SPIM

Thèse de Doctorat

This is the one

is the one

DAVID STRUB

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 my family to have always trust in my ability despite the reality :) ✡✌👁✖✔ ✡ ❤

Contents

1

INTRODUCTION

1.1/ CONTEXT AND MOTIVATION

do some things!

2

State Of Art

Contents

The surveillance and control domain is a wide field of research which contains a lot of aspects such as: object tracking, object recognition, 3D mapping and area coverage among others. Our work focused specifically on the latter, i.e. finding a procedure which allows to capture a minimal number of images of a given area maximising its coverage. This can be achieved using a set of visual sensors. Many parameters have thus to be taken into account: size and shape of the area itself, field of view of the cameras, number of cameras (or views) to name a few. The following chapter will survey the different methods and techniques proposed in the literature to solve this problem. Keeping in mind that our goal is to find out a technique that works both indoor and outdoor, flexible enough to handle two scenarios: (1) a set of cameras observing simultaneously the area; (2) a single camera moving along a pre-computed path to cover the area.

2.1/ CAMERA POSITIONING

The first step for solving the area coverage problem is the camera positioning, i.e. how to estimate an optimal viewpoint selection to ensure an acceptable coverage. It is known that an efficient camera positioning is a bottleneck in many applications, as for example in the video surveillance field [1, 2, 3, 4, 5], where an efficient camera positioning is essential to monitor correctly an area. The following section will deal with the question of:

- What is a good position and orientation for a camera (i.e. a good camera pose)?
- What are the purposes of the application requiring camera positioning?

These questions have already been investigated and some solutions have been proposed in the literature.

2.1.1/ EFFICIENT POSE IN A CAMERA NETWORK: CHALLENGES AND OBJECTIVES

The first point to address is defining the pose of the camera (or set of cameras). In computer vision, the pose of a camera is composed of its position in space and orientation (or looking direction), i.e. of 3 translations and 3 rotations in a world coordinate frame. The second point to be addressed is how to define a ”good” or ”optimal” camera pose(s)? To do so, it is essential to identify the purposes, tasks and priorities of the application:

- what is the final goal?
- what are the important features for the application (e.g. to track an object, to have a high-resolution mapping, etc.)?
- what are the shooting conditions and physical constraints?

All of these aspects will have an incidence on the definition and formalization of a ”good camera poses”. For instance, in [6], the purpose is to detect tags placed on people torso which forces the camera to be positioned at a certain height with a looking direction almost parallel to the ground; on the contrary, in [7], a camera is mounted on a UAV to monitor a

vast outdoor area, which forces it to have a looking direction almost perpendicular to the ground.

These two articles share the same objective, i.e. to get the best possible coverage of an area, but since the constraints and secondary objectives they must comply with are different (e.g. number of cameras, resolution, luminosity, tracking, etc.), it leads to different formulation and approach.

The next section focuses on positioning the camera to maximize viewing areas. The viewing area or coverage rate of the area is directly related to the estimated position of each camera and their orientation. To get the best coverage, it is essential to find the best pose for each camera, depending on the constraints and possible secondary goals.

Camera positioning for maximizing the coverage rate has been studied these past decades, using many different approaches. The following sub-section will provide an overview of different ways of defining the coverage drawn from the literature.

2.1.2/ OBJECTIVE: COVERAGE

Before formalising the coverage rate to be maximised, it is necessary to define exactly what is meant by coverage: is the aim to maximise the surface area of a three-dimensional object or a 2D zone whose perimeter has been circumscribed? Is there a priori knowledge of the area to be covered (its perimeter, its bounding box, etc.) or not? Is the area to be covered homogeneous or does it contain prioritized sub-areas?

- Object coverage:
  In Hoppe et al. [8] a good coverage is defined by the ability to have full 3D reconstruction of an object (in their 3 dimensions) with no occlusion. In this work, some prior knowledge on the object is exploited such as a rough surface description (mesh). The camera follows a trajectory ”around” the object and its looking direction is oriented towards the center of the mesh (see Figure 2.1). Since it is a matter of covering the 3D surface of an object, in a next-best-view strategy, this application remains too far from the one we wish to implement. It is therefore barely applicable to our problem.

- Path to cover:
  The point here is to observe the entire trajectory commonly taken by users (car, pedestrian, . . .). When the area to cover is a well-known place, the main trajectory taken by the users can be estimated or extracted [10]. If the area to cover is a road, for instance, then the trajectory of the driver is known [11]. In this condition, the aim is to cover the common trajectory of the user as presented in [11, 10, 12, 9] (see the Figure 2.2). The path coverage is interesting due to the restricted area to cover: not an entire area, but only a path within a given area, that can be seen as a priority sub-area.

- Coverage priority:
  A natural way of defining the coverage in a context of insufficient number of cameras, is to define as priority. In [5, 13, 14], some predefined regions are set as

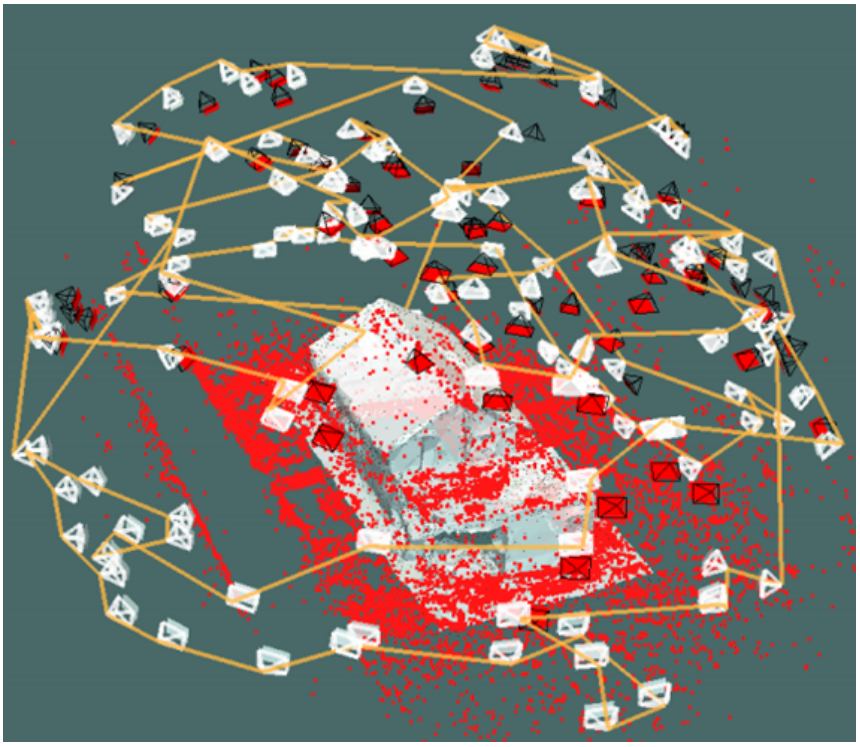


Figure 2.1: Full coverage of an object in the 3D space. The coverage is made by selecting a set of adapted waypoints. the coverage must be good enough to can reconstruct the 3D shape of the object without any occlusion. This result is from Hoppe et al. [8].

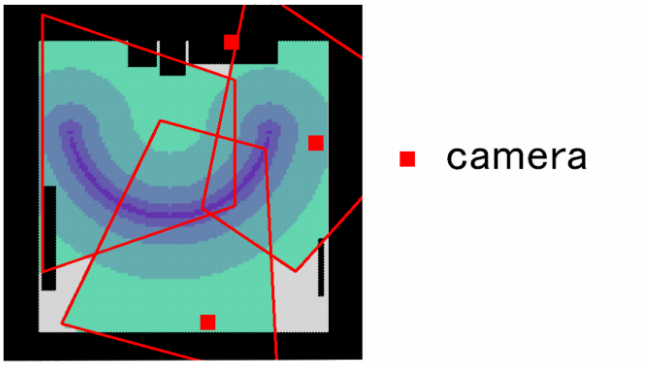


Figure 2.2: The path to cover is illustrate in the work of Nikolaidis et al. [9]. The aim is to is focused on cover a road (walk path) in a small room by using 3 cameras.

