hengel et al [74] have tested the random distribution and the uniform grid pattern distribution before to conclude the grid pattern and the random distribution proposes globally the same result, when there is no priority zone in the area. Based on this observation hengel et al [74] decide to use the uniform grid notably because of its simplicity of implementation.

The "random distribution" is less popular but in [87] a hybrid distribution is proposed. The idea proposed is to reduce the number of points in the uniform grid when it has a to hight density. To reduce the number of points in a random selection is used.

4.1.1.3/ SPECIAL MODELLING (3D OR 2D)

To design the occupation grid, one other element should be taken in consideration. It is the spacial modelling. After deciding the useful density and the distribution of the grid, the position in the space of the point g_i need to be studied. In fact depending on the problem the grid can be modelled differently in order to properly cover the area to control. Commonly the occupation grid is placed on the floor and calculate visibility only in 2D by computing the camera projection as in [73, 11, 91, 84, 34, 87], but depending on the context the grid have interest do describe a 3D space. Hegle et al[74], calculates the visibility, where it is relevant : for example, on the upper torso or head of the possible target rather than the floor. This article proposes a 2D grid but inside the 3D space in order to characterize properly the volume by placing the grid at a specific height.

In [12] the grid is formalized in the full volumetric space by numerous "control points" (as g_i) to control the area. The points of the grid is uniformly (or can be randomly) distributed along the axis of x, y and z. Also in [51] the grid are formulated in the volumetric space by using a uniform 3D grid distribution. The formulation proposed show the complexity even more important to use a 3 dimensional occupation grid and for practical reasons (mostly the inadequacy has computational power due to the increased complexity) the 3D grid is replaced by a 2D grid at a specific height to optimize the coverage.

While an occupation grid designed to estimate the volume cover along the x, y and z axes of the area already exist. His implementation is unusual due to the increasing complexity of the optimization process. Nevertheless some solution was discussed [1, 74] to take into account the volume of the area to cover which cannot be only limited to an occupation grid along the axes x and $y$as a simple 2D grid.

In [74] is focussed on estimating the area to cover inside a 3 levels of a building. To estimate the coverage in the building the grid was placed at each level. This solution is efficient in order to limit the number of points compares then a full volumetric description of the area. Also this design allows to keep the 3 dimensional information by adding a layer at each level of the building.

In [1] propose a 2D grid adapted to the relief of the area. This article are focused on covering a large outside area with an important relief. To estimate properly the area to cover a grid has been placed following the altitude of the relief as showed in Figure4.4.

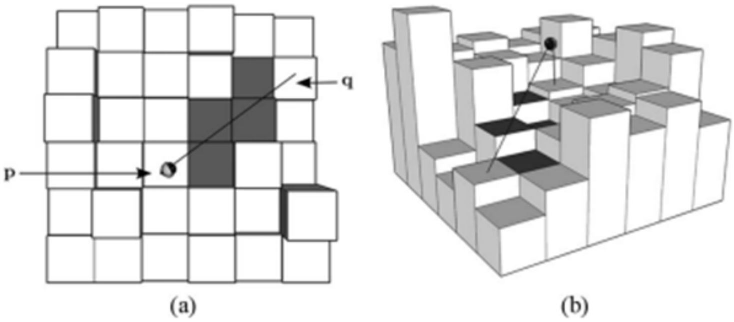


FIGURE 4.4 – Relief grid use to discretize an area with taking in account the relief.

4.1.1.4/ ZONES OF INTEREST.

Among the area to cover some zones may have a particular interest to be covered. These zones can be discriminate, by the grid design. The zones of interest are designate by different manner depending than the goal. Manly 3 methods can be discerned.

- The multi coverage zone.
- The priority zone.
- Non-interesting zone.

The multi coverage zone The aim of the multi coverage zone is to have on a specific zone of the area controlled by numerous cameras. Multiple coverage may be called k-coverage as in [86]. Where refers k to the number of cameras mandatory to cover the zones of interest. Every points of the grid G should be covered at least by one camera and for some specific zone of the area by k cameras.
Consequently a binary list of points is created to count the covered part of G which are noted as Pc.

$$Pc_i = \begin{cases} 1, & \text{if } g_i \text{ is covered by } k_i \text{ cameras} \\ 0, & \text{otherwise} \end{cases} \quad k_i \in K \tag{4.3}$$

Where k_i is the number of cameras uses to cover the point g_i of the grid. K is the list of k_i associate to the number of points in the grid. The list K is initialized at one by default, except for the zones of interest where have to be superior to one as in [12].

The priority zones. The zones to cover in priority are used especially in the case where the number of cameras are not sufficient to fully cover the area. This priority of coverage can be expressed by different way.
In the case where the grid is uniformly distributed a weighting may be fixed on the zone of the area to cover in priority [1, 83]. This method was implemented in [1] to optimize the position of the camera on the road passing through the area to cover.
In [34] the weighting of the priority zone is made by increasing the sampling frequency of the zones thanks to random distribution as in Figure 4.3(b). Using this method the zones of interest are more dense and that push the optimization process to cover this area in

priority (more density mean more interest).
Otherwise the priority zone in the uniform grid distribution can be formulate as :

$$Pc_i = \begin{cases} 1 * p_i, & \text{if } g_i \text{ is covered and } p_i \text{ is the weight} \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

Where p_i is the weight of the point g_i on the grid G. P is the list of p_i which contain the weighting of the area associate to the points of the grid g_i.

Non-interesting zone. Non-interesting zones are the zones without interest to be covered noted U, with the set U composed of points g_i. These zones are not strongly prohibited. That mean the zones considering as non-interesting can be covered but their coverage or un-coverage has no impact on the estimation of the coverage of the area. In the case of random grid, distribution the non-interesting zones have a sampling frequency null as in [1]. For uniform grid distribution in the non-interesting zones are removed to the list Pc. This method is currently used like in [87, 84, 1, 34, 83] .

$$Pc = G - U \ , \ \ U = \{g_i | g_i \in G, g_i \text{ are the non intresting points}\} \tag{4.5}$$

Finally these methods use to design the zones of interest are not fully independent and can be associate with the same model as in [1, 34, 83]. The combination of all these zones of interest can be formulated as :

$$Pc_i = \begin{cases} 1 * p_i, & \text{if } g_i \text{ is covered by } k_i \text{ and } p_i \text{ is the weight} \\ 0, & \text{otherwise} \end{cases} \tag{4.6}$$

$$Pc = G - U \ , \ \ U = \{g_i | g_i \in G, g_i \text{ are the non intresting points}\} \tag{4.7}$$

4.1.1.5/ Atypical design

Previously the method to set-up a classical occupation grid depending on the problem has been discussed. Few other solutions more atypical have been developed. On solution coming from the field of wireless sensor network is the topologies grid. The topologies grid is clearly explained in Chakraborty et al [11] the interest of these methodologies is to reduce the number of points to cover. But in this case the number of points is not related to the resolution wished but by the sensor range. Indeed the size between the point of the grid has been defined by the size of the minimum range of the sensors. The distance of the minimum sensor range are used as nodes for a topological relation.
Another atypical solution is to develop a grid composed by rectangles. Each rectangle may have different size adapted to the obstacle in the area. A rectangle is considered as covered if most of the area of the rectangle is covered by the cameras as in [84]. This method is adapted to the area with few obstacles as is shown in the Figure 4.5.

Grid sum up

The following paragraph surmise the different way to represent the area. The area represented has to be covered by set of cameras. the grid map may be used to represent

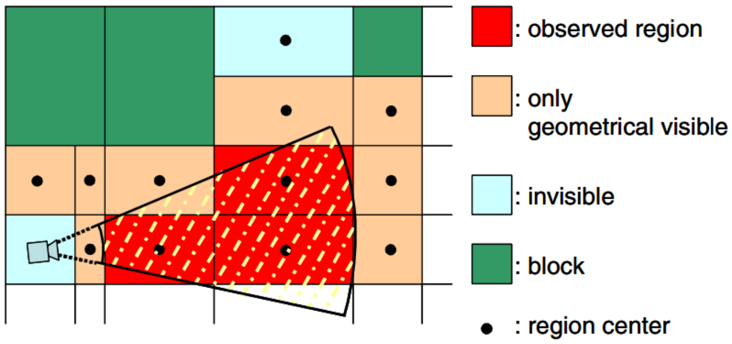


FIGURE 4.5 – Observed regions from camera candidate.

different aspect and constraint of the problems. The map representation take an crucial role in the area coverage. The design aborted previously are summarized here in this Table 4.2. This table show the more important aspect of each grid representation in different paper.

	Grid pattern	Grid random	Volumetric space	Sampling frequency	Zone of interest
[91] zhou2011	✔			For real-time	✔
[87] zhao2008	✔	✔		Incremental	✔
[12] chrysostomou2012	✔		3D grid		✔
[74] van2009	✔	✔	2D grid at each level		
[51] morsly2012	✔		Superposition of 2D grid	Adaptable	
[1] akbarzadeh2013	✔		2D grid on relief		✔
[11] chakrabarty2002	✔			Topologies sensor	
[73] valente2013	✔		overlap by shifting of z		
[84] yabuta2008	.			For zone segmentation	✔
[34] horster2006		✔			✔
[86] zhang2016					✔

TABLE 4.2 – sum-up of the grid map.

4.1.2/ OUR APPROACH

Base on the different design the one finally adopted is a grid G as in Eq 4.1 with an uniform repartition following the 2D grid pattern. The frequency adopted is fixed depending than the size of the area, the precision required and the cameras property. The frequency

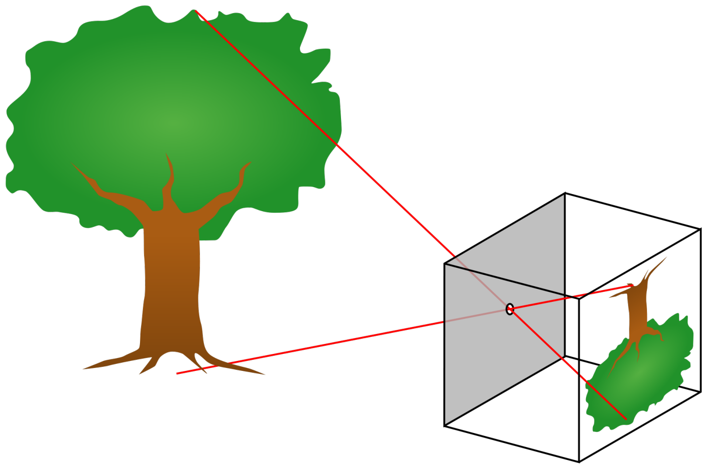


FIGURE 4.6 – Pin hole camera model.

adopted for the following part as to be considered as dense (high sampling frequency). The grid is placed on the floor of the area to control. Floor is always considered as flat without relief. The zone of interest can vary depending on the need of the experimentation, but the design chosen is flexible to apply if necessary the formulation form the Eq :4.6 with mostly $k = 1$ and $p = 1$, also a set U is used to represent the non-interesting zone as it was presented in eq 4.5.

4.2/ CAMERAS COVERAGE

Once the area to cover is described by the grid, the next step is to verify for each point of the grid if one or more cameras cover it, based on Eq : 4.3 with $k = 1$.
To verify if each points of the grid is covered by a camera. It is primordial to talk about what is a camera, what kind of camera are appropriate and their projection model.

4.2.1/ CAMERAS DEFINITION

The closest projection model to the human view is the perspective projection. The perspective projection is also the more common and more especially in the field of area coverage as in example [72, 58, 91, 12, 87]. Other model of cameras or vision sensor can be used as for example omnidirectional with a $360°$ of field of view as [19, 11, 86]. Here we are only focussing on the camera perspective due to its wide use.

The pin hole or in Latin the "camera obscura" (see Figure4.6) is at the origin of the geometry model for the perspective projection.
The pin hole model is commonly composed by box (or chamber) hermetically closed to light, excepted by a small pin hole on the middle of the front side. All the ray of light reflected by the object of the world and passing by the small hole are projected onto the back side of the box. Each ray of light passing by the hole is projected on the plan (inside

FIGURE 4.7 – The rotation composed by 3 degrees of freedom on pan tilt roll(α, β, γ).

the box). This plan became the reversed image of the world and can be recorded by a film or a digital sensor. Due to the simplicity of the pin hole model, the calibration and camera projection estimation is simplified.

— Three degrees of freedom of the sensor's position : (x, y, z) ;
— Three degrees of freedom of the sensor's orientation : with the the rotation in pan, tilt, and swing angles : (α, β, γ)
— Optical parameters including : the focal length f of the lens, the sensor size $Sw \times Sh$, u_0 and v_0 which would be ideally in the centre of the image. σ_{uv} represents the skew coefficient between the x and the y axis.

Among the parameters of the camera, only some of them are useful to estimate the projection. They can be formalized as a vector :

$$v = (x, y, z, \alpha, \beta, \gamma, f, Sw \times Sh, u_0, v_0, \sigma_{uv}) \tag{4.8}$$

Each element of the vector v are used to compute the camera projection on the discretized floor.