”priority” and called respectively ”region of interest”, ”crucial sub-area” (see Figure 2.3) and ”importance space weighting”. In the solutions proposed by [5, 13, 14], the camera poses are in priority affected to this specific and restricted region which has the effect to neglect the other parts of the area.
If the environment is composed of some regions of interest, there should be also ”normal” sub-areas. These ”normal sub-areas” should be covered, but with lower priority. Furthermore, some sub-areas can be defined as ”no interest”, which

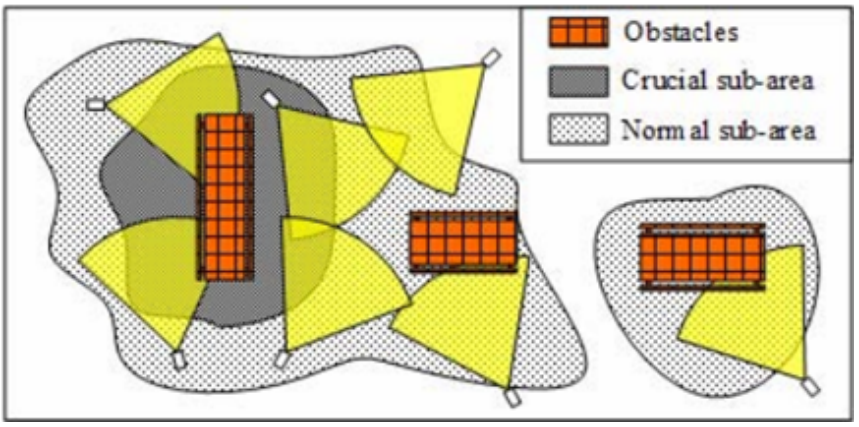


Figure 2.3: Map of an area to cover with crucial sub-area (region of interest) the normal sub-area and obstacle. This map is an example of area coverage introduce in Jiang et al. [13].

mean ”not to be covered”. In [13, 14] for example, the obstacles are defined as ”no interest” regions with also the consequence to be occluding area. The idea is to keep a maximum of freedom in the camera network positioning and allow the system to handle local priority and constraints.

- Inside or outside area:
  Another important feature to define the coverage is related to indoor/outdoor scenes. The area to cover can be typically a room with walls (indoor). Each wall must be a considered as an obstacle occluding the camera field of view, which results in having to manage the visibility of the environment according to these obstacles and to the position of the cameras. For outdoor scenes, it is often necessary to take into account the size of the environment according to the reduced field of view of the camera. This has the effect of increasing the number of required cameras (or views) and leveraging the combinatory.

The common points in all the examples discussed is the aim of maximizing the coverage rate. The positioning of the cameras, and its effects on the coverage rate, is thus constrained by the application itself, the context and the type of observed scenes.
Of course, additional constraints can have also a significant impact on the camera pose. Some of the more common constraints found in the literature are listed below:

- The numbers of cameras:
  In many cases, the number of cameras used for the coverage should be minimised such as in [4, 14, 6]. Limiting the number of cameras is primordial to decrease the computation time and the bandwidth. It also reduces the cost of the setup [15]. Reducing the number of cameras and optimising their poses to get an optimal coverage are closely related tasks, not necessarily competing. Too few cameras can shrink the coverage rate by leaving black-holes in some areas of the scene. Too many cameras can result in too much overlap and unnecessary redundancy.

(a) Initial coverage for 5 cameras dedicated to detect the input of target.

(b) The covered area when the objective have to track several target.

Figure 2.4: Illustration of an covered area for tracking target. Experiment form Ding et al. [3]

- Object tracking:
  Constraints can arise by the objective of detecting and localising a given target [3, 2, 16, 17, 18, 6]. In such a case, camera poses must be estimated in order to track one or more targets, and possibly, dynamically adapted. These applications very often require an adaptation of the camera orientation (looking direction) more than its position. This is the reason why they actually use PTZ cameras (Pan, Tilt and Zoom) as in [3, 19, 2] (see Figure [3]). Keeping a full area covered and at the same time tracking efficiently one or more targets can be contradictory. The solution is then the result of a trade-off between coverage and tracking, as in [3] and [19]. In Liu et al. [19] target tracking in a wide area is decomposed in two steps: detection and localisation. Each of these steps is done independently on each camera. Area coverage is essential to detect the targets, less for localisation as the priority is, in this step, to track a target previously detected within the covered area. In the entire camera network used, one camera may be in detection mode while another is in location mode. Obviously, by adding target tracking as a constraint, camera poses and coverage are usually less efficient because of the subset of cameras assigned to the tracking.

- Luminosity and environmental setup:
  Intrinsic image quality is also a constraint that can guide area coverage. The quality of an image can result in sufficient brightness or an almost uniformly distributed histogram, etc. In other words, the captured images must be such that they guarantee a usable signal. For example, Reddy et al. [20] addressed first the coverage problem of a complex area and in second time, target localization. In order to decide

which target must be tracked, the quality of the image is taking into account to avoid dark areas where the target is hardly detectable. In this case, the tracking and coverage trade-off discussed in the previous paragraph is ruled by the image quality.

- Energetic cost:
  Authors suggest to estimate camera positioning or path planning by minimizing a cost function that represents the energy consumption, such as in [19, 21]. For instance, in Lui et al. [19], the objective is to cover most of an area to detect whether a target is entered in or not. In a second time, the target is tracked by smart and autonomous cameras of the network. The set of cameras are randomly distributed in the area and the coverage problem is, in this case, to select the best cameras in order to detect the target. The selection of the cameras is estimated by both maximizing the area coverage and minimizing the energy consumption. The consumption can be obviously reduced by restricting the number of cameras set in detection mode. Indeed, the cameras are more or less power-consuming depending on the activated mode (which can be ”detection”, ”tracking” or ”sleepy”). If we consider now path planning, the energy cost can be represented by the distance between two cameras or views and energy minimization is equivalent to finding the shortest path [22, 23].

- Multi coverage:
  Among the numerous possible constraints, the multi-coverage is interesting (as for example in [24, 4, 25, 26, 27]). It can be seen as a coverage problem where one or a few specific sub-areas must be covered by a minimum of k cameras at the same time, that's why it is also called k-coverage. Multi-coverage does not necessarily mean priority: a sub-area which is to be covered by several cameras is not necessarily a sub-area which must be covered in priority (more details in Section 4.1.1.4). However, mobilizing multiple cameras in a given sub-area means that fewer cameras can be used to cover the rest of the area, which can be compensated by adding cameras, if allowed. On the contrary full-coverage of the area and k-coverage of some sub-areas will conflict and lead to a trade-off.

- Resolution:
  In order to keep or increase the quality of the captured images, a minimum resolution threshold or value can be fixed and used as a constraint [10, 20, 14, 25, 28]. The focal length, the size of the pixel grid or the camera-target distance can served as a measure for the resolution. However, in many applications, it is easier to adapt the distance than to change the focal length (which can be fixed or affect the calibration) or, of course, to change the size of the pixel grid. In most of the cited works, the distance from the target to the camera along the optical axis is therefore used as a measure for the resolution.
  Full coverage will tend to move the cameras at the farther distance (or higher elevation) in order to maximise the field of view. Resolution constraint will impose the cameras to be positioned in a certain range of distance or elevation. Full coverage and resolution constraint will lead to a trade-off between guaranteeing most of the area to be covered and a sufficient image resolution. The trade-off is particularly beneficial when the number of cameras is more important than the optimal need, in

this case, the distance will be reduced and the resolution mechanically increased.
In [20] the problem has been formalized 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.
Here, the depth of view is used to define the range of distances. The focus point and the aperture of the camera will define the optimal distance and range in which the target is optimally focused [29]. Constraining the positioning with the depth of view can be seen somehow as a constraint on the resolution itself.

The constraints are numerous and varied, we just introduced a few of them which seems interesting to us and related to our work. Among them, some are closely related and can be interconnected, they can even be combined as in [20], where the targets coverage, luminosity, resolution, are all associated to find the best camera positions maximizing the area coverage and the tracking target with good visibility condition.
One interesting point to study is the impact of these constraints on the full coverage itself as we have seen that they introduce a trade-off between their own particular goal and the main goal of area coverage but also between themselves. No need to say thus, that the constraints have to be chosen carefully and accordingly weighted as in any multi-objective problems.

2.1.3/ ART GALLERY PROBLEM

The Art Gallery Problem (AGP) is a theoretical and historical problem closely related than the camera positioning. The AGP is commonly cited in the literature as a source of the problem and also used. The AGP paradigm is used to formulate the problem of camera positioning (as example [30, 31, 24]). For these reasons the AGP as to be well understood before to go further.

In the following sections a definition of the AGP is given with a brief historic. Next to this introduction on AGP, some of the more interesting solution proposed are discussed before to see the limits of this paradigm.