4.2.1.1/ COVERAGE ESTIMATION IN THE LITERATURE

In order to compute the camera projection onto a grid the pin hole model is used with the parameters of the vector v.
The detail to estimate the camera projection on to the floor, based on the pin hole model and the parameters (v), has been detailed numerous times as in [24, 77, 35]. In [24, 77, 35] the camera projection is used to estimate if a point is visible by a camera, for each point of the grid. These articles handles the classic camera projection (with the 6DoF in [24] and 5DoF in [77]), both are used to estimate the 2D projection of the camera onto the floor.
In [24], the camera projection has been computed for several rotations for all the DoF. In this case, the projection can have numerous shapes (mostly parallelogram shape). In [77], the model of camera projection begins to be simplified by assuming some fixed parameters. The fix parameters allow more efficient by economizing part of projection computation of the camera at each time.
In [35], the model of camera projection is used to compute one time for a fix pan and roll in order to have a coverage estimation use in a 2D map. The camera projection is finally simplified by using a kind of triangle shape.
Some of them as [35, 77, 1] include the object occlusion. To detected the part of the

area occluded by an object, the solution commonly proposed (as is well explain in [77]) is to check the line made between a point covered ($g_i \in Pc$) and his camera. If this line is intersected by at least one object in the scene, the point g_i cannot be considered anymore as covered.

To go further other model and formulation inspired by the pin hole model has been proposed as [51, 1, 38, 23]. These models are inherited for the camera projection and adapted to fit their problems.
In [51, 23] the camera is considered to be placed on the floor with a fix pan (with the looking direction almost parallel to the floor). Therefore the camera projection is simplified by an isosceles triangle where the shape depends on the focal length.
In [1] the camera projection is also simplified in order to have a kind of isosceles triangle shape with considering the depth of view of the camera.
In [38] thanks to a fix pan and focal length, the camera projection is simplified in order to have a rectangle projection onto the ground. The sweep is designed consequently to the size of the camera projection, in order to minimize the overlap and have full coverage of the area.

One of the common point of the method present in [51, 1, 38, 23, 87, 58, 24, 77, 35] is the computation of a camera projection on to a grid. The computations necessary to estimate, if a point of the area is cover are not considered as really greedy (in time). But have to be done to each point g_j of the grid and for each camera v_i of the network.

$$\sum_{j=1}^{m} \sum_{i=1}^{n} f(v_i, g_j) \tag{4.9}$$

Where n the number of cameras in the network ; m the number of point in the grid ;
This equation (Equation 4.9) does not take in account the occlusion by few obstacle Obj.
If we add the potential occlusion by k object :

$$\sum_{k=1}^{k} \sum_{j=1}^{m} \sum_{i=1}^{n} f(v_i, g_j, Obj_k) \tag{4.10}$$

The function $f(...)$ in charge to compute a camera projection will be called in the worst case for each camera, in order to evaluate the complete coverage of the area. This numerous call will greatly increase the time computation of the coverage. It is even worst when numerous cameras projection (n) has to be computed at each turn of a long optimisation process.
In this condition the efficiency of the function $f()$ in charge then, computes the coverage estimation is primordial, due to the numerous call.

4.2.1.2/ Coverage estimation optimization

Design an efficient cost function is necessary to reduce the time computation and estimates the area covered. A good estimation of the area covered is primordial for the optimisation process. The computation of a camera projection have to be minimized with some basic assumption depending on the problems.
Considering our case, where a camera is fixed on a UAV with a looking direction orthogonal to the ground, without any rotations in α (pan) and β (tilt). It is possible to compute for

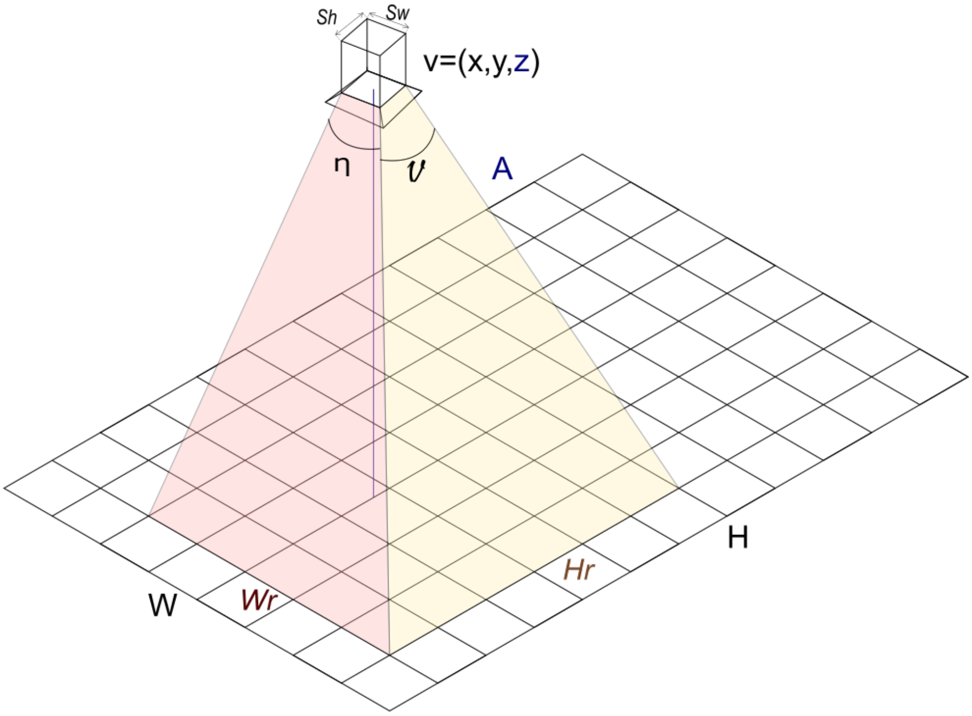


FIGURE 4.8 – Camera projection onto a grid. The grid is placed on the floor to discretize the area covered.

a given altitude the area covered by one camera, based on the given parameters. Especially the focal length f, the altitude of the cameras and the sensor size $Sw \times Sh$. In this model the camera projection is always a rectangle as described in Figure 4.8. To estimate the shape of the rectangle with is deduced from the ratio of s and the size of this rectangle with is deduced from (f, s, A). Where A is the altitude. The altitude is the distance between the grid and the cameras, in the simple case A is considered to be equal to z, when the grid is on the floor.
The basic computation to estimate the size of a camera projection :

$$\eta = 2 \times \tan^{-1}\left(\frac{Sw}{(2 \times f)}\right)$$
$$\upsilon = \left(\frac{Sw}{Sh}\right) \times \eta \tag{4.11}$$

Where η, υ are the horizontal and vertical camera fields of view.
Estimating the width and height of the rectangle projected on the ground depend on the altitude A :

$$Wr = 2 \times A \times \tan \eta$$
$$Hr = 2 \times A \times \tan \upsilon \tag{4.12}$$

Therefore, the size of the rectangle projected onto the floor is (Wr, Hr) for the width and height. The values of Wr and Hr are directly linked to the altitude of the camera. In the simple case and in the initialization, A is equal to z and z is selected in a range given by the UAV or the user.

Once the couple (Wr, Hr) as been computed for a $A = z$ it is easier to change the altitude of the camera. The rectangle projection will be affected in proportion.

$$\begin{aligned} f(A) &= (Wr, Hr) \text{ based on eq 4.12} \\ f(A.Coef) &= f(z) = (Wr.Coef, Hr.Coef) \end{aligned} \tag{4.13}$$

Therefore, to any altitude z is existing $A.Coef$ where the size of the camera projection onto the floor is $(Wr.coef, Hr.coef)$. Thanks to this the eq : 4.11 and eq : 4.12 have to be compute once for a given focal length f and sensor s with an simple $A = z$. Adding simple $Coef$ the size of camera projection can be easily simplified in order to limit the useless computation (only 2 multiplication instead then equation 4.8 and 4.11. The model of camera projection is greatly simplified by the UAV assumption (fix pan, tilt and focal length). That permits to consider the camera projection as a simple rectangle with a size directly related to the altitude (by using a simple coefficient $Coef$).
All this simplification helps the cost function to be fast and efficient.

4.2.2/ Parameter to optimize

By dint of the simplification presented previously the vector of parameters (4.8) can be simplified too.
To summarize, the computation of one camera projection onto the floor, where the floor is represented by a grid for the area to cover, and the camera is in altitude with a looking direction orthogonal to the floor. Just few parameter are necessary as showed in equation 4.11 to 4.13. Thanks to that the equation (4.8) can be reduced with keeping only the position of the camera and the roll as :

$$v = (x, y, z, \gamma) \tag{4.14}$$

or also

$$v = (x, y, A.coef, \gamma) \tag{4.15}$$

Reducing the number of parameters, passing to the equation 4.8 to 4.15 are really usefully to the optimisation of the problem. The reduction of the number of parameter will greatly affect the optimisation process.
Indeed, in addition to reduce the time computation, this simplification reduce the number of parameters to optimize.

Until now the camera projection estimation as was addressed with only one camera. But we want to compute the coverage for a set of cameras.
The solution is based on the previous estimation for the camera projection, but adapts to the location of each camera. By positioning the rectangle projection to have the center of it at the x and y position and compute the occupation grid.

In order to win a bit of time, each point of the grid already cover by a camera are note tested for the next cameras. This small modification will impact positively the computation time for the coverage estimation of a network of camera. That mean in the equation 4.9 the value of m decrease as i increase. More exactly the size of m decrease as the area coverage increase.

Representation of the parameters to optimize.
Until now, the camera formulation is adapted to one camera, to represent the problem we need a set of cameras. The precedent notation can be extended to have a set of V composed by n cameras defined by the parameters of v :

$$V = \{v_i\} \ , \ \forall i = [1; n] \ , \ n \in \mathbb{N}^* \\ \text{and } v_i = (x_i, y_i, z_i, \gamma_i) \tag{4.16}$$

Where n is the given number of cameras in the network. The coordinate of a camera v_i with are the ith camera of the network is defined with x_i, y_i, z_i, for a given room and γ_i the roll rotation (portrait or landscape). The parameters not contained in V and used to compute the cameras projection are identical for all the set V, and are fixed at the beginning of the optimization.
Therefore, V represents a solution. V contain all individual positions and orientations of the set of cameras for a predefined focal length, sensor size and map depending on the problem.Obviously all the solution V are not a "possible solution" for our problem. Some solution V does not respect the set of constraint noted E.

So that the V should respect the constraints of the set E (see Eq.4.17). Among the constraint few of them was already disused, as the occlusion, the map restriction, the k-coverage, or some constraint more specific to the problems (as saw in chapter 2).
The "possible solution" Vs must take in consideration with the set E as :

$$Vs = V \ , \text{ iff } E(V) = \begin{cases} 1, & \text{iff } E_i(V) = 1 \ , \text{ with } i = 1...Nc \\ 0, & \text{otherwise} \end{cases} \tag{4.17}$$

Where $E_i(V)$ is the function applied to verify the ith constrains of the set E on the solution V. Nc is the number of constraints needs to be satisfied to have an acceptable solution. That mean among all the possibles combination of parameters V only the one intersect the set of the constraint E are an possible solution. If we are considering all the V and all the E as two subset Vs is defined as $Vs = V \subset E$.

The problem of monitoring an area and more specifically the problem of area coverage may contain many constraints depending of the environment and the context. As example : the room shape, minimizing the altitude, have the best resolution, orientation of the camera, the possible occlusion,... All this constraints are included in the set E. The constraint have to be defined depending on the problems and the goal.

4.3/ Cost Function

The cost function has to evaluate the quality of an given answer. To estimate the quality of the answer one of the main criteria is the coverage rate. The precedent section has been focus on the computation of the coverage rate.
To create an efficient cost function, other criteria has to be taken in to account as the constraints. To establish the cost function a list constraint has to be done. Each constraint of this list does not have the same importance. In order to split the constraint depending then their impact on the problem two type will be explained before to dicusse about the implementation of the cost function.

4.3.1/ CONSTRAINT LIST

The constraint can be numerous and depend mainly of the problem formulation and the context. Like that few of them was briefly introduced in the previous section as in chapter 2, section 4.2.1.2,... This part is focus on the list of the constraint used in our case and detail their design.

Fixes number of the cameras One of the first constraint is the fix number of cameras. This constraint as some others (detailed latter) are useful to simplify and restrict the possibility of the problem. This constraint permits us to focus on the fine optimisation (as in [87] where both are tested). The number of cameras is fixed at the beginning of the optimisation and no more camera will be added during the optimization process.

Fixes parameters of camera and no rotations Fixes parameters of the cameras and no rotations (α and β) has been introduced previously (section 4.2.2). These constraints imposed by the use of an UAV are also an advantage for the optimization by simplifying the coverage estimation and limit the number of parameters to optimize. The parameters are fixed at each beginning of the optimization.

Fixes altitude The fix altitude is a constraint use in order to limit the number of parameters to optimize. The use of this constraint is used to reduce the complexity (see section 4.4). It is also useful for other assumptions, as a camera on the celling or for a submarine [25]. This constraint is an optional constraint and is not commonly used in the experiment presented in the following section.

The altitude boundary When the altitude is not fixed some limit must be chosen to avoid the extremely high and low altitude. The highest altitude will be fixed depending on the UAV ability and other restrictions as the laws. The lowest altitude has to be fixed for the safety of the user under the UAV. In practice the boundary of the z is defined with :

$$\inf z \leq A.Coef \leq \sup z \tag{4.18}$$

Where $\sup z$ is the maximal altitude of the camera and $\inf z$ the minimum altitude. A is the fix altitude of a camera where the camera projection has been computed with $A = z$ with only $coef$ vary as introduced in Equation 4.13.

The map boundary The map boundary is a constraint similar then the altitude boundary. Despite the shape of an area to cover some maximum boundary can be made. In fact for any shape as complex as it, it is possible to encapsulate in a rectangle. The rectangle map boundary is defined by a width W and height H. The boundary on x and y are :

$$\begin{aligned} 0 \leq x \leq W \\ 0 \leq y \leq H \end{aligned} \tag{4.19}$$

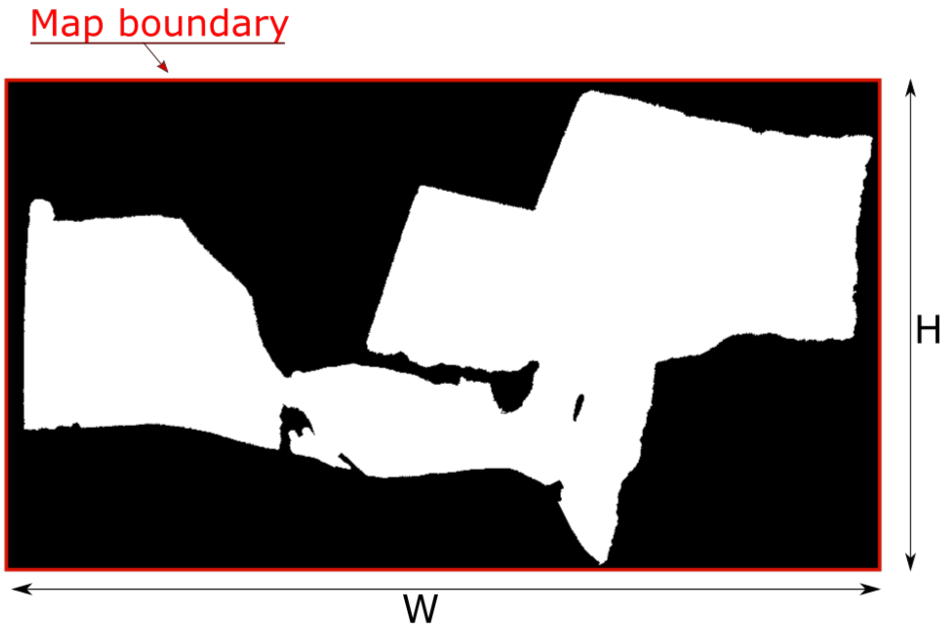


FIGURE 4.9 – Map to cover with the map boundary in red (W and H size) in black the sub-part have no interest to be covered.

By associating the altitude boundary (from Eq4.18), a cube boundary limits the position of the cameras in the 3 dimensional spaces.

$$\begin{aligned} &0 \leq x \leq W \\ &0 \leq y \leq H \\ &\inf z \leq A.Coef \leq \sup z \end{aligned} \tag{4.20}$$

Non rectangle map with possible hole. Despite the rectangle boundary of the area, the map to cover can be much more complex than a simple rectangle and can take any kind of shape. Also the shape of the area can be composed by holes. The figure 4.9 illustrate the map complexity. The black part of the map are the zone with are out. That mean these sub-parts have not interest to be covered. To take in account this constraint the gird has been designed with removing some of this points. The grid G is reduced in order to have only the points in the white sub-part. In this example each white pixel of the map is points of the grid. Concretely this implementation is easier for the complex map and has also some advantage.
Among the advantage, the flexibility of the grid customization. That allows the optimization to try some exotic solutions, as allowing the camera position on the black sub-part or in a border of it during the optimization. Obviously the exotic solution with a cameras position on the black side does not increase the coverage rate but if the optimisation converge correctly no camera will be on the the black sub-part of the map (or small part of it).

Some fix cameras in the set Having some fix cameras position in the set of cameras, is an optional constraint. This allows to have few manual cameras position from an user or other algorithms. One case can be to have some specific area as the entrance where have to be surveyed by a cameras dedicated to. To implement this constraint the solution applied during the experiments (presented later) is to adapt the map by removing the point of the grid G cover by the sub-set of fix cameras (as if these points of the grid was covered).

The resolution The resolution of the images is related then the sensor size (in px) and the distance between the camera and the object filmed. In our case, the sensor size is fixed by the properties of the camera mounted on the UAV and the object filmed is the floor of the area to cover. In this case, the distance between the camera and floor is the altitude.
The resolution constraint has to maximize the resolution. In order to maximize the resolution during the optimization the altitude criteria is modified in order to be the lower possible.
Considering only the resolution constraint as minimizing the average altitude of the cameras is harmful for the coverage optimization. Consequently, during the optimization a trade off between the altitude (and the related resolution) and the coverage rate have to be done $\min \frac{\sum_{i=1}^{n} A.coef_i}{n}$ and the coverage rate. In order to manage this trade off, the average altitude of the cameras is included in the cost function (see section 4.3.3).

4.3.2/ Constraint types

Among the constraints listed different priorities and restrictions exist. Indeed the constraint can be considered in 2 sub-class. The two sub-class are the hard constraint and the soft constrain presented in the following paragrapher.

Hard Constraint Some of the constraints presented are called "Hard constraint".
The hard constraints limit the possible solution by do not allowing the solution with does not respect it. This hard constraint is directly used during in the optimisation process to prohibit any solution to be out of this boundary. This hard constraint has to be integrate in the optimization process in order to cannot generate a solution with does not respect it. Consequently the hard constraint can some times slow down the generation of the individuals.
For example, the 3D boundary as defined in Equation 4.20 is a hard constraint. Each cameras position must be inside the 3D boundary.

Soft Constraint In the other case, some constraint can be considered as "Soft Constraint".
The soft constraints has to minimize the set of error. If a soft constraint is not fully respected the solution can be considered as acceptable and this small amount of error does not affect so much the final answer. In this case the soft constraint is assimilate to small acceptable error.
The soft constraint leaves the possibility during the optimization to do some mistake in order to learn about it. If the soft constraint is noted ϵ and the hard constraint are noted

ε' like that the constraint set is $E = \epsilon + \varepsilon'$.

$$\max f(Vs) - \min \epsilon \ \forall Vs \subset \varepsilon' \tag{4.21}$$

The objective is to maximize the coverage of a set of cameras ($f(Vs)$) with respect the hard constraints ε' and minimized the error form the soft constraints ϵ. Concretely the soft constraints are commonly integrated in the Cost Function as can be the resolution or the fix number of the cameras by the grid design.

4.3.3/ The cost function implementation

The cost function has the mission to estimate the quality of an answer. In our case, an answer is the position of a set of cameras. The cost function is essential in the process of optimization as that was introduced in the section 3.3.2.

The cost function has to estimate the area cover by a set of cameras in order to do that the area is discretized by a grid as in section 4.1.
The grid customization permit to introduce some of the soft constraint as the complex shape of the map by removing the point out of the area to cover. Also the fix cameras are added in the grid points as already cover area.
The grid modification will allow the cameras position to cover the area already covered and removed form the grid. The consequence of it will be to reduce the coverage rate possibility. The optimization will have to minimized this error.

To evaluate the coverage of a set of cameras is essential to can estimate the cameras projection of each, as detailed in the section 4.2.1. The area cover by the j-Th camera is noted as in equation 4.3 (where $Pc \in G$). By iteratively repeated this for each camera of the set the full area coverage is computed (as equation4.9).

Based on, the simplest cost function is the coverage estimation.

$$C(Vs) = \frac{\sum_{i=1}^{N} Pc_i}{m} \tag{4.22}$$

Where N is the number of cameras ; m represent the number of points needed to describe the grid G (as in Equation 4.1) ; Vs is the solution with respect the hard constraint. The cost function $C(Vs)$ give the quality of the solution Vs.

This version of the cost function $C(Vs)$ does not take in account the resolution constraint. The resolution is strongly linked with the camera altitude z (as show in 4.3.1). A criteria must be added in the cost function formula of the Equation 4.22. The average of the altitude z is used and have to be included in the cost function.

$$\overline{z} = \frac{\sum_{i=1}^{N} z_i}{N} \tag{4.23}$$

If the resolution is strongly related then the altitude the average of it ($\overline{z}$) can be considered as a part of the soft constraint (ϵ) in the equation 4.21 and the equation 4.22 may be updated as :

$$C = \frac{\sum_{i=1}^{N} Pc_i}{m} - \frac{\sum_{i=1}^{N} z}{N} \tag{4.24}$$

The equation 4.24 is used in the cost function to add the resolution constraint. The consequence of it, is the optimization will try to minimize the average altitude and maximize the coverage with no priority. Concretely by just applying this equation 4.24 the optimization will first minimize the average altitude by positioning all the cameras at the minimum altitude (with respect the hard constraint of altitude boundary) and in second time try to maximize the position (on x and y) of the cameras.

In order to have a priority between the coverage and the altitude a weigh has to be made on the equation 4.24. The weigh have to be chosen carefully. The weight has to be auto-adaptable depending on area covered. In order to give more priority to the coverage when the coverage rate is low and add importance to the resolution when the area is already well covered. The coverage has to stay the priority the resolution must be optimized in a second time. The best solution to do that, is to link the weight of the resolution criteria with the coverage rate.

$$\sigma \times \sum_{i=1}^{N} Pc_i \times \frac{\sum_{i=1}^{N} z}{N} \tag{4.25}$$

Where σ is a weighting coefficient at 0.06 to reduce the priority on the resolution criteria. Based on it the final cost function is :

$$C(Vs) = \frac{\sum_{i=1}^{N} Pc_i - \delta \times \sum_{i=1}^{N} Pc_i \times \frac{\sum_{i=1}^{N} z}{N}}{m} \tag{4.26}$$

Thanks to this formula, a proposed answer Vs can be evaluated and returns the quality of the solution for the problem of the coverage maximisation and the minimization of the altitude in the second time. The cost function integrate all the soft constraint either by the design of the grid or by the formula of the cost function $C()$.

The cost function presented is the final one, but the building of it was an incremental work and numerous version was test in term of weight, priority, and constraint. The one presented here is the more equilibrated.
Despite that some of the work presented in the following section are made with the basic cost function from the equation 4.22. In this case, the element composing v can be reduced as only $v = (x, y, \gamma)$.

The final and complete cost function $C(Vs)$ have as input a vector Vs with are composed by all the cameras position and orientation of the network. It is also composed by the map of the area to cover G where G include the soft constraints as the room shape, the fix camera. The constraint of resolution is added by using the average altitude in the equation 4.26. The value is returned by the cost function $C(Vs)$ is the quality of a solution to our problems of coverage using an UAV.

4.4/ Optimization complexity

In spite of the simplification presented before, the problem stays complex. There exist many positions for each camera to cover an area with a certain amount of sensors. This number of position can be estimated as follows.

Each camera defined by the position on x, y, z and γ can be set anywhere in the search space named Sp :

$$Sp = (W \times H \times (\max(A) - \min(A)) \times 2) Sp \in E \tag{4.27}$$

Where W and H are the size as width and height of the area to cover, $\max(A) - \min(A)$ is the range of possible altitude. 2 is to define the roll γ, as the rectangle projection is horizontal or vertical (landscape or portrait). The search space Sp allows to take in consideration some strong constraint E like the boundary of the area or the restriction in the degree of freedoms.

The problem of the search space is the propensity to increase rapidly as the area grows. This phenomena is accentuated by the size of the set of cameras N.

$$\binom{N}{Sp} = \frac{Sp!}{N!(Sp - N)!} = |Vs| \tag{4.28}$$

Where $|Vs|$ is the number of possible solution for a set of N cameras in the worst case. In fact the size of the search space (as Eq. 4.27) associate to the set of N cameras (as Eq.4.28) make an exponential number of possible solution depending mainly then the size of the area and the number of the cameras in the network.
Obviously in view of this behaviour the use of a deterministic solution based on a heuristic does not seem to be a good answer to have an efficient solution. In addition the number of possible solutions $|Vs|$ makes the computation of an optimal almost impossible due to the numerous local minima. This kind of problem can not be used with an algorithm dedicate to the research of global optimal solution but need to be used with an algorithm focusing of optimizing the solution and returning an acceptable solution.
The formulation of the problem is primordial in order to reduce the size of the search space and give a chance to the optimization to converge.

5

Comparative result and experimentation on Waypoints positioning

Sommaire

To remind, the main objective is to propose an efficient path to can cover an area using a camera mounted on an UAV. The solution proposed here, is to focus on optimizing the position of a cameras set to fully cover an area in a first time. When set of optimized cameras pose is found the position can be used as waypoints for an UAVs path. Indeed find an optimized position for each camera of a given set is primordial. The following sections are dedicated to the optimization of it.