2.1.3.1/ 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 guards to cover an art gallery. The particularity of a art gallery is the complexity of the room shape, with many walls 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 assimilated as a polygon P, composed by n vertices $(v_1; v_2; \dots v_n)$. The vertices are linked by n edges $(v_1v_2; \dots; v_{n-1}v_n)$ 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 one of boundary (a wall) of the polygon P, 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 it .
A guard x can have a $360°$ field of view to cover all around him, with no depth of field limitation (except the wall obstacle). Clearly that mean the guard can see and monitor

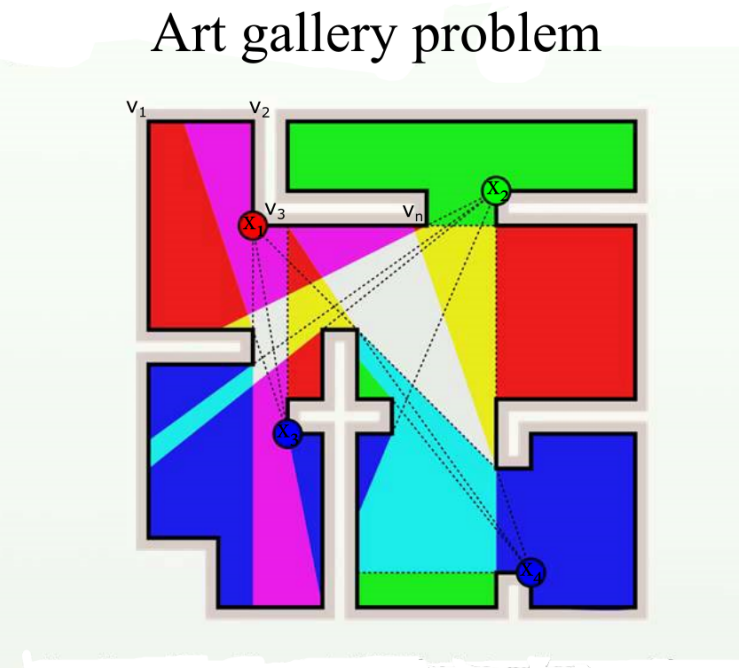


Figure 2.5: Illustration of the AGP. The gallery is cover by 4 guard $(x_1; ...; x_4)$ for a polygon composed by n vertice $(v_1; ...; v_n)$.

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 art gallery (see Figure 2.5).

When the polygon is more complex, it is necessary to estimate the minimum number g of guards x and the position of the guards x in polygon P. The set of minimum number of guards are listed in X. Where X contain the useful x to fully cover the polygons P, with g the minimums number of guard in order to have a set of points $X = \{x_1 \dots x_i \dots . x_g\}$. So that every point y in P are cover by at least on guard x of the set of X.
The AGP in addition to estimating the numbers of guards also are interested on finding the optimal position of this restricted number of guards. This 2 questions can be solved at the same time by using one of the solutions proposed.

2.1.3.2/ SOLUTION

The 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 [30]. 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 guards 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 chronology order as it is recommended in [32]. The book of O'ROURKE et al. [32] is an early work about the AGP with the formulation, proofs and advancement of the field clearly explained.
Once the proof of the minimum number in the worst case founded the objective became to find an optimal guard position for all kind of polygon 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 [33]. The solution finally proposed work in $O(n^3)$ complexity in the worst case.

2.1.3.3/ LIMIT OF AGP AND CAMERA COVERAGE RELATION

The AGP can be considered as a reduction of the best cameras pose estimation to maximize the coverage of a complex area. Based on the algorithm developed to solve the AGP and the strong relation between AGP and the cameras positioning for maximum coverage problem. It is logic to have some proposed algorithm which tries to extend the AGP to the problem of camera positioning as example in [34, 20, 28].
The algorithm developed for AGP cannot be applied directly on the problem of cameras positioning for maximum coverage. The main reasons are the cameras limitations as field of view and depth of field (see [15, 35]). The cameras limitation makes unreadable the algorithm proposed to solve the AGP. Because the AGP considering the guard with no limitation for the depth of field and field of view. Due to these differences the geometric model of AGP may not be applicable for perspective cameras. Also another reason make the AGP solution not applicable for the camera positioning can be the diversity of cameras in the same system. The AGP may have many guard, they are all interchangeable. The interchangeability is due to the guard ability (or skill) to monitor the area. In contrary it is possible to have for a cameras network different kind of cameras with different lens. Finally the perfect assumption for the AGP formulation create important weakness when is time to replace the guards by cameras. These weakness make the algorithms form AGP not adapted in our problem, as is showed in [9, 14].

However, some part of the AGP formulation and especially some proof has their importance, as the proof of Chvàtal [30] or the NP-hard complexity proof. In fact, the AGP is proof as a NP-hard problems [32]. The NP-hard proof is available on the book of O'Rourke section 9.2 of the book [32]. The NP-hard mean the problem cannot be solve in a deterministically way in a reasonable time. To proof the AGP is NP-hard, the first part is to reduce the problem to an other problem well 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 3SAT is used to be exact). 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 another well known problem (for more explication see the book of O'Rourke section 9.3 [32])). Despite the limit of AGP, numerous articles are based on the AGP to formulate the problem as in [28, 31]. For example in [28] a similar approaches then the AGP is used in order to estimate the occluded regions. As explained earlier the cameras positioning can be reduced as an AGP [31] notably by removing most of the constraint due to the

camera properties (as depth of field and field of views).

In the literature numerous articles use the AGP as reference to explain the complexity of the problem as for example [36, 30, 24, 4]. Where their assumes the problem is at least NP-hard or NP-complete. The complexity of the problem will have an impact on the solution used to try to solve it and optimize it.
One other impact of AGP in the problem of camera positioning is the shape of the rooms. In [35, 14, 20, 28] the shape of the room to cover are close then the definition of an art gallery. The art gallery room is a complex polygon composed by many vertices which may occlude the view. This phenomena can be imputed to the link did between the number of vertices and the useful number of guards to cover it. In addition the occlusion formulation made for the AGP is commonly used. The occlusion in AGP is defined by a segment $xy \not\subset P$ where $x \in P$ is the guard position $y \in P$ is a points in the room P. Moreover the use of a complex room inspired by AGP is therefore a good choice in order to verify the effectiveness of the algorithms developed for the cameras positioning in a complex environment.

The AGP can be at same time for part, the historical source of the cameras positioning and give some beginning of 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 algorithms can be found in other related fields.

2.1.4/ WIRELESS SENSOR NETWORK

The Wireless Sensor Network (WSN) can be as AGP considered as inspiration for the problem of cameras positioning to maximize the coverage. The WSN is an active field of research and in many aspects related to the cameras positioning. These sections are focused on the WSN and this relation with the cameras positioning.
To begin:

What is the Wireless Sensor Network (WSN)?

The WSN is a distributed network of sensors or in some case actuators, is also called WSAN (Wierless Sensors and Actuators Network). Each sensor of the network is a relay for the information to the rest of the network. The sensors are at same time the nodes and the relay for the network. The node has to objective to transmit by relaying the information to the other node. The information can be centralized or not :

- When the system is centralized the information has to be transmit node by node until the centralized agent. The computation and the decision about the network is taken by this centralized agent before to be transmit back to the node.
- Otherwise the node have to be the sensors for collect informations and decided to communique with the others nodes depending then the situation. The nodes has to manage alone or with this neighbourhood the informations and computation before to react in consequences.

The information collected by the sensors are vast 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 with antenna positioning [37], military surveillance field [19, 38], airport surveillance [39], video surveillance and tracking [19], environmental monitoring

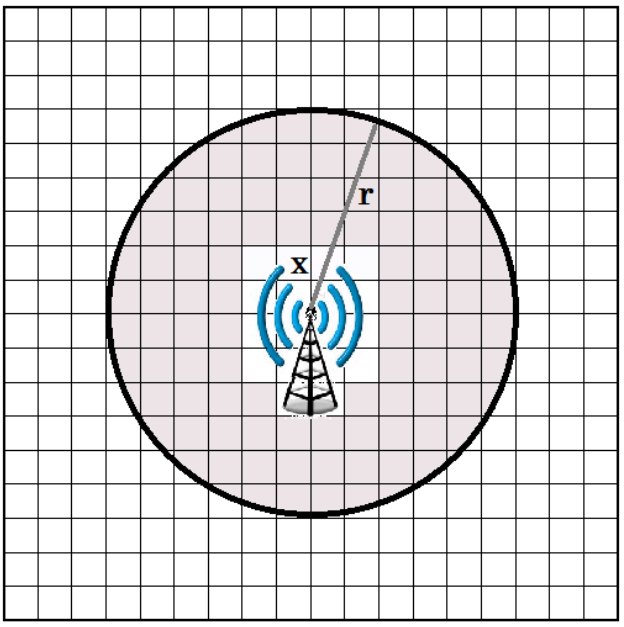


Figure 2.6: One omnidirectional sensor centred on x, with a radius r for the range.

[21]... Logically numerous sensor and information can be collected depending then the need 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 sensors in different aspects as for example [17] focus on an adapted architecture efficient enough for data transfer (here the data are images) or like in [18] the WSN are dedicate to adapt the network around static nodes and energetic resources in order to keep the network connected.
For our case the aspect the more interesting of the WSN, is the coverage of an area with his and secondary objectives. The other discipline of the WSN as the network optimisation will not be addressed in the following document. 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 which has to be taking in account for the optimization.

2.1.4.1/ SENSOR AT 360

The WSN refer commonly to sensors or actuators with no restriction in the view angle, it is considering to have a $360°$ field of view. The sensor Field of view can be represented in a 2D plan as in [40, 26, 41] (as illustrate in the Figure 2.6) and in some case a spherical for the 3D environment example in [27, 37].
Each sensor have a position x in the area and a power range. From the sensor power the radius r of the circle is deduced from x as center (see Figure 2.6). 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 with do not impact the covered area as in [40, 26] (see the Figure 2.7). Others more complex solutions can be used. A more complex solution but also more realistic as in Wang et al. [37]. Where are taking in account the

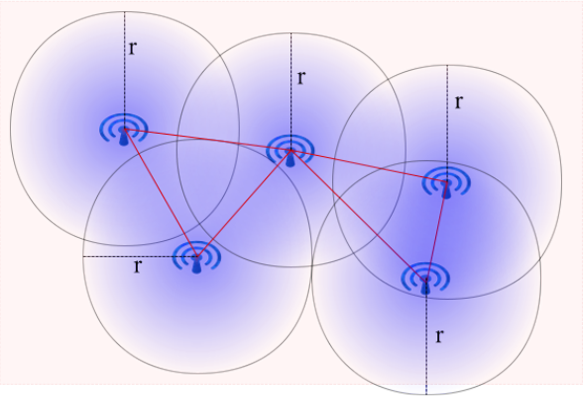


Figure 2.7: Illustration fo a simple wireless sensor network with omnidirectional sensor centred on x, with a radius r.

relief and obstacle. In Zhang et al. [26] 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 radius. The communication radius a bit bigger then the perception radius. These 2 radius correspond to area covered for one antenna (sensors and actuators) and the distance of emission/reception of the data. In order to have an efficient coverage of the area, the antennas 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 two 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 Medhi et al. [27]. This approache 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 zhang et al. [26] 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 cameras 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 families of algorithms focused on optimization. Among the family of algorithms, the evolutionary algorithms (disused in detailed in Chapter 3) is commonly used as in [40, 37], and [41].
  These solutions propose to optimize the position in order to maximize the coverage depending on constraints and the secondary objectives. The method of optimization have to be adapted to the problem. In Chakrabarty et al. [41] the integer linear programming is chosen in order to maximize the coverage with two types of sensors. One standard with a smaller area coverage but with a smaller cost (can have a $100m$ radius for 150$) and the other sensors can cover a wider region (can have

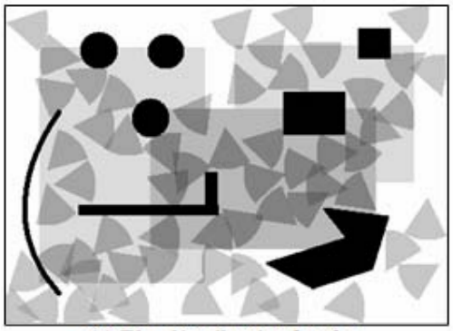


Figure 2.8: Illustration of the area coverage with visual sensors. The area is cover at 47%. Experiment form Jiang et al. [13]

a $200m$ radius for 200$) and the objective is to cover the region with a second objective is to reduce the financial cost. The solution proposed is to use the integer linear programming adapted to the problem of coverage optimization with the financial cost as constraint.
In [37] and [40] the solution proposed is based on two different evolutionary algorithms in order to optimize the sensors positioning. In kulkarni et al. [40] the camera positing with a multi coverage is solved with using an evolutionary algorithm called Particles 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 sensors fail. In Wang et al. [37] one evolutionary algorithms is also used to optimize the position of antennas. The objectives 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 is used in order to find quickly a position for each antenna of the network.

Among the solution proposed, the second, based on optimize a set of sensors position depending then the secondary objectives, is the more interesting and the more flexible to the add of new constraints (and secondary objectives).
The following sections is dedicated to see if these solutions is applicable to the problem of positioning a set of cameras. Despite the camera constraints.

2.1.4.2/ VISUAL SENSORS

Logically, the result obtained with the Omni-directional sensors are interesting. These result must be applied to the problem with even more constraints and objectives for the positioning of Visual Sensor Network (VSN). The term of visual sensor contains several types of cameras with several constraints and properties. Despite these several types of cameras, the more commonly used and studied is the perspective camera (as in [24, 26, 29, 21, 13])(see Figure 2.8). In fact the visual sensor or camera have a limited depth of field but also a limited field of view. This new constraints makes the cameras positioning more complex, as that was for AGP (see Section 2.1.3) to passes from guard to camera. The advantage in this case, is first the sensor has been designed with a limited depth of

field (the sensor has a limited power range). Moreover the solution applied previously for circular sensor was not only geometric or based on heuristic algorithms. The problem has also been formalized as an optimization problem. These problem formulations allow to use optimization algorithms and meta-heuristics. In addition, these problems formulations appear as the more suitable to add constraints and secondary objectives. Thanks to that, the solutions developed for the WSN can mostly be adapted to the problem of cameras positioning. The following sections as to aims to disuse about the algorithms applied until now to the problem of camera positioning.

2.2/ SOLUTION NOT BASED ON EVOLUTIONARY METHOD

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

2.2.1/ THE CONSTRUCTIVE SOLUTION

The cameras pose can be done by construction. By construction that mean a deterministic method is applied to pose iteratively one camera after another or to adjust their positions based an initial set-up.
In Liu et al. [19] a constructive solution is applied in order to select the smart cameras of the network. Each smart camera, are a node of the network to transmit informations and images. The smart cameras are fully autonomous in term of energies and decision-making ability (no central master). The nodes can be in 3 different types of mode:

- 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 computations are 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 active thus energy consumer. The camera is turn on, with a high frame rate and numerous computation has been done to track and localize the targets. Also more informations have to be transmit by the network. The informations are useful to localize the targets by communicate with the potential other smart cameras with may have a view on the targets.

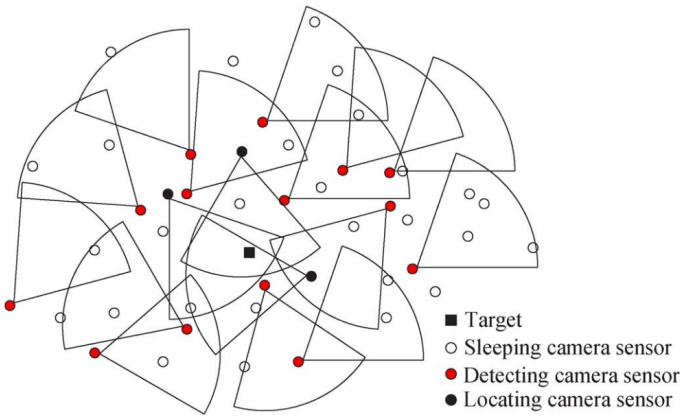


Figure 2.9: Illustration of the area coverage with smart cameras with an target tracking objective. The smart cameras can be in 3 modes (sleepy, detecting , locating). Experiment made in Liu et al. [19]

The objective in [19] 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 cameras in order to maximize the coverage of the cameras turn on detection mode. The solution proposed by Liu et al. in [19] is to use a constructive algorithms. The network of camera self-recognized by a talk between each smart cameras. Also each smart cameras is able to estimate precisely this localization, it is an important assumption to select the cameras. The solution proposed is directly inspired by the distributed network talks.
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 answers have been received, that mean the required density of the camera around are not enough. Thus 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 area is considered covered when the density of camera is good enough.
The method introduce by Liu et al. in [19] is efficient enough. The solution proposed has the advantage to can work in wide area and to be dynamic. As example if a camera do not have any more power the network will self-recognize. This algorithms is 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 and precise (see illustration in Figure 2.9).