During the previous sections, the problem was discussed as an optimization problem. The formulation of the problem was presented and the complexity of the problem was disused in the section 4.4. Thanks to this preliminary result the following section is only focused on the optimization process.

5.1/ Other algorithm used

5.1.1/ PSO

The PSO (Particle Swarm Optimization) is an algorithm dedicated to the optimization problems. It is a stochastic algorithms from the family of evolutionary algorithms (see chapter 3). The PSO is a relatively young compared then the other EA. It was developed by Russel Eberhart and James Kennedy in 1995 [148* bis][**?**]. The concept of PSO is to optimize iteratively a continuous non linear function. To do that the PSO is inspired by the behaviour of animals. As it appends here from the bird flocking, fish schooling and swarming theory. These are animals working in a group to seek food. The direction to take is not decided by one leader, but by all individuals in the swarm by relaying just few informations as what quantities of food they found. The swarm composed by numerous individuals became smarter and more efficient to reach their objective. The algorithm proposed by Russel Eberhart and James Kennedy in [148* bis] [**?**] are directly inspired by these behaviours.

The methodologies used, is to examine each individual or also called particles as a solution of the problems. The problem is optimized at each iteration. To do that each solution must be comparable and quantifiable. At each iteration, each particle has to be tested by a cost function in order to discriminate the best particles of the swarm. The cost function and the design of it has been detailed in the chapter 4. When the best particle is found at the end of an iteration, the other particles of set, try to change their initial direction to converge more or less quickly to the actual best. Indeed the power of this algorithm is to obtain a very basic individuals behaviour to guide the particles. Each particle is guided by 3 behaviours.

- This own velocity V_k.
- This own best solution P_i.
- The best solution P_g.

Here the velocity represents the useful speed of the particle to converge to the best solution. More the velocity is high more the step at each iteration will be long. The behaviour of the particles X_k are modelled by the following equation to obtain the new position X_{k+1} :

$$\begin{gathered} V_{k+1} = \omega V_k + b1(P_i - X_k) + b2(P_g - X_k) \\ \text{and} \\ X_{k+1} = X_k + V_{k+1} \end{gathered} \tag{5.1}$$

Where ω is the inertia. $b1$ is random value between 0 and ϕ_p and $b2$ is random value between 0 and ϕ_g. ϕ_g and ϕ_p are the scaling factor to search away from the particles is best known position (Default : 0.5).

Thanks to this basic behaviour of the particles the swarm can coverage to a global solution. To have an efficient optimization just few parameters must be set-up for the PSO. The more important are the inertia of the particles, the size of the swarm and the initial dispersion.

- The inertia will globally help the particles to keep their initial velocity. The consequences of the high inertia, is to explore more the search space and therefore the convergence will be longer.
- The size of the swarm have an impact on the convergence time (in number of iteration) and also the time computation. Indeed a big amount of particles in the swarm

means more exploration of the search space at each iteration, but also more comparison to find the best particles (the comparison may have a non negligible computation time). The swarm size is commonly fixed but can be as the population in the GA (see section 3.3.3) dynamically adjusted during the optimization process.

— The initial dispersion of the swarm can be a decisive element as the population for the GA (see in 3.3.3). For the PSO the use of an heuristic to initialize all the particles of the swarm is not recommended due to this important risk to converge prematurely in a local minimum. The random dispersion appear as the more appropriate for a global optimization. On the other hand the fast convergence and the PSO ability to climb the small hill to go out of the local minima can be used in order to refine an other optimized solution. The principal risk is to optimize around the initial dispersion and do not explore correctly the search space.

— Other criteria as ϕ_g and ϕ_i are minor but can be useful to have a realy fine adjusted PSO.

Finally to summarize the PSO is efficient in term of optimization despite a very basic behaviour of each particles. Each particle has this own velocity defined part way by the random and controlled by a global parameter; the inertia. The power of PSO is at the same time this efficiency to solve the optimization problem and this simplicity of use. In fact, the PSO needs at minima few element to work properly : A cost function, an inertia parameter and the size of the swarm. These efficiency and simplicity of use explain this popularity during the last decade.

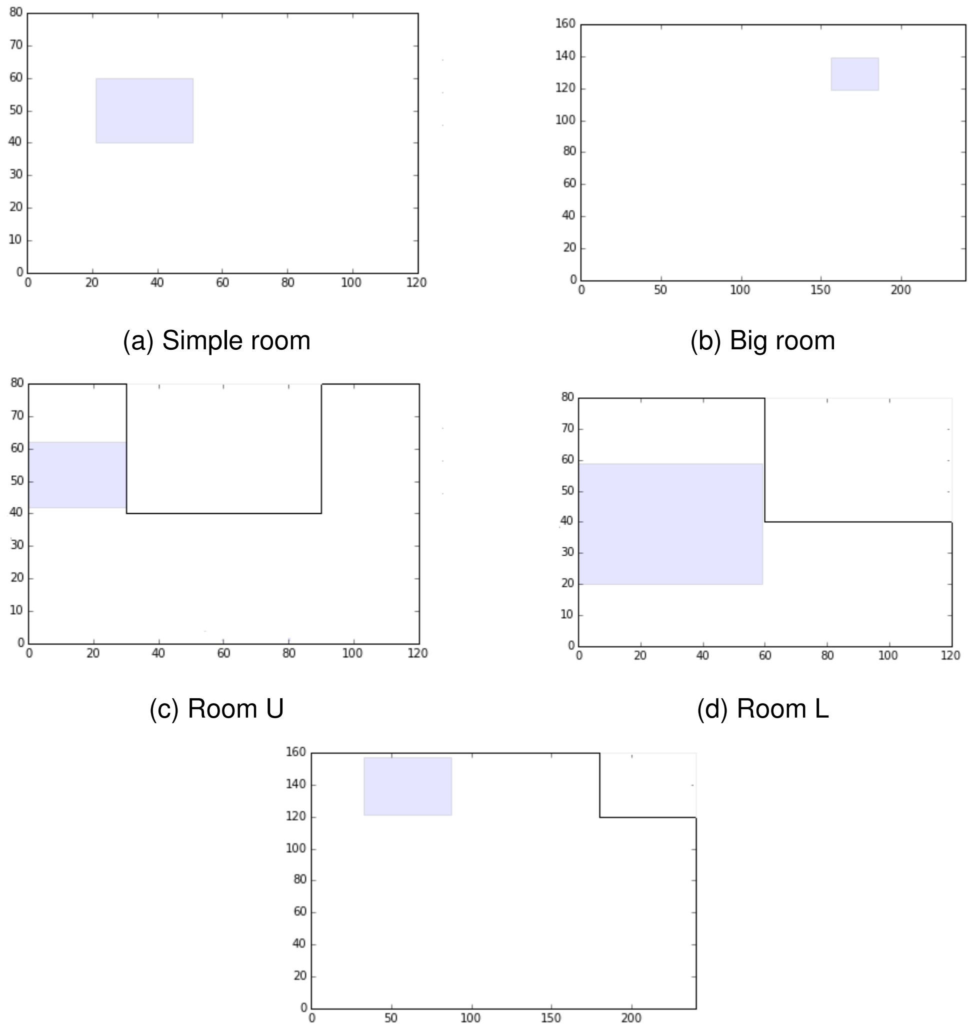
5.1.2/ Random selection

The random selection (named RS) is a very basic algorithm. It serves as a reference points for the comparison of different algorithms. The RS does not take a complex metaheuristic and is perfect to compare the efficiency of the other algorithm.

The random selection works by randomly generate numerous solutions. Among the solutions randomly generated the best solution is kept as the optimized global solution. The RS allows to look through the search space by randomly try different possible solution. The search space exploration by the RS is only made by random sampling without other optimization method. Indeed the RS is invoked as a reference points for the other algorithms. If the RS get a similar result with the same numbers of the cost function call, the algorithm compared can be considered as not more efficient then a simple random solution. The RS objective is to serve as reference point for the other optimization algorithms. It will be a perfect reference to judge the efficiency of the optimization from other algorithms.

5.2/ Algorithm comparison

To solve the problem of cameras positioning (or waypoints positioning) the usable algorithms are varied as that was discussed in the chapter 2 (see the sum-up tables 2.1 and 2.5.2). Among the algorithms studied in the literature the EA family appear as the more suitable to have an appropriate answer despite the numerous constraints of our problem. The EA is a vast family of algorithms. Among this family the more used for our problem is the PSO (see 2.5.2). The PSO gives a good and fast result in many case. In the EA



(a) Simple room

(b) Big room

(c) Room U

(d) Room L

(e) Big room L

FIGURE 5.1 – For the experiments : (a), (b), (c) the blue rectangle represents the field of view of one camera projected onto the ground with z=1 (30×20px) and (d), (e) with z=2 (60×40px)

family, the GA is one of the founders and was one of the more popular due to this great flexibility and efficiency. After more investigation the GA is under estimate for the problem of camera positioning unlike the PSO (see [58, 91, 83, 44, 24, 23]). Base on the work of Boeringer et al in [6], where the PSO and the GA have been conscientiously compared. The conclusions of the comparison in [6] is relatively open and highlights the similarity of result between the two algorithms. An experimentation has to be done to find the best algorithms for the problem of cameras position in a complex environments.

5.2.1/ DESIGN OF EXPERIMENT

To find the best coverage, many experiments have been used to compare PSO and GA. PSO is easier to implement and runs faster, but GA is more flexible and generic thanks to the many tunable parameters. The following subsections will provide a comparison between PSO and GA with using RS as a reference.To compare and evaluate their performance, we tested them in different scenarios. The scenarios have been designed to have a different shape and size. The shapes of the room have been designed to estimate the exact number of cameras in order to have a ground trough. Due to the use of a non heuristic algorithms the shape can be considered as complex. The rooms are depicted in Figure 5.8, with areas of different size and shapes, where :

- z is the height of the camera between (within the range $[1/z; z]$).
- Figure 5.1a is an area of size 120×80 (named Room).
- Figure 5.1b is an area of size 240×160 (named Big Room).
- Figure 5.1c is an area of size 120×80 (named Room U).
- Figure 5.1d is an area of size 120×80 (named Room L).
- Figure 5.1e is an area of size 240×80 (named Big Room L).

The design of the experiments in Table 5.1 has been set up to identify the most efficient algorithm for the positioning of a set of cameras with maximum coverage depending on the numerous cases. The Design of Experiments (DOE) has been made to take in account ; the shapes, sizes, some constraint as the fix altitude and many size for the set of waypoints. The DOE has been established to highlight the impact of the constraints on the optimization process with the GA and PSO.

<table>
<tr><td colspan="2">z=1</td><td colspan="2">GA</td><td colspan="2">PSO</td><td colspan="2">RS</td></tr>
<tr><td colspan="2"></td><td>GT</td><td>NC</td><td>GT</td><td>NC</td><td>GT</td><td>NC</td></tr>
<tr><td rowspan="2">Room</td><td>120x80</td><td>16</td><td>20</td><td>16</td><td>20</td><td>16</td><td>20</td></tr>
<tr><td>240x160</td><td>64</td><td>70</td><td>64</td><td>70</td><td>64</td><td>70</td></tr>
<tr><td>Room U</td><td>120x80</td><td>12</td><td>20</td><td>12</td><td>20</td><td>12</td><td>20</td></tr>
<tr><td colspan="2">z=2</td><td colspan="2">GA</td><td colspan="2">PSO</td><td colspan="2">RS</td></tr>
<tr><td colspan="2"></td><td>GT</td><td>NC</td><td>GT</td><td>NC</td><td>GT</td><td>NC</td></tr>
<tr><td rowspan="2">Room</td><td>120x80</td><td>4</td><td>10</td><td>4</td><td>10</td><td>4</td><td>10</td></tr>
<tr><td>240x160</td><td>16</td><td>20</td><td>16</td><td>20</td><td>16</td><td>20</td></tr>
<tr><td rowspan="2">Room L</td><td>120x80</td><td>3</td><td>10</td><td>3</td><td>10</td><td>3</td><td>10</td></tr>
<tr><td>240x160</td><td>15</td><td>20</td><td>15</td><td>20</td><td>15</td><td>20</td></tr>
</table>

TABLE 5.1 – Design of the experiment for comparing the efficiency of PSO and GA in different conditions. (GT is Ground Truth and NW is Number of Waypoints).

The Ground Truth (GT) is the minimum number of cameras required to fully cover a given area. The size of the area has been selected so that the GT can be easily estimated. NW is the maximum Number of Waypoints (or cameras) used for the experiments. At each experiment a solution is computed for a number of cameras from 1 to NW. To compare the different algorithms fairly, only 10 000 calls of the cost function are allowed for each optimization. The optimization has been executed 8 times for each optimization process. 8 times is the minimum number of test has to be done to can have a usable average despite the hight volatility due to the randomness of the algorithms.