Another cameras pose estimation by construction is proposed by Höster et al. [14]. 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 [14] a first greedy search solution has been presented before an other algorithms with extend the greedy solution. The algorithms developed in [14] is called Dual Sampling.
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 non covered yet. The area is discretized by several 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 best cameras positions and orientations depend then the number of other points globally covered in the area. The second step, is repeated and the set of uncovered control points are reduced at each iteration. This procedure is applied until the stopping criterion are reached. That mean enough points of the area are covered.
The constructive solution have some inconvenient notably in terms of efficiency. Indeed this solution is limited by the number of cameras and size of the area due to the exponential difficulty.

In Nikolaidis et al. [9] the cameras placement is studied for cover a basic mobile robot trajectory. The trajectory of the mobile robot is modelled as the regions of interest, with a gradually decreasing interested from the trajectory center.
The solution applied in [9] is to do a local optimization one camera after another with the “steepest decent method (related to the gradient decent)”. If this local optimization give a better solution the network of camera is modified otherwise the cameras stay at the same place. This operation is repeated until the convergent arrangement is obtained or no more upgrade can be found. The result presented in the experiment done by Nikolaidis in [9] are interesting despite the simplicity of the area and the very small amount of cameras used (no more then four). The principal limitation is due to the number of steps required for optimize independently each camera of the network. Also a multitude of local optimisation is not obviously the same or better then the global optimization.

Ma et al. [39] propose a solution for the problem of finding the minimum cameras 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 defined ; if for any direction of the target there always exist a camera to monitor the face of it.
The solution purposed is based on constructive solution with adapted heuristic. The heuristic used is presented in detail in [39]. The global idea is to cut the perimeter in different sub-regions and applies the method proposed to have a full-view coverage at 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 for vast coverage area. The first limitation is the number of cameras use to fully cover the area. The important number of cameras is mainly due to this definition of “full view covered”. That definition imply many overlaps to have the multi directions coverage.

The previous solutions and algorithms presented in [19, 39, 9, 14] are based on constructive methods for the cameras placement and their local optimization. Each camera

is individually placed depending than the network with an iterative process. The iterative process has to chose the best position for each camera of the set. This methods has some consequence, nobly the fast increasing number of iterations required to have a solution good enough for each camera. The time complexity is event more problematic with increasing the size of the area and even more with the number of cameras. Also these solutions are extremely dependent than the formulation and the constraints and cannot be easily adapted to other closely related problems (new constraints or modified objectives). The rigidity of adaptation is mostly due to the use of heuristic design for very specific problems.

2.2.2/ LINEAR PROGRAMMING OPTIMISATION AND LIMITS

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 tested in the literature in order to have a reference point to compare the other more appropriate solutions (as example [4, 15]...). 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 locked in local minima. as presented in the flowing examples.

2.2.2.1/ LINEAR PROGRAMMING

The linear programming is applicable for the linear and convex problems in order to minimize a linear and convex cost function.
In Erdem et al. [28] is based on AGP and the WSN and propose a fusion off their two paradigms in order to use their assumptions. Some modification have been done as example to take in account the field of view limitation. The interesting aspect is how some camera properties have been modelled to fit to the problem of AGP.
The solution proposed being workable with using an omnidirectional or considering a PTZ as omnidirectional cameras by using an efficient angular sweeping. The simile omnidirectional cameras are simulated by PTZ camera with a non-continue zoom as in the experimentation proposed in [28]. Where the PTZ can have two focal lengths at 50mm and 35mm. Finally the solution proposed is to discretize the area to cover and also the different possible parameters for a camera (as: localization, orientation, focal...). In order to have a combinatorial formulation of the problem. Thanks to this formulation close than the Binary Integer Programming (BIP) and apply a well-known method “Branch and Bound” in order to optimize the cameras placement.
This solution proposes a good coverage with the minimum of cameras in a reasonable time. The main limit of the solution is due to the use of omnidirectional or simile omnidirectional cameras.

Zhao et al. [6] are trying to find the optimal position for a set of cameras in order to maximize an indoor area coverage (similar than an AGP room). The coverage of an indoor area is not the only objective, but the requirement to can do an efficient tag detection. The solution proposed for the coverage is to adapt the number of points of interest which must be covered by the camera depending than the coverage rate.

The area is discretized with a grid. The grid is composed by points selected smartly. Each point of the grid simulate a potential location of the target. The camera position is limited at few position fixed on the room boundary. The adapted grid and the limited cameras positioning are used to limit the size of the search space (as number of possible solution). Thanks to this limited search space a linear optimization with the BIP can be used.
Use a BIP is popular and others use it as in [6, 28]. In [6] the solution proposed is to use BIP formulation, the smart sampling of the grid and branch and bound form LP_solve libraries to optimize the cameras positions and orientations.

2.2.2.2/ LIMITATION OF LINEAR METHOD

The methods of linear optimization were applied and tested 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 [42, 4, 15]). 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. [43] propose a solution with an atypical problem formulation. The solution proposed in [43] 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 points. To do that the solution proposed is to decompose the area in order to give more points in the grid was 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 points to describe it. Less points to describe the area to cover mean also a winning time efficiency in computation during the cameras pose estimation (cost function is faster). Despite this interesting solution the result presented in the experiment does not appear to be conclusive.

The principal problem of the linear optimisation appears when the problem became complex. The complexity can come form the problem formulation and some extra constraints. But most of the time the increased complexity is coming from the increased size of the search space. Concretely when the goal is to place a more important number of cameras or when the number of positions and orientations are too important the linear optimisation begin to be inefficient.

2.2.3/ GAME THEORY

Among possible solution an atypical method is to use the game theory [44]. The game theory is use to optimize the looking direction of the cameras as in [2, 3, 44, 45]. These articles are based on game theory to find an equilibrium (also named Nash equilibrium) between two contradictory objectives. The objectives are in one side to maximize the resolution and in the other side the multi target tracking.

Soto et al. [2] propose a network of a dozen of PTZ cameras. That mean the position of the cameras are fix and the solution proposed is to find the best orientations 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 of the images 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. The higher number of target push the cameras 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 [3] and [45] different experiments have been proposed with a number of targets slightly increased. In this article 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 cameras of the network at same time, depending on each target trajectory.
Furthermore, the decentralized solution is more adapted to prevent the security issue, as wrong transitions and interceptions by hostile opponents. The security may be an important factor in many applications.
To have the decentralized system the cameras must have some autonomy.
In the experiment proposed in [2] and [3, 45] each camera are smart enough to have their own tracking and control module. Also the camera are able to communicate with each other, until to reach a consensus. The consensus is when a Nash equilibrium is found between the two 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 have to optimize its simultaneously in order to reach a consensus. The consensus is reached when it became impossible to upgrade one of the objectives without downgraded the other one.
In the experiment of [3, 45] the consensus is found thank the cameras communication. The cameras communication have to maximize numbers of target covered with the higher resolution. The experiment is based on numerous targets moving freely in the area. Also in [3] one of the target must be cover in priority with a high resolution. That have an impact on the other camera position. The result of the experiment as in Figure 2.10 form [3] 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 computations.
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 sensors is not well optimize to cover the area. Moreover to use properly this method for maximum coverage it requires a large number of simulated target in the region to observe (require to have more targets then cameras). Finally this solution is more adapted for a self-reorganization for a set of PTZ cameras.

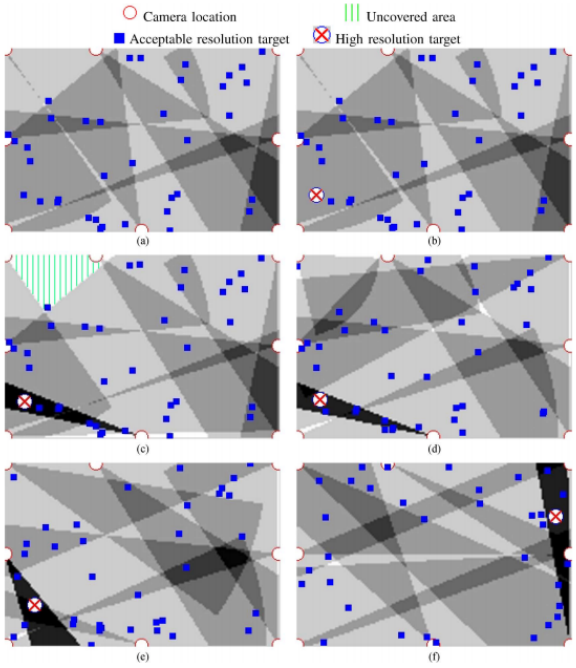


Figure 2.10: Result of coverd area after the game theory omptimization. The objective of the game therory is to maximize the tracking and the resolution. The results shown are from the experiment made in [3].The images represent the iterative sequences for the targets tracking and maximized resolution.