5.2.2/ ANALYSIS OF THE RESULT

After performing several experiments (see Table 5.1), it appears that the GA and PSO algorithms are close in performance in numerous case. Among several experiment of the

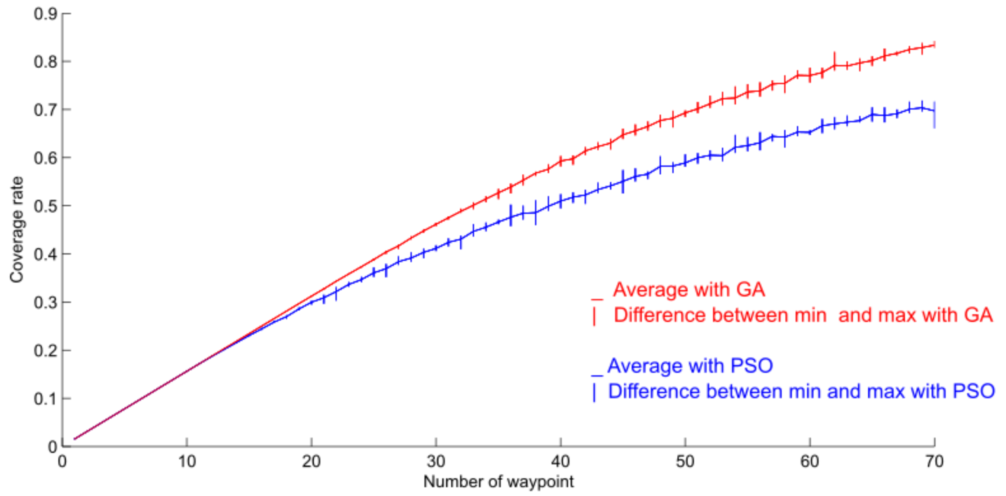


FIGURE 5.2 – Comparison of eight solutions given by the GA, with eight solutions given by PSO algorithms with a fixed altitude (z equal to 1) in the big room 240×160. The ground truth for this room equals to 64.

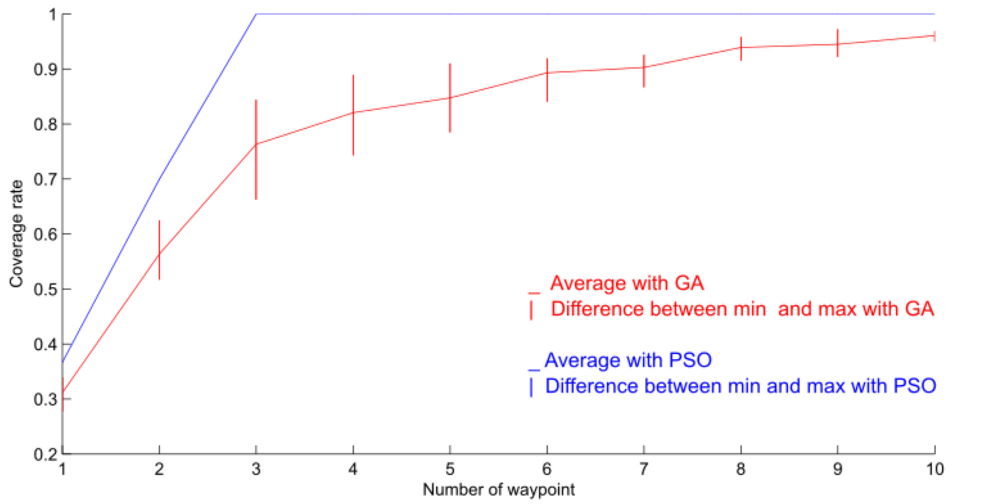


FIGURE 5.3 – Comparison of eight solutions given by the GA, with eight solutions given by PSO algorithms with a Z between $[1/2; 2]$ in the room with L shape 120×80 and ground truth equal to 15.

DoE some particularity appear despite the globally close result of GA and PSO. Also as expected the RS is always the worst solution. In the following subsection just few experiments are taken to illustrate some interesting phenomena specific to the GA and PSO for our problems.

In the case where the search space is large and numerous dimension have to be optimized by the GA appears globally more efficient as in Figure 5.2. In contrary in this case (big room with z=1) the PSO gives a very bad answer close than simple RS. Instead, PSO is more effective for optimizing small areas as in Figure 5.3. In the small room in L shape with a z between 1/2 and 2, the PSO reach quickly (quicker than the 10 000 calls) to the optimal solution. Where, here the optimal solution is known and equal to GT. In the same case the GA propose an optimized solution (compared then the RS) but far from the PSO.

This efficiency can be explained by the slight variation of the solution introduced by the PSO. However, this slight variation is not enough to find an optimized solution in a big search space that occurs when many cameras are required or when the local minimum is deeper. The PSO appears really efficient in a relatively small search space where the number of dimensions to optimize is not too high. On the other hand, the variety introduced by the GA allows to escape from the local minima. This variety is helpfully in the big search space in order to explore quickly a wide part of it. The variety introduced by the GA became a handicap for a more fine optimization. That explains the bad result obtained during the experiment in the small room. The variety of the GA negatively affects the accuracy of the solution and may require a further optimization step to refine.

5.3/ Hybrid GA PSO

Thanks to the experiment done and presented in the previous section (see 5.2) the GA and PSO are two algorithms efficient and complementary to solve the problem of camera positioning in the complex and potentially vast area. To summarize the preceding comparison, it is difficult to rank the two algorithms in all the environments. GA and PSO have both advantages depending on the area and the number of waypoints to pose estimate. GA is better in the vast search space and for several dimension to optimize. When PSO is efficient to refined faster the solution. The hybridization can be the key to optimize the camera positions in all the condition. The aim of the hybridization is to exploit the better of both algorithms, trying to further refine the solution.

5.3.1/ The different hybridization

Different hybridization of the GA and PSO can be made. Each hybridization of GA and PSO has the advantage and disadvantage. This following section is focused on the main hybridisation of GA and PSO.
In Premalatha al et 76* [**?**] propose three different solution to hybrid the GA and PSO :

— GA and the PSO are employed in parallel. The best solution between both algorithms is used into the other algorithm. For example : If the best solution at the end of the first generation is from PSO, this solution is used as a new individual for the crossover on the GA. Or if the best solution is from the GA, this good individual is employed in PSO as best particle for the next draw. This operation continues until such time as the convergence of both algorithms.

— The GA is used to introduce variety on the PSO, when the PSO is stagnating. Stagnated states are reach when no solution upgrades after a predefined number of iteration. In this case, the GA introduces variety by proposing other solutions for PSO. This hybridization has to be managed carefully due to this high risk of non convergence.

— The GA is used until the convergence point. When the GA converge to a solution, the PSO is used to refine with one more optimization. This solution is costly in time due to this double optimization and this double convergence. Finally this hybridization uses the GA optimization as an initialization for the PSO.

The last hybridization, using GA as initialization for PSO is probably one of the most suitable for the problems of camera positioning in a vast and complex area. The experiment made until now (see 5.2.1) confirm the mechanism described by the last hybridization of GAPSO. In fact for our problems GA is efficient to run through all the search space and in the other hand the ability PSO to refine the solution is also confirmed. In this case, the GA can be a very good initial guess for the PSO.

In Shi et al [64] the hybrid PSO GA (same as GA and PSO employed in parallel) was studied for 6 problems listed F1 to F6. The 6 problems have a global optimal knew. In this article [64] the different problems are used to demonstrate the efficiency of hybrid PSO GA and search the appropriate set up for their parameters. One of the interesting aspects presented in [64] is the importance given to find the best set-up for each algorithms. The set-up of the algorithms has to be adapted to the hybridization and the problems. As that was disused in shi et al [64] numerous tests have to be done to find it.

In our case, the GAPSO is used within a first time a GA and a PSO next to refine the solution. In this case, the GA has to introduce even more variety in order to be more efficient. Consequently the GA has to be modified to have a mutation ratio higher.

5.3.2/ Experimentation

To compare the efficiency of the hybridization GAPSO to the GA, one experimentation is proposed. The experimentation followed the rule fixed during the comparison as in Table 5.1. It appears the big room in L shape as the Figure 5.1e is the most suitable to test the hybridization. The L shape room proposes a big search space which can require a big amount of cameras to cover it. This configuration is the most likely to be improved on a different situation, also closer to a realistic configuration.
The proposed experiment uses the GA for a maximum of 100 generations and the GA solution as an initialization for the PSO. Also the PSO is locked at 100 iterations.

The set-up of the GA has been slightly modified by increasing the mutation ratio and the PSO is also adapted by reduce the inertia. Before to reduce the inertia of the PSO few test was quickly made, especially with a dynamic inertia. A dynamic inertia can be efficient and allow the PSO to start with a bigger inertia to visit more the search space at the beginning and time after time the reduction of it help the PSO to converge faster. In this case, a small decrement of the inertia is applied at each iteration. Finally this method does not provide a significant gain. The dynamic inertia is finally not really useful in the context of hybrid GAPSO. The solution preferred for the PSO set-up has a slightly lower inertia parameter (around 0.4).

5.3.3/ Result and comment

The big room in L shape where the comparison between simple GA and PSO was performed, is also used to compare the GAPSO efficiency. The GAPSO is compared with the single GA in one side and the single PSO on the other side.

In the Figure 5.4 it is appearing the hybridization of GA, PSO increases slightly the per-

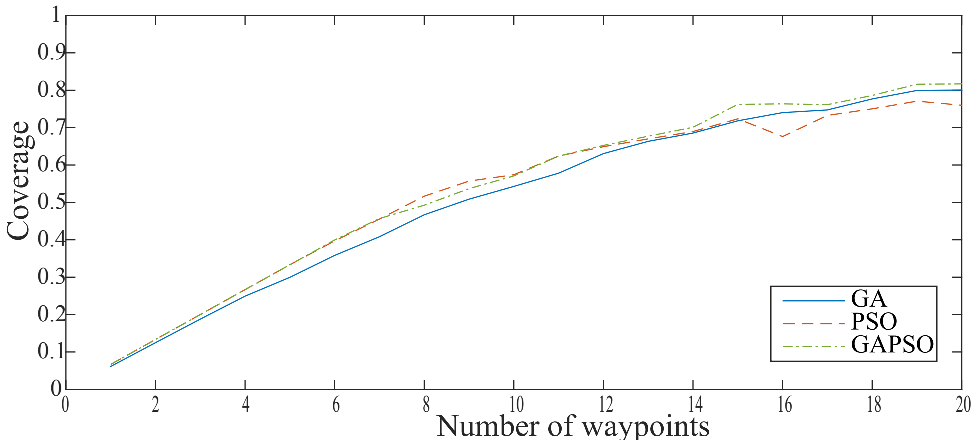


FIGURE 5.4 – comparison between GA PSO and the hybridization of GAPSO.

centage of coverage.This graphic can be split in 2 parts, the left side with a relatively low number of waypoints (or cameras) to pose estimate (until 15) and the right part with more waypoints. To remember in this experiment for each camera (or waypoint) is defined in x, y and z. That means for 15 waypoints, 45 dimensions have to be optimized. The two sides of the graphic show efficiency of the different algorithms and confirmed the mechanism of GA and PSO. The PSO is more efficient in the beginning when the numbers of dimensions to optimize are reduced. Otherwise as we saw previously the GA is efficient in the big search space with an important amount of dimension to optimize. The GA became better than the PSO in the right part of the graphic 5.4. The solution proposed by the GAPSO on the left side of the graphic 5.4 is equal or a bite better then the PSO. On the other side, the GAPSO propose a solution more refine than the simple GA. This refinement is due to the PSO ability to optimize the solution from the first optimization (GA).

Finally the biggest advantage of the GAPSO is to propose most of the time the best solution and some time slightly better by combining the advantage of both algorithms. The GAPSO can reduce the limitation of the GA and help to go deeper in the optimization process. The GAPSO beside to upgrade the solution initially proposed by a simple GA or PSO offer more flexibility and allow only one solution to be efficient. The GAPSO is efficient despite the number of dimensions to optimize and potentially more robust depending on the size of the search space. Despite these great advantage (better solution and more flexibility). The principal inconvenient of the GAPSO is caused by this double convergence. In fact with a hybrid GAPSO, as we decide to use a GA has to be executed until a convergence and in the second time optimized with a PSO. Obviously this implementation increases the time of computation.

5.4/ GOING FURTHER, MORE EXPERIMENT

The previous section with the different experiments shown the efficiency of the GA for the vast areas and the flexibility offer by the hybridised GAPSO (with a small refinement of the solution). The experiments made until know was focused on different area relatively simple the next step is to increase the difficulty of the scene. The increased complexity is made by adding :

- More obstacle.
- Hole in the area.
- Increase size of the area.
- Increase the search space by adding more parameters(as the roll).

The following section present the results obtained by increase step by step the difficulty. Each step presented chronologically in the following section has been made to test and some time to refine the parameters of the GA and GAPSO in different contexts. In the following section only the more significant step has been presented.