2.2.4/ SUM-UP

In order to summarise the previous section a Table 2.1 is proposed with contain the interesting articles. The first column is for the reference, the author name and the year. The second column is dedicate to the best solution used. In fact some articles has tries few methods only the best is named here. The next columns are dedicated to precise the optimized parameters. The ✔ in the X and Y column indicate if the solutions proposed optimize the cameras position along the X and Y axis. If it is noted by a random dispersion is mean the cameras are randomly dispersed in the map and when the cases are filled with 0 is to indicate non optimization of the position. The columns Pan, Tilt and Roll show with a ✔ the solutions which propose to optimize the rotation of the cameras. The 8^th column are used to precise if the focal lens is optimized or fixed. When the focal lens are optimized but with a discrete range of values the annotation "(discrete)" appear. The 9^th column show the area representation. This representation is manly effective on the grid design. The 10^th column show the maximum number of camera placed and optimized during the experimentations presented in the articles. The last columns is are dedicated to specifies the possible secondary objectives which have to be optimize.

Table 2.1: Sum-up ref

Ref	**Best solution**	X	Y	Pan	Tilt	Roll	Focal lenght	Coverage room	Number of cameras	Secondary objectives and constraints		
[39] Ma 2012	heuristic	✔	✔	✔	0	0	fix	2D	≈ 10	barrier coverage	conectior depen-dance	
[19] Liu 2010	heuristic	random dispersion		0	0	0	fix	2D	≈ 600	resolution	tracking	
[10] Bodor 2005	non-linear	✔	✔	✔	0	0	fix	2D		tracking	resolutio	
[28] Erdem 2006	branch and bound	✔	✔	✔	0	0	✔ (discrete)	omni di-rectional (by pan)	≈ 10	cost reduction	resolutio	DoF
[2] Soto 2009	game theory	0	0	✔	✔	0	✔	2D	≈ 10	resolution	tracking	multi target
[3] Ding 2012	game theory	0	0	✔	✔	0	✔	2D	≈ 10	resolution	tracking	multi target
[45] Song 2008	game theory	0	0	✔	0	0	✔	2D	≈ 14	resolution	tracking	multi target
[9] Nikolaidis 2009	heuristic	✔	✔	✔	0	0	fix	2D	2 to 4	Region of interst	trajectory coverage	
[6] Zhao 2008	branch and bound	✔	✔	✔	0	0	fix	2D	≈ 10 (8-11)	tag detection	tag visbility	
[35] Yabuta 2008	linear programming relaxation	✔	✔	✔	0	0	fix	2D discrete square	≈ 20	region of interest		
[14] Hörster 2006	branch and bound	✔	✔	✔	0	0	✔ (discrete)	2D	≈ 10	cost reduction		
[43] Wang 2017	multistage grid subdivi-sion	✔	✔	✔	✔	0	✔	2D	9 to 15	region of interest	big area	

2.3/ SOLUTION BASED ON EVOLUTIONARY METHOD

The solutions approached, formulate the problem of camera positioning for a maximum coverage (as in the section 2.2) from various points of views. Until now the formulations and solutions the more common have not been approach yet. In numerous works the cameras positioning for maximum coverage is assimilated to a problem of optimization. Due to the complexity of the problem (as that was introduced in the AGP section 2.1.3) the linear optimization or based on heuristic may not be appropriate. The other solution, is to apply some stochastic method with an appropriate meta-heuristic. Among the various possibilities the Evolutionary Algorithm (EA) has been used at numerous times. The following sections, are focused on the different algorithms based on the EA. The EA family are commonly used in the literature to optimize a set of cameras for maximum coverage depending than different secondary objectives.

2.3.1/ AMONG THE EA ALGORITHMS

In this subsection different algorithms to solve the problem of camera positioning is discussed. The algorithms presented try to optimize the coverage depending then different specific constraints with different formulation adapted to the constraints.

The work of Zhao et al. [4] present few solutions to optimize the positions of a cameras set. Among the solution proposed a greedy with a local optimization, an integer programming with a solver, a sampling algorithms and the Simulated Annealing (SA) from the EA family has been compared. Also the SA is used to customize the sampling mechanism in order to propose a new customized solution. In Zhao et al. [4], the problem is formalized as a BIP (binary integer programming). The goal is to find among a restricted numbers of possible positions and orientations the best pose for a set of cameras. The best pose for the set of cameras has to cover the area, despite of the obstacles and the regions of interest.
Among the solutions proposed, the greedy algorithm has a good approximate solution, as long as the number of cameras stay relatively small. The problem of the greedy algorithm and the integer programming solver is to have an important risk to be stocked in the local optimum (local minima). This risk increases proportionally then the size of the environment. Otherwise the sampling technique with the SA is more adapted to the problem with times constraint and offer a well solution.
Despite that the solution proposed have some limitations. The major limitation is due to the experiment proposed. In fact, the experiment is made in a very small room with only few poses possible for each camera. Consequently, the solution proposed can appear better than a real situation and must be widely worst in a bigger area (in terms of time computation and optimisation of the poses). The more interesting aspect in this work [4] is the use of the SA form the EA family which allows to have an efficient coverage much faster than the integer programming solver for a similar result.

In the recent work of Akbarzadeh et al. [42] the cameras positioning for outside area, have been designed as an optimization problem with a few algorithms tested. Among the solution tested the SA and GA (Genetic Algorithm) both form the EA Family. The work of Akbarzadeh et al. [42] are interesting in many points.
The problem formulation is one of the interesting points. The goal is to cover an outside area with multi coverage (exactly a $k-$coverage discussed later in the section 4.1.1.4), the

relief of the terrain and obstacles with may generate potential occlusion. Due to several relief of the considered maps, the cameras are always pose at the same distance then the floor depending then the floor altitude. Like that the altitude of a camera is automatically deduced form his position and the relief associate. Moreover, depending then the position and orientation of the cameras the relief can be a source of occlusion for the cameras and has to be taken in account. All these elements combined make the problem formulation relatively complete and realistic.
Among the solution studied in [42], the non-linear have been tested with a quasi-Newton optimization method (called BFGS see in [42] section C), the SA and the CMA-ES form the EA family with some mechanism close then a Genetic algorithm (see in [42] section D) have been tested too. The result of the comparison gives a net advantage of the algorithms from the EA family in term of coverage rate with a fix number of cameras but also in term of time computation. The SA gives a close coverage rate (some time slightly better) than 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 than SA. The advantage of the CMA-ES appears more important in the bigger area. The quasi-newton and other algorithms tested are far away in terms of time and coverage, compared to the EA solutions. The conclusion of this work [42] is is the efficiency of the EA algorithm in a vast and realistic environments.

In Chrysostomou and Gasteratos [15] present an optimization system for maximum coverage with several constraints for 2 closely related problems. In the first time, the goal is to cover an inside area inspired by the AGP with a minimum of cameras. The minimum is fixed depending on a coverage threshold required rate. In the second time, the goal is to maximize the coverage for a fix number of cameras. This 2 problems are relatively close and just few elements has to be adapted to pass from on to another problem.
The coverage of the area is related to the camera visibility and few constraints has been proposed to control it, as the visibility, viewing angle, field of view, resolution, viewing distance, and occlusions. The solution proposed to optimize the positions, orientations and some of the camera parameters (as the focal lengths) is to apply an algorithm form the EA family called Bee Colony. The Bee Colony is inspired by the Ant colony algorithm and associate to the bee exploration in the nature. The Bee colony is in this paper [15] the more efficient of the two problems formulation (minimum of cameras and maximum of coverage with a fix number of cameras) after a brief comparison with a GA and a broach and bound algorithms.
The experiment proposed is relatively limited. Only one room has been tested with only one test per algorithm, which is not relevant for stochastic algorithm (due to the importance of randomness). Despite that the result proposed are encouraging.

The works proposed in this section has the common point to apply different algorithms form the EA family to optimise the problem maximize the coverage for cameras positioning. The algorithms form the EA family appear well adapted to the problem. To go further in the EA some specific branch as the GA has to be studied.

2.3.2/ Solution used Genetic Algorithm or close related

Among the solutions applied from the EA to optimise the coverage area, the GA or the algorithms closely related has been studied, as in the following section with some examples. In the solutions proposed as in [15, 20, 42] the GA have been applied as a

comparator. The GA and solution strongly inspired by the GA are not only used as a comparator, but it can appear more efficient as in [46, 38, 13, 25].