5.4.1/ Rectangle obstacle

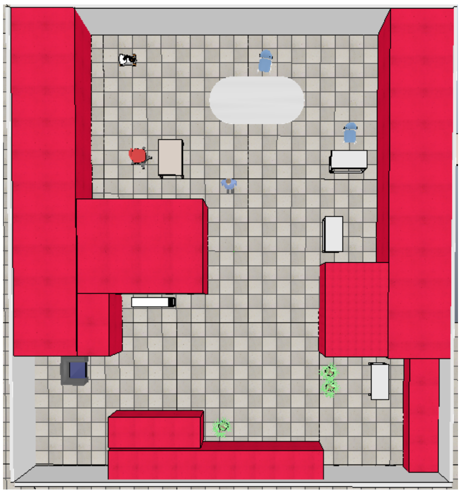
Thanks to the result obtained (see 5.3) an experiment is made using a slightly bigger size than the big room (in Figure 5.8). The environment of this experiment to try to be more realistic, consequently the room is designed with more obstacle. The obstacles are added in the map as non-interesting area to cover as explained in section 4.3.1. Also for the first time the design of this room is not made in order to have perfect ground trough unlike the previous room design (Figure 5.8). The consequence of it is the number of camera use to cover all the area is not an integer and some overlap must append. For this first realistic experiment a simulation tool dedicate to robotics is used to illustrate the result.

The simulated room is 15×14 m^2 which corresponds more or less to a large lecture hall. The areas in red (see Figure 5.5a)represent the zones which do not require coverage. Every camera can cover a 4×3 m^2, when z is equal to one. The z factor can be equal at $[0.5, 1, 1.5]$, and the cameras can turn at 90° to have the image in portrait or landscape. All of these parameters are taken into account in order to compute the waypoints position.The optimization of the waypoints position is made using only the GA, but this time it is applied until the convergence.

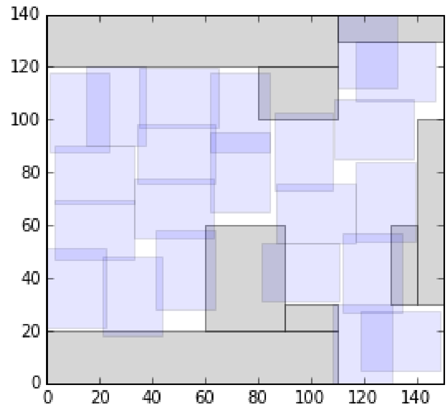
After running the single GA a well optimized waypoints positioning is given (see Figure 5.5b). Not perfect but good enough with a limited number of cameras. At Each waypoints an image is captured in order to offer a mosaic image of the scene with a restricted number of small black holes (see Figure 5.5c). The solution obtained is comparable to the experiments made previously (see 5.2.1). As aspect, despite the increasing number of obstacle and increased size of the search space. This confirms again the ability of adaptation of the EA optimization and more exactly the single GA. These results encourage us to go further.

5.4.2/ Rectangle obstacles with hole

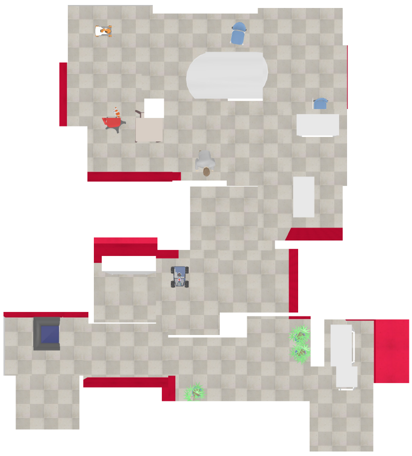
The previous experiment shown the efficiency of the single GA despite numerous obstacle and slightly bigger room. The following experiment try to push a bit more the optimization process using a single GA. The increased complexity of map was made by adding much more obstacle with some holes in the middle of the map. In fact until know all the obstacles was added around the bounding of the area. Add Obstacle in the middle to create a hole in the area, that increase significantly the complexity of the coverage estimation. To simulate a realistic environment the map is designed manually based on a satellite images (see Figure 5.6a). Each rectangle obstacle has been placed to reproduce the buildings as in the satellite images.



(a) Room in V-rep.



(b) Poses of every image captured in the room.



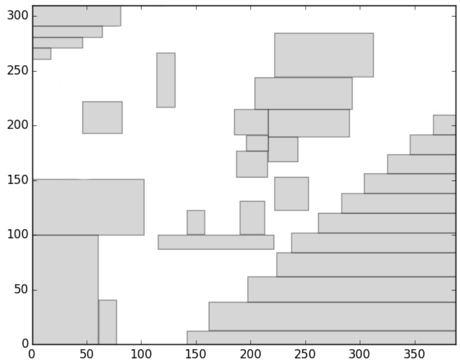
(c) Reconstructed room based on the waypoints positions.

Parameters	**Value**
z	[0.5;1;1.5]
γ	portrait or landscape
Size of the area	150x140 px
Maximum size of camera projection	30x22 px
Number of waypoints	22

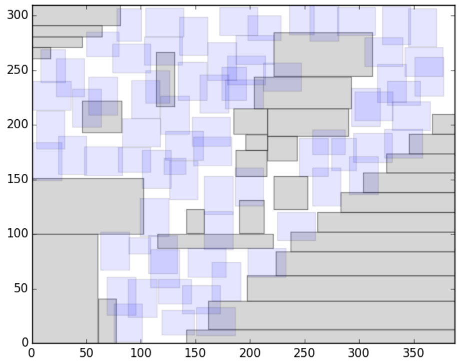
FIGURE 5.5 – Indoor area coverage using V-rep to simulate a realistic environment.

The result obtained by the GA optimization (see Figure 5.6c) show one more time the adaptation power of the single GA to the complex scene. The total coverage of the area is around 76.5% for 75 waypoints. The answer proposed is not perfect and can be improved in order to reduce some overlap and black hole. The important number of dimension to optimize (75) and the increased size of the area (twice bigger then previously) mark the limit of the simple GA optimization.
The more interesting aspect of this experiment is to show the efficiency of the single GA in a real complex environment. The solution proposed can be considered as good for the purpose of the challenge (obstacle, hole, numerous waypoints to pose estimate, vast

(a) The area to cover taken form satellite images.



(b) The mask of the area to cover.



(c) Area cover by the network of cameras using the GA.

Parameters	**Value**
z	[0.5;1;1.5;1.75]
γ	portrait or landscape
Size of the area	380x310 px
Maximum size of camera projection	42x54 px
Number of waypoints	75

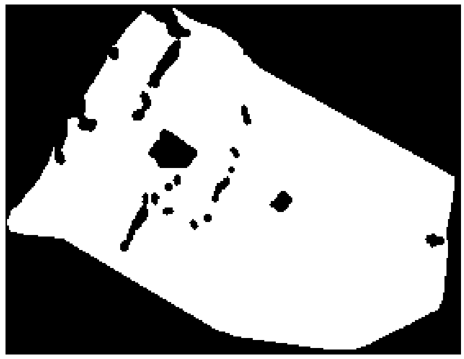
FIGURE 5.6 – Coverage area from satellite images with 75 cameras for a coverage of 76.39% using the the GA.

area,...).

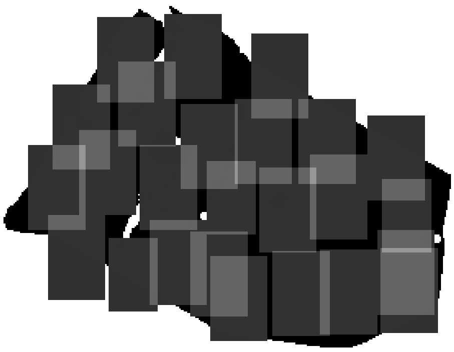
5.4.3/ USING MASK TO DESCRIBE THE AREA

Based on the limitation of the map design and the necessity to go one step further the paradigm has to evolve. The solution proposed representing the area until now, was to add rectangles obstacles by removing the corresponding points of the grid (as explained in 4.1.1.4). The primary advantage to use rectangle obstacles was in the coding implementation. This facility becomes a lock for the more complex area. In addition, it is revealed not user friendly.
The solution chosen for the flowing experimentation is to use a binary mask of the area to cover. The mask represents in the white side the area to cover and in the black side of the non interesting zone (also called obstacles, see 5.7b). This solution is finally more "user friendly" and do not change the fundamental of the grid map used until here. Each

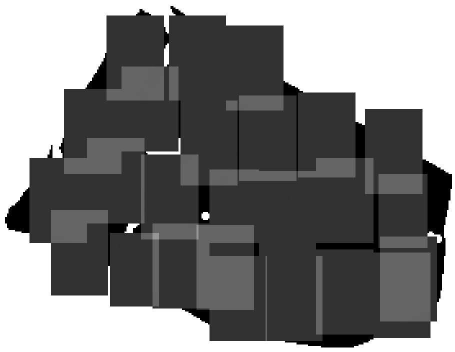
(a) The area to cover taken form satellite images.



(b) The mask of the area to cover.



(c) Area cover by the network of cameras using the GA 90.76%.



(d) Area cover by the network of cameras using the GAPSO 92.81%.

Parameters	**Value**
z	[1;1.5;1.75]
γ	only portrait
Size of the area	278x214 px
Maximum size of camera projection	35x52 px
Number of waypoints	25

FIGURE 5.7 – Coverage area from satellite images with 25 cameras for 90.76% of coverage using the GA and 92.81% of coverage using the GAPSO.

white pixel of the mask is a point of the grid to cover.

For this first experiment with the binary mask to describe the area to cover, a smaller area is selected. A smaller area involved a smaller amount of similar waypoints necessary to fully cover it. In the Figure 5.7a the area to control is extracted from a satellite images to have a mask (see5.7b). The single GA is performed with 25 waypoints to pose estimate. The solution obtained by the single GA is re-injected in the PSO. Each waypoint has to be placed on $x;y;z$. In order to test this new paradigm the rotation γ is removed from the parameters to optimize.

To begin the single GA was performed. The result are visible in Figure 5.7c. The optimized waypoints positioning, cover 90.76% of the area with 25 cameras and the single GA converge after just 67 generations. The solution given by the single GA is already good enough despite the complexity of the map. The solution obtained is conformed to the expected result. The solution of the single GA gives a well optimized waypoints poses, despite the new paradigm with the complex map.

Among the experiment presented until now (5.4.2 and 5.4.1) only the single GA was employed for the optimization. On the last experiment (5.4.2) with a bigger and more complex area composed by hole, the limiting of a single GA appears slightly. This observation is confirmed and is getting bigger for the area more complex. More complex as the one outcome the map designed according to the satellite images with mask (as Figure 5.7a and 5.7b). The solution proposed being to apply a GAPSO. The PSO will allow the refinement of the GA solution. The PSO is used with an initialisation from the first optimisation (using the GA solution). Finally the result presented in the Figure 5.7d shown a much more refined coverage with significant reduction in the amount of black hole and overlap (coverage is over 92.8%).

The main result of this experiment was to evaluate if the paradigm modification may have a significant impact on the waypoints positioning. The conclusion of this experiment is, in fact not so much when the GAPSO is applied. Despite the increased complexity due to the area shape (possible by the mask) the use of the hybrid GAPSO permit to composentes it. The experiment allows also to evaluate the improvement made by the GAPSO. The GAPSO hybridization is robust and flexible despite the strong constraint due to the non geometric area.

Thanks to this tries the size of the area to cover and the number of waypoints can be increased. The next experiment has to test the boundary of the GAPSO optimization in term of size and number of cameras using a mask for describing the area.

5.4.4/ Using mask for bigger area

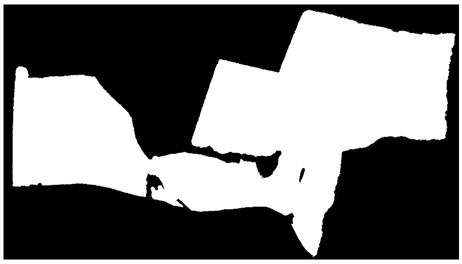
Based on the last experiment (5.4.3) a much bigger area with much more waypoints to pose estimates are presented here. The goal of the following sections is to see the boundary of the GAPSO optimization when the sliders are pushed to the maximum. The maximum in term of area size, number of waypoint and shape complexity.

5.4.4.1/ Vast and complex outdoor

In this experiment, a much bigger and complex area is presented. A much bigger area involves the increasing of the search space and consequently the increasing of the complexity. In the following example Figure 5.8b the satellite images are used to define the area to control (in white). The size of the area has been increased to have a grid composed by almost half million of points with nearly to 200 thousands points to cover. The objective is to increase the difficulty in this experiment. In addition to the increased size of the area to cover, the shape of it has been complicated. To do that an area with several sub-parts composed by small space and hole has been selected.
During this experimentation, a high coverage rate is required. We aspect more than 95% of coverage rate. More than 95% of coverage rate is a hight requirement and push the optimization in term of precise positioning to the limit of the GAPSO. The risk to ask a very hight coverage rate is to need a lot of waypoints with several overlap and consequently a long time before to converge.

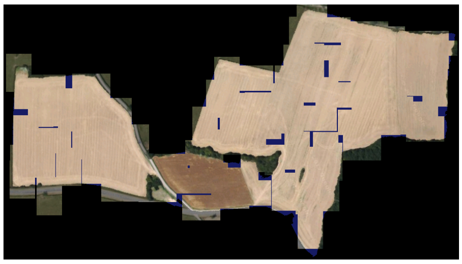
In order to cover the big area, the solution can be to use a bigger focal length or higher altitude to have a wide area covered at each waypoint, thus keep few waypoints to control the area. The other solution is to increase the number of waypoints. The

(a) The area to cover taken form satellite images.



(b) The mask of the area to cover.

(c) The result of the coverage with the waypoints position.



(d) The final mosaic images

Parameters	**Value**
z	[0.6;0,8;1]
γ	portrait or landscape
Size of the area	895x502 px
Maximum size of camera projection	75x50 px
Number of waypoints	80

FIGURE 5.8 – Optimization of the waypoints poses with a vast outside area and just a few black holes.

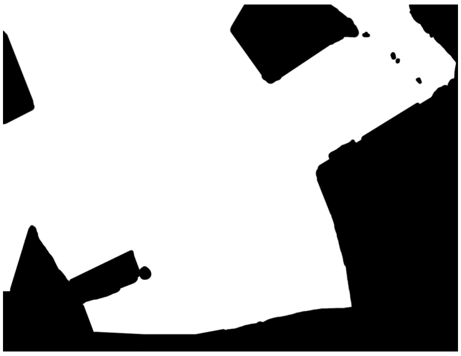
increased number of waypoints can be a source of difficulty for the optimization. Although the difficulty to manage more waypoints to the GAPSO associate to the adapted cost function allows the example Figure 5.8c more waypoints to be placed and optimized. In the Figure 5.8c and 5.8d the area is covered by 80 waypoints for 98.48% of coverage. To reach this coverage rate the GA convergence is achieved after 4'856 generations. The important number of necessary generation before the convergence of the GA and a similar increasing time computation for the PSO, allow us to glimpse the limits of the GAPSO for the too big search space with numerous waypoints to pose estimate. Despite this potential future limitation the answer of the GAPSO is relatively fast and efficient.

5.4.4.2/ Biggest map with numerous waypoints

The precedent experiment (see 5.4.4.1) shows the efficiency and flexibility of the GAPSO to the big map with lots of waypoint in a really complex area. Despite an important increasing time of computation the GAPSO can use to go even more further and try to touch the boundary.

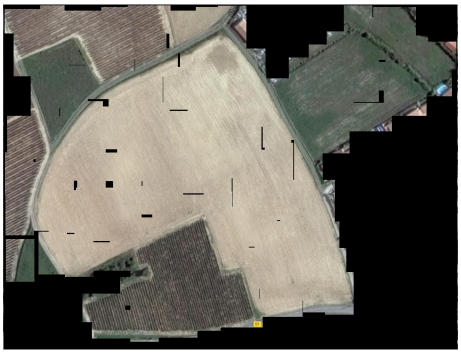
For this experiment, the area proposed here is the biggest never tested yet (among the one presented). The grid is composed of almost 1 million of points, with more than 616

(a) The area to cover taken form satellite images.



(b) The mask of the area to cover.

(c) The result of the coverage with the waypoints position.



(d) The final mosaic images

Parameters	**Value**
z	[1;1.5;1.75]
γ	portrait or landscape
Size of the area	1113x848 px
Maximum size of camera projection	69x104 px
Number of waypoints	110

FIGURE 5.9 – Optimization of the waypoint pose with a big outside area : (a) is the area to cover take form a satellite images,(b) is a mask of the area to cover, (c) is a result of the coverage with the waypoint position, (d) is the representation of the black hole.

thousands points to cover. The number of waypoints to cover this vast area, has been increased to reach the 110 waypoints. The number of waypoints to pose estimate, is the among more important compared to the literature for example, in [91, 58, 12, 44, 88, 76, 35, 24, 77, 17]. The GAPSO is executed with success and the final coverage is over 98.26%.

To reach this coverage rate the GA convergence is achieved after 170'501 generations. The important number of generation before to reach the convergence with the GA in a first time, and proceed to a PSO optimization for a second time reveals a long time computation before the final solution given by the GAPSO. That show at the same time the great efficiency of GAPSO to optimize the position for numerous waypoints in a big search space with proposing a really good answer. This great optimization is conditionally upon for an important time of convergence. This show the limit of the GAPSO due to the important number of generation and consequently important time computation (a few hours with a core i7). To nuanced the really important time useful to reach the double convergence (GA and PSO) the context of the experiment has to be highlighted. In Fact the area to cover is important but also the coverage rate respected is also an important factor because more this coverage rate is important more a fine tuning of the numerous parameters of the solution must be done.

5.4.5/ Waypoints positioning limitation

Among numerous experiments done, the GAPSO appear as a good solution to optimize the position and orientation of numerous waypoints (or cameras) in a vast and complex map. In the first time the GA appear efficient enough for the optimization in the big room but after more experimentation the single GA appears weak to finally refine the solution. The contribution of PSO was essential to have a more refine solution and also allow more flexibility especially when the number of waypoints is restricted.
During the differents experiment proposed some limitation appear. Among the limitation the more important is mostly the consequences of a high numbers of waypoints to have a high coverage rate. The size of the area and moreover the number of waypoints to pose estimate has an important impact on the time convergence of the GAPSO. This time convergence increase even more when the number of waypoints is hight for have a high coverage rate.
To illustrate this phenomenon the a GAPSO ran on the last experiment (same map) but with less waypoints(only 80). Consequently the coverage rate is also smaller around 85.5% of coverage for the 80 waypoints. In this condition, the number of generation for the GA before to converge are just around 200 generation. This 200 generations for the GA and a similar number of iteration for the PSO appears as a huge difference between the 98.26% of coverage of the 110 waypoints in 170'501 generations. This huge difference is due at the same time to the number of dimension to optimize and the fine refinement due to the high coverage rate. Thanks to these observations it is appearing the main limitation of the GAPSO is not the big or complex area, but is mostly the high number of waypoints (or number of dimension to optimize) and the high coverage rate with involved a fine refinement of each waypoints position.

6

Cover path planning project

Sommaire

6.1/ Splitting the problem

Splitting the problem

6.1.1/ Positioning of the waypoint

6.1.2/ Number of waypoint

6.1.3/ Sorted waypoint

6.1.3.1/ Graph theory multi objective

6.1.3.2/ TSP solution

6.1.3.3/ GA

6.1.4/ Experiment

6.2/ Use GA with non splitting in sub problem

6.2.1/ Explosion de la complexité

6.2.2/ Result

Use GA with non splitting in sub problem Explosion de la complexité Result

6.3/ Proposer une définition

La définition 1 illustre la proposition d'une définition.

Définition 1 : Une thèse

Ouvrage présenté devant un jury universitaire pour l'obtention d'un doctorat.

6.4/ Inclure un tableau

Le référencement de la table peut être réalisé à l'aide des macros :

```
\tabref{labelid}
\tabpageref{labelid}
```

6.4.1/ Exemple 2

La table 6.1 est un exemple de table avec 5 colonnes, et dans laquelle le titre de la table a été également ajouté en sommet.

Titre de la table				
Col1	*Col2*	*Col3*	*Col4*	*Col5*
a	b	c	d	x
e	f	g	h	z

Table 6.1 – Titre de la table

Vous pouvez placer un texte $^{\text{en exposant}}$. Vous pouvez placer un texte $_{\text{en indice}}$.

Vous pouvez mettre en avant un texte, ou le mettre **encore plus en avant**.

Vous pouvez formatter les noms de personnes de manière uniforme, comme par exemple Stéphane Galland

6.5/ Détails de la contribution

6.6/ Conclusion

6.7/ Bilan

6.8/ Perpectives

Bibliographie

[1] Akbarzadeh, V., Gagné, C., Parizeau, M., Argany, M., et Mostafavi, M. A. (2013). **Probabilistic sensing model for sensor placement optimization based on line-of-sight coverage**. *IEEE Transactions on Instrumentation and Measurement*, 62(2) :293–303.

[2] Alander, J. T. (1994). **An indexed bibliography of genetic algorithms papers of 1996 (in proceedings)**. Rapport technique, Report 94-1-96PROC, University of Vaasa, Department of Information Technology and Production Economics, 1995.(ftp ://ftp. uwasa. fics/report94-1/ga96PROCbib. ps. Z) ga96PROCbib.

[3] Arabas, J., Michalewicz, Z., et Mulawka, J. (1994). **Gavaps-a genetic algorithm with varying population size**. Dans *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 73–78. IEEE.

[4] Bodor, R., Drenner, A., Janssen, M., Schrater, P., et Papanikolopoulos, N. (2005a). **Mobile camera positioning to optimize the observability of human activity recognition tasks**. Dans *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1564–1569. IEEE.

[5] Bodor, R., Schrater, P., et Papanikolopoulos, N. (2005b). **Multi-camera positioning to optimize task observability**. Dans *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, pages 552–557. IEEE.

[6] Boeringer, D. W., et Werner, D. H. (2004). **Particle swarm optimization versus genetic algorithms for phased array synthesis**. *IEEE Transactions on antennas and propagation*, 52(3) :771–779.

[7] Box, G. E. (1957). **Evolutionary operation : A method for increasing industrial productivity**. *Applied statistics*, pages 81–101.

[8] Bremermann, H. J. (1962). **Optimization through evolution and recombination**. *Self-organizing systems*, 93 :106.

[9] Bulusu, N., Estrin, D., Girod, L., et Heidemann, J. (2001). **Scalable coordination for wireless sensor networks : self-configuring localization systems**. Dans *International Symposium on Communication Theory and Applications (ISCTA 2001), Ambleside, UK*.

[10] Cerf, R. (1995). **An asymptotic theory for genetic algorithms**. Dans *European Conference on Artificial Evolution*, pages 35–53. Springer.

[11] Chakrabarty, K., Iyengar, S. S., Qi, H., et Cho, E. (2002). **Grid coverage for surveillance and target location in distributed sensor networks**. *IEEE transactions on computers*, 51(12) :1448–1453.

[12] Chrysostomou, D., et Gasteratos, A. (2012). **Optimum multi-camera arrangement using a bee colony algorithm**. Dans *Imaging Systems and Techniques (IST), 2012 IEEE International Conference on*, pages 387–392. IEEE.

[13] Chvatal, V. (1975). **A combinatorial theorem in plane geometry**. *Journal of Combinatorial Theory, Series B*, 18(1) :39–41.

[14] Couto, M. C., de Rezende, P. J., et de Souza, C. C. (2011). **An exact algorithm for minimizing vertex guards on art galleries**. *International Transactions in Operational Research*, 18(4) :425–448.

[15] Deb, K., Agrawal, S., Pratap, A., et Meyarivan, T. (2000). **A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization : Nsga-ii**. Dans *International Conference on Parallel Problem Solving From Nature*, pages 849–858. Springer.

[16] Di Franco, C., et Buttazzo, G. (2016). **Coverage path planning for uavs photogrammetry with energy and resolution constraints**. *Journal of Intelligent & Robotic Systems*, 83(3-4) :445–462.

[17] Ding, C., Song, B., Morye, A., Farrell, J. A., et Roy-Chowdhury, A. K. (2012). **Collaborative sensing in a distributed ptz camera network**. *IEEE Transactions on Image Processing*, 21(7) :3282–3295.

[18] Eiben, A. E., Smith, J. E., et others (2003). **Introduction to evolutionary computing**, volume 53. Springer.

[19] Erdem, U. M., et Sclaroff, S. (2006). **Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements**. *Computer Vision and Image Understanding*, 103(3) :156–169.

[20] Fleishman, S., Cohen-Or, D., et Lischinski, D. (2000). **Automatic camera placement for image-based modeling**. Dans *Computer Graphics Forum*, volume 19, pages 101–110. Wiley Online Library.

[21] François, O., et Lavergne, C. (2001). **Design of evolutionary algorithms-a statistical perspective**. *IEEE Transactions on Evolutionary Computation*, 5(2) :129–148.

[22] Friedberg, R. M. (1958). **A learning machine : Part i**. *IBM Journal of Research and Development*, 2(1) :2–13.

[23] Fu, X., WANG, R.-c., SUN, L.-j., et Shuai, W. (2010). **Research on the three-dimensional perception model and coverage-enhancing algorithm for wireless multimedia sensor networks**. *The Journal of China Universities of Posts and Telecommunications*, 17 :67–72.

[24] Fu, Y.-G., Zhou, J., et Deng, L. (2014). **Surveillance of a 2d plane area with 3d deployed cameras**. *Sensors*, 14(2) :1988–2011.

[25] Galceran, E., et Carreras, M. (2013). **A survey on coverage path planning for robotics**. *Robotics and Autonomous Systems*, 61(12) :1258–1276.

[26] Goldberg, D. E. (1985). **Optimal initial population size for binary-coded genetic algorithms**. Clearinghouse for Genetic Algorithms, Department of Engineering Mechanics, University of Alabama.

[27] Goldberg, D. E. (1989). **Genetic algorithms and walsh functions : Part i, a gentle introduction**. *Complex systems*, 3(2) :129–152.

[28] Grefenstette, J. J. (1986). **Optimization of control parameters for genetic algorithms**. *IEEE Transactions on systems, man, and cybernetics*, 16(1) :122–128.

[29] Han, K.-H., et Kim, J.-H. (2000). **Genetic quantum algorithm and its application to combinatorial optimization problem**. Dans *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 2, pages 1354–1360. IEEE.