In Van et al. [46] the problem of cameras positioning for video surveillance to control a building is studied. A building with several floors is examined and must be covered by a set of cameras. To do that the environment has to be considered in the three dimensions (3D) with the ceiling occlusion. The solution proposed being 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 the problems. For the crossover a swap is done between two cameras form different sets.About the mutation, a Gaussian is used to mutate some parameters of the cameras in a set. The GA is used with an elitist selections. All these parameters of the GA are essential to have an appropriate optimization adapted to the problems and to understand the performance of it (more detail about the GA and these parameters see the next Section 3.3).
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 this case, the GA is well appropriate and works efficiently.

In Jiang et al. [13] the solution for the coverage problem in a wide area with the regions of interest and obstacles have been optimized with using a standard GA. In [13] the GA has been tested for a wide area with numerous cameras orient. The goal is to find the best orientations to maximize the coverage for a set of cameras randomly placed in the space. The experiment has been done in a vast area $400X300m^2$ with 60 cameras. All the cameras has the same depth and field of view. The number of cameras are not enough to control the full area and some choice must be done between the different regions of interest. The GA proposed in this article offer the best result around 47%, with a minimum of overlapping.
The solution proposed with the GA is interesting and work well to optimize the orientation of the cameras in a relatively big and complex area. The main element missing in the solution proposed is an experiment with a number of cameras supposedly enough to fully cover the area.

In Wang et al. [25] a variant of the classical GA has been used. The Multi-Agent Genetic Algorithm (MAGA is from [47])) have been developed by the fusion of 2 algorithms, the GA and the multi agents systems. The principal advantage of this algorithm is this supposed efficiency on the optimization for a huge number of dimensions (Zhong et al. estimate the efficiency range around 20 to 10 000 dimensions). In this case, the number of dimensions means the number of parameters to optimize for a given problem.
MAGA is used in [25] to optimize the area coverage. The area to cover is a 3D space area and most of the volumetric space have to be covered. Unlike the other articles the cameras have to be positioned in the 3 dimensions (in x, y and z) and the orientation too. The constraints of volumetric space associate to the optimization of the position, orientation and cameras property, increase greatly the size of the search space (also the number of dimension to optimize). Despite this interesting solution the experiment proposed are not enough consistent to properly judge it. Despite that the MAGA appear promising in term of potential increasing search space and answer quality.

In Topcuoglu et al. [38] the solution proposed is a Hybrid Evolutionary Algorithms (HEA). The HEA is based on the GA with different modifications, notably on the operators. As example, the crossover is redesigned in order to have two different parts of it. One of the

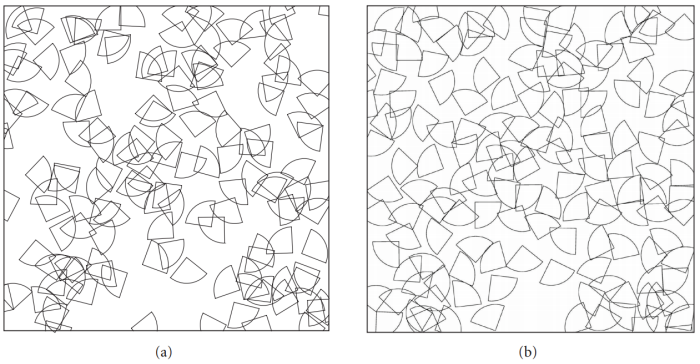


Figure 2.11: Illustration of the area coverage by a set of cameras randomly scattered (a). The cameras orientation has been optimized to maximize the area coverage (b). Experiment made in Xu et al. [5]

parts is a local crossover and the other is a more classical crossover. The HEA proposed in this article is relatively close than another EA algorithms called the Mimetic algorithms. The solution applied in [38] is presented with different experimentation dedicate to maximize the visibility and minimize the cost of the sensors network. The experiments propose to compare a simple random selection to the HEA. The result of this experiment shows a real efficiency in terms of minimizing the number of useful sensors and the total utility (the utility is related then the multi objectives.)

In view of these articles, the GA appears appropriate for the optimization of numerous cameras in a vast area as show in [13]. The GA optimization can be a good starting point to develop a new method by tuning and customizing the numerous parameters as in [38, 25] or [46]. Also despite the example presented the GA is relatively under-exploited in view of this customized ability and efficiency.

2.3.3/ 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 algorithms the most commonly used and not discussed yet, is the Particle Swarm Optimization (PSO) as in [5, 48, 20, 49, 29, 50, 40]. In this following section, the use of PSO and similar algorithms is discussed for the problem of coverage optimization.

In Xu et al. [5] the PSO is used to optimize the orientation of several cameras (around 150). The cameras have been randomly posed on the area. Each camera have the same properties and cannot change its position, but may adjust its orientation to any directions (as a PTZ cameras). The area is designed in 2D and the looking direction is described with only one rotation in pan (around the z axis). Finally the PSO is used to optimize the looking directions in order to have the best coverage as possible. The PSO optimize the area to reach a coverage around 65% after 1000 iteration and 20 particles for each iteration. The gain between the initial random dispersion and the optimized looking direction (with PSO) is around 12 points of percentage. The intimal coverage in the experiment was around 53% after the random dispersion (see the result in Figure 2.11.

In this case, the PSO allow with a relatively quick optimization a much better solution.

In Fu.x et al. [50] the same problem as [5] is discussed. The optimization of the orientation for an important set of cameras (100 cameras) using PSO. The difference is the number of parameters to optimize. In [50] the looking direction is not only optimized on pan but also with the tilt. The PSO is compared with other algorithms as the SA. Finally the PSO outperform the other solution experimented in this article despite an important number of parameters to optimize (100 cameras with 2 parameters by camera).

In Zhou et al. in [48] 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 positions. The coverage of the target is optimize for a fix number of cameras. The experiment made in [48] show the ability of PSO in term of efficiency and speed compares than a hierarchical method. The proposed hierarchical approached is a greedy constructive heuristic (see more in [48] section III). The experiment is made in a basic square room described by a grid of 81 possible targets location with all have the same importance and have to be covered.
The result of the experiment did it in [48], shown the slightly advantage of the hierarchical method. In fact, the hierarchical method give a better solution, but require a much more time computation. Otherwise the PSO propose an efficient solution close then the hierarchical method. The solution proposed by the PSO is also time efficient, between 2 too 6 time faster than the hierarchical method. The PSO combines result efficiency and fast time computation. The PSO appears more adapted to a realistic solution due to fast and efficient answer. Conversely the experiment proposed can be considered as limited due to the poor number of possible targets location in a simple room and the low level of choice for the cameras poses. An experiment in a bigger environment will probably shown an even more important gap between the PSO and the Hierarchical method.

As in Zhou et al. [48], Reddy et al. [20] 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 [20] is also taking into account the visibility parameters as depth of field and light intensity in a room inspired by the AGP. All these constraints affects greatly the coverage resulst. Finally the experiment did it shown the efficiency of PSO for a small environment with few camera (around 7 cameras) for a multi objectives problem. The advantage in [20] is the addition of more objectives and constraints than in [48]. Moreover the area has a better discretization and allow to confirm the efficiency of PSO for the coverage with multi objectives in a small room.

In Fu.y [29] the PSO is use to optimise the problem of maximum coverage as the other. The camera pose (with the orientation) need to covered the totality of a small square room. The orientation of the camera has to be optimized in pan and tilt. In addition, all the cameras have an identical focal lens. It is a relatively basic objective with few common constraints. The interesting part of this article [29] 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 based on a combinatorial problems. Despite the atypical formulation the solution proposed is greedy in term of cameras for cover a small space without obstacles.