[30] Han, K.-H., et Kim, J.-H. (2002). **Quantum-inspired evolutionary algorithm for a class of combinatorial optimization**. *IEEE transactions on evolutionary computation*, 6(6) :580–593.

[31] Herrera, J. L. A., et Chen, X. (2012). **Online configuration of ptz camera networks**. Dans *Distributed Smart Cameras (ICDSC), 2012 Sixth International Conference on*, pages 1–6. IEEE.

[32] Holland, J. H. (1962). **Outline for a logical theory of adaptive systems**. *Journal of the ACM (JACM)*, 9(3) :297–314.

[33] Hoppe, C., Wendel, A., Zollmann, S., Pirker, K., Irschara, A., Bischof, H., et Kluckner, S. (2012). **Photogrammetric camera network design for micro aerial vehicles**. Dans *Computer vision winter workshop (CVWW)*, volume 8, pages 1–3.

[34] Hörster, E., et Lienhart, R. (2006). **On the optimal placement of multiple visual sensors**. Dans *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pages 111–120. ACM.

[35] Jiang, Y., Yang, J., Chen, W., et Wang, W. (2010). **A coverage enhancement method of directional sensor network based on genetic algorithm for occlusion-free surveillance**. Dans *Computational Aspects of Social Networks (CASoN), 2010 International Conference on*, pages 311–314. IEEE.

[36] Kulkarni, R. V., et Venayagamoorthy, G. K. (2011). **Particle swarm optimization in wireless-sensor networks : A brief survey**. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(2) :262–267.

[37] Li, N., et Marden, J. R. (2013). **Designing games for distributed optimization**. *IEEE Journal of Selected Topics in Signal Processing*, 7(2) :230–242.

[38] Li, Y., Chen, H., Er, M. J., et Wang, X. (2011). **Coverage path planning for uavs based on enhanced exact cellular decomposition method**. *Mechatronics*, 21(5) :876–885.

[39] Liu, L., Zhang, X., et Ma, H. (2009). **Dynamic node collaboration for mobile target tracking in wireless camera sensor networks**. Dans *INFOCOM 2009, IEEE*, pages 1188–1196. IEEE.

[40] Liu, L., Zhang, X., et Ma, H. (2010). **Optimal node selection for target localization in wireless camera sensor networks**. *IEEE Transactions on Vehicular Technology*, 59(7) :3562–3576.

[41] Lu, X.-S., Huang, H.-J., et Long, J. (2011). **A bi-level programming model for network traffic surveillance of optimal camera location**. Dans *Computational Sciences and Optimization (CSO), 2011 Fourth International Joint Conference on*, pages 1035–1039. IEEE.

[42] Ma, H., Yang, M., Li, D., Hong, Y., et Chen, W. (2012). **Minimum camera barrier coverage in wireless camera sensor networks**. Dans *INFOCOM, 2012 Proceedings IEEE*, pages 217–225. IEEE.

[43] Mais, H.-R., Bloch, C., Ramdane-Cherif, W., et Chatonnay, P. (2010). **Différentes opérateurs évolutionnaires de permutation : sélections, croisements et mutations**. *LABORATOIRE D'INFORMATIQUE DE L'UNIVERSITE DE FRANCHE-COMTE*.

[44] Maji, K., Ghosh, A., Kar, R., Mandal, D., et Ghoshal, S. (2015). **An evolutionary algorithm based approach for vlsi floor-planning**. Dans *Science and Technology (TICST), 2015 International Conference on*, pages 248–253. IEEE.

[45] Matsui, K. (1999). **New selection method to improve the population diversity in genetic algorithms**. Dans *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*, volume 1, pages 625–630. IEEE.

[46] Mavrinac, A., et Chen, X. (2013). **Modeling coverage in camera networks : A survey**. *International journal of computer vision*, 101(1) :205–226.

[47] Medhi, N., et Pal, M. (2013). **Sixsoid : A new paradigm for k-coverage in 3d wireless sensor networks**. *arXiv preprint arXiv :1401.0200*.

[48] Meiting, W., Junjian, T. S. D., et Liwen, Y. (2007). **Complete coverage path planning of wall-cleaning robot using visual sensor**. Dans *Electronic Measurement and Instruments, 2007. ICEMI'07. 8th International Conference on*, pages 4–159. IEEE.

[49] Miller, B. L., Goldberg, D. E., et others (1995). **Genetic algorithms, tournament selection, and the effects of noise**. *Complex systems*, 9(3) :193–212.

[Moeini et al.] Moeini, M., Kröller, A., et Schmidt, C. **Une nouvelle approche pour la résolution du problème de la galerie d'art**.

[51] Morsly, Y., Aouf, N., Djouadi, M. S., et Richardson, M. (2012). **Particle swarm optimization inspired probability algorithm for optimal camera network placement**. *IEEE Sensors Journal*, 12(5) :1402–1412.

[52] Mühlenbein, H. (1989). **Parallel genetic algorithms, population genetics and combinatorial optimization**. Dans *Workshop on Parallel Processing : Logic, Organization, and Technology*, pages 398–406. Springer.

[53] Nikolaidis, S., Ueda, R., Hayashi, A., et Arai, T. (2009). **Optimal camera placement considering mobile robot trajectory**. Dans *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 1393–1396. IEEE.

[54] O'rourke, J. (1987). **Art gallery theorems and algorithms**, volume 57. Oxford University Press Oxford.

[55] Owais, S., Snasel, V., Kromer, P., et Abraham, A. (2008). **Survey : using genetic algorithm approach in intrusion detection systems techniques**. Dans *Computer Information Systems and Industrial Management Applications, 2008. CISIM'08. 7th*, pages 300–307. IEEE.

[56] Packer, E. (2008). **Computing multiple watchman routes**. *Lecture Notes in Computer Science*, 5038 :114–128.

[57] Rechenberg, I. (1965). **Cybernetic solution path of an experimental problem**.

[58] Reddy, K. K., et Conci, N. (2012). **Camera positioning for global and local coverage optimization**. Dans *Distributed Smart Cameras (ICDSC), 2012 Sixth International Conference on*, pages 1–6. IEEE.

[59] Robbins, H., et Monro, S. (1951). **A stochastic approximation method**. *The annals of mathematical statistics*, pages 400–407.

[60] Ronald, S. (1997). **Robust encodings in genetic algorithms : A survey of encoding issues**. Dans *Evolutionary Computation, 1997., IEEE International Conference on*, pages 43–48. IEEE.

[61] Schwefel, H.-P. (1984). **Evolution strategies : A family of non-linear optimization techniques based on imitating some principles of organic evolution**. *Annals of Operations Research*, 1(2) :165–167.

[62] Serpell, M., et Smith, J. E. (2010). **Self-adaptation of mutation operator and probability for permutation representations in genetic algorithms**. *Evolutionary Computation*, 18(3) :491–514.

[63] Sheikh, R. H., Raghuwanshi, M. M., et Jaiswal, A. N. (2008). **Genetic algorithm based clustering : a survey**. Dans *Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on*, pages 314–319. IEEE.

[64] Shi, X., Liang, Y., Lee, H., Lu, C., et Wang, L. (2005). **An improved ga and a novel pso-ga-based hybrid algorithm**. *Information Processing Letters*, 93(5) :255–261.

[65] Sohrabi, K., Gao, J., Ailawadhi, V., et Pottie, G. J. (2000). **Protocols for self-organization of a wireless sensor network**. *IEEE personal communications*, 7(5) :16–27.

[66] Song, B., Soto, C., Roy-Chowdhury, A. K., et Farrell, J. A. (2008). **Decentralized camera network control using game theory**. Dans *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pages 1–8. IEEE.

[67] Soremekun, G., Gürdal, Z., Haftka, R., et Watson, L. (2001). **Composite laminate design optimization by genetic algorithm with generalized elitist selection**. *Computers & structures*, 79(2) :131–143.

[68] Soto, C., Song, B., et Roy-Chowdhury, A. K. (2009). **Distributed multi-target tracking in a self-configuring camera network**. Dans *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1486–1493. IEEE.

[69] Srinivas, M., et Patnaik, L. M. (1994a). **Adaptive probabilities of crossover and mutation in genetic algorithms**. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(4) :656–667.

[70] Srinivas, M., et Patnaik, L. M. (1994b). **Genetic algorithms : A survey**. *computer*, 27(6) :17–26.

[71] Thierens, D., et Goldberg, D. (1994). **Elitist recombination : An integrated selection recombination ga**. Dans *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 508–512. IEEE.

[72] Topcuoglu, H. R., Ermis, M., et Sifyan, M. (2009). **Hybrid evolutionary algorithms for sensor placement on a 3d terrain**. Dans *Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on*, pages 511–516. IEEE.

[73] Valente, J., Sanz, D., Del Cerro, J., Barrientos, A., et de Frutos, M. Á. (2013). **Near-optimal coverage trajectories for image mosaicing using a mini quad-rotor over irregular-shaped fields**. *Precision agriculture*, 14(1) :115–132.

[74] van den Hengel, A., Hill, R., Ward, B., Cichowski, A., Detmold, H., Madden, C., Dick, A., et Bastian, J. (2009). **Automatic camera placement for large scale surveillance networks**. Dans *Applications of Computer Vision (WACV), 2009 Workshop on*, pages 1–6. IEEE.

[75] Walters, G. A., et Smith, D. K. (1995). **Evolutionary design algorithm for optimal layout of tree networks**. *Engineering Optimization*, 24(4) :261–281.

[76] Wang, C., Qi, F., et Shi, G.-M. (2009). **Nodes placement for optimizing coverage of visual sensor networks**. *Advances in Multimedia Information Processing-PCM 2009*, pages 1144–1149.

[77] Wang, M., Liu, X., Zhang, Y., et Wang, Z. (2017). **Camera coverage estimation based on multistage grid subdivision**. *ISPRS International Journal of Geo-Information*, 6(4) :110.

[78] Wang, Y., Dang, G., Si, Y., et Zhou, H. (2008). **An inversion propagation model using ga for coverage prediction of a single urban cell in wireless network**. Dans *Communications, 2008. ICC'08. IEEE International Conference on*, pages 4894–4898. IEEE.

[79] Wang, Y.-K., et Fan, K.-C. (1996). **Applying genetic algorithms on pattern recognition : An analysis and survey**. Dans *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 2, pages 740–744. IEEE.

[80] Whitley, D. (1994). **A genetic algorithm tutorial**. *Statistics and computing*, 4(2) :65–85.

[81] Wright, A. H., et others (1991). **Genetic algorithms for real parameter optimization**. *Foundations of genetic algorithms*, 1 :205–218.

[82] Wu, D., Ci, S., Luo, H., Ye, Y., et Wang, H. (2011). **Video surveillance over wireless sensor and actuator networks using active cameras**. *IEEE Transactions on Automatic Control*, 56(10) :2467–2472.

[83] Xu, Y.-C., Lei, B., et Hendriks, E. A. (2011). **Camera network coverage improving by particle swarm optimization**. *EURASIP Journal on Image and Video Processing*, 2011(1) :458283.

[84] Yabuta, K., et Kitazawa, H. (2008). **Optimum camera placement considering camera specification for security monitoring**. Dans *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 2114–2117. IEEE.

[85] Zhang, Q., et Li, H. (2007). **Moea/d : A multiobjective evolutionary algorithm based on decomposition**. *IEEE Transactions on evolutionary computation*, 11(6) :712–731.

[86] Zhang, Z., Willson, J., Lu, Z., Wu, W., Zhu, X., et Du, D.-Z. (2016). **Approximating maximum lifetime k-coverage through minimizing weighted k-cover in homogeneous wireless sensor networks**. *IEEE/ACM Transactions on Networking*, 24(6) :3620–3633.

[87] Zhao, J., Cheung, S.-C., et Nguyen, T. (2008). **Optimal camera network configurations for visual tagging**. *IEEE Journal of Selected Topics in Signal Processing*, 2(4) :464–479.

[88] Zhao, J., Yoshida, R., Cheung, S.-c. S., et Haws, D. (2013). **Approximate techniques in solving optimal camera placement problems**. *International Journal of Distributed Sensor Networks*, 9(11) :241913.

[89] Zhong, W., Liu, J., Xue, M., et Jiao, L. (2004). **A multiagent genetic algorithm for global numerical optimization**. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(2) :1128–1141.

[90] Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., et Zhang, Q. (2011a). **Multiobjective evolutionary algorithms : A survey of the state of the art**. *Swarm and Evolutionary Computation*, 1(1) :32–49.

[91] Zhou, P., et Long, C. (2011b). **Optimal coverage of camera networks using pso algorithm**. Dans *Image and Signal Processing (CISP), 2011 4th International Congress on*, volume 4, pages 2084–2088. IEEE.

TABLE DES FIGURES

Liste des tables

LISTE DES DÉFINITIONS

II

ANNEXES

A

Premier chapitre des annexes

Abstract:

This is the abstract in English abstract of the thesis.

Keywords: Keyword 1, Keyword 2

SPIM

■ École doctorale SPIM - Université de Bourgogne/UFR ST BP 47870 F - 21078 Dijon cedex

■ tél. +33 (0)3 80 39 59 10 ■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr

UNIVERSITÉ DE BOURGOGNE