In the recent work of Maji et al.[49] the finality is different but the problem can be considering close then the problem of cameras positioning. The objective is to position several transistors in a rectangular Printed Circuit Board (PCB). The transistors must have a rectangular shape with different sizes and ratios. Despite the apparent simplicity the several

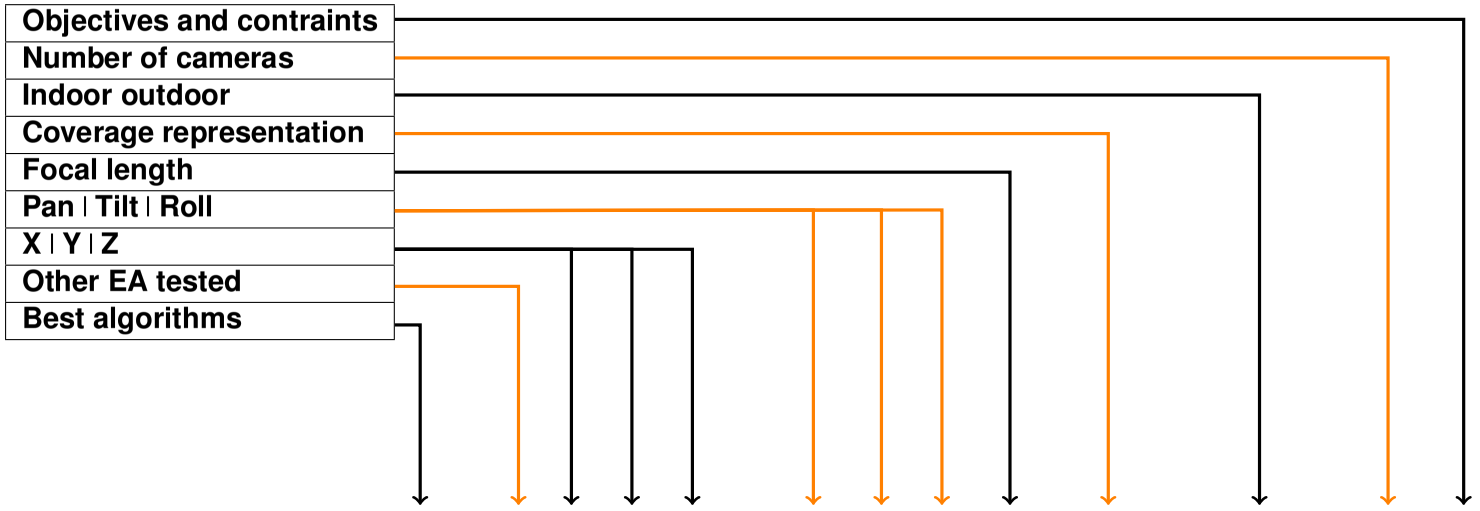
rectangular shape for the chips. Some potential additional constraints, as the relation dependence between chips, or the strict non-overlapping of the transistors, makes the problem of positioning transistor becomes more complex. The solution is to optimize the positions and orientations of the chips on the board using EA. 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 helps to pass over the several local minima. The more interesting aspect is to use a PSO in a similar problem then cameras positioning which was commonly answered with a linear convex optimisation as in [51] and based on the work on the floor planning defined Boyd and Vandenberghe in their book Convex optimization [52](8.8 page 438).

The solutions proposed is based on PSO are promising. The PSO or closely related have numerous advantages as showed 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 an 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. Among the few parameters to set-up manly the number of particles and the global inertia have an important impact (see more about PSO latter in the Section 5.2.1). On another side the PSO is not really efficient to optimize a lot of dimension at same time (compare then other EA). The PSO can quickly be limited as the increasing size and complexity of the area. Due to this limit the PSO has been customized as in [49] and [50].
The popularity of the PSO is mostly due to the association of the quick implementation, fast computation and efficient enough optimisation for various problem as coverage optimization.

2.3.4/ SUM-UP

In order to summarise the previous section a Table 2.3.4 is proposed with contain the interesting articles. The first column is dedicate to the references, the authors name and years. The second is dedicate to the best algorithms. In these articles most of the time several solutions has been tested. The second column show the best algorithm among the algorithms tested. The third column notify if among the other algorithms tested some of them which are from the EA family. The 4^{th}, 5^{th} and 6^{th} column precise if the algorithm proposed optimize respectively on x, y and z coordinated the position of the cameras. In some case the position on x and y are not optimize but randomly dispersed. The 7^{th}, 8^{th} and 9^{th} column precise if the algorithm proposed optimize respectively on pan, tilt and roll rotation of the cameras (if no optimization the orientation is fixed). The 10^{th} column is dedicate to precise if the focal length is optimized or fixed. The column for coverage representation precise the area to cover is designed in 2D or 3D (11^{th} column). In some case, the 2D representation of the area is enriched (noted with a "+" or "+relief"). The 12^{th} column show if the experiments proposed are indoor or outdoor (in, out). The 13^{th} column is dedicated to show the maximum number of cameras tested in the experiments. The last columns are dedicated to list the main secondary objective and constraints.

Table 2.2: Sum-up of the solution based on evolutionary method for maximize the coverage.



[38] Topcuoglu 2009	HEA GA	✔	✔	x	o	✔	o	o	fix	2D	out	10 to 200	max visibility	min cost
[20]Reddy 2012	PSO	o	✔	✔	o	✔	o	o	fix	2D	in	≈ 10	tracking	light intensity
[48]Zhou 2011	PSO	o	✔	✔	o	✔	o	o	fix	2D	in	1 to 20	tracking	resolution
[15]Chrysostomou 2012	Bee Colony	✔	✔	✔	o	✔	✔	o	(2)	2D	in	≈ 10	cost reduction	resolution
[46]Van 2009	GA	o	✔	✔	o	✔	o	o	fix	2D+	in	≈ 32	min overlap	multi level
[5]Xu 2011	PSO	o	random dispersion			✔	o	o	✔	2D	out	50 to 600	region of interest	
[42]Akbarzadeh 2013	CMA-ES	✔	✔	✔	relief	✔	✔	o	fix	2D +relief	out	10 to 110	relief	region of interest
[49]Maji 2015	CR-PSO	✔	✔	✔	o	0	o	✔	✔	2Drectangle		≈ 15	Rectangular proportion	dependency
[4]Zhao 2013	SA*	✔	✔	✔	o	✔	o	o	fix	2D	in	6 to 30	tracking	
[25]Wang 2009	MA-GA*	o	✔	✔	✔	✔	✔	o		3D	in	<30	Visible FoV	cuboid obstacle
[13]Jiang 2010	GA	o	random dispersion			✔	o	o	fix	2D	in	≈ 60	region of interest	
[29]Fu.y 2014	PSO	o	✔	✔	0	✔ (4)	✔	o	fix	2D	in	15 to 25	resolution	focus
[50]Fu.x 2010	PSO	o	✔	✔	✔	✔	✔	o	fix	2D	out	<150		

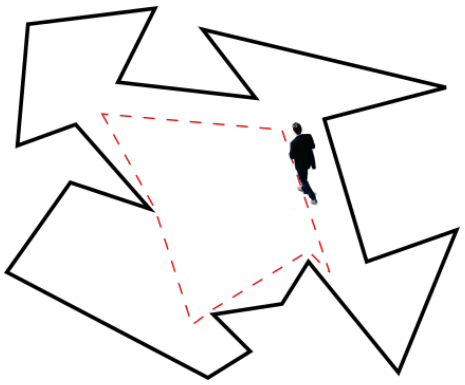
2.4/ COVERAGE PATH PLANNING

The coverage path planning as objective to design the more efficient path to cover an area. the path as to cover all or at least most of the area using the shorter path. The solutions put forward in the literature until now is to keep on eyes a vast area composed by numerous obstacles was to pose numerous cameras or robotics cameras (as PTZ cameras and smart camera). The solutions proposed until now are interesting only to monitor a vast area in continue with several cameras. The disadvantage of positioning a set of cameras appears quickly with the cost. The expensive cost is due to the several cameras and the communication network required to can centralize the collected images. All the images must be collected in a real time. This costly system is not always useful. In fact, in some application the area needs to be controlled periodically. The periodic control of the area does not require an installation for the set of fixed cameras. Until now the installation was composed by a set of immovable cameras unlike the solution presented in section see Section 2.3 and Section 2.2). The periodical control requirement can be illustrate with some example : The cartographies with need just one fly over an area roughly bounded as in [53, 54], forest fire detection which require a periodic fly over specific and vast region as in [55], the hoovering robots with need to cover all the room with some time some on-line computation [23, 56, 57, 58] and the agriculture. About the agriculture application, an UAV need to fly over the field few times the year in order to control by photography the hydration, the maturity and any thing else of the field as in [54, 59, 60, 61, 62, 63]. For all these applications the area must be covered but does not need a coverage of all the area instantly and continuously. The solution commonly proposed is to use only one sensor mounted on a mobile robot. The Mobile robot need to be adapted to the task and the environment, as flying [64], driving [12, 65], swimming [53]. The mobile robot with the sensor moves in the space to cover all the area. In this case, the objective is to determine the best path for the mobile robot to cover the area. The best path is dependent then the constraint of each problem. However, the common point, is to have the shorter path to all the area or at least most of it. The following section is focused on finding the best Coverage Path Planing (CPP) for a sensor mounted on a mobile robot. The sensor can be varied, but the great majority of the case it is a camera perspective. To estimate the best CPP different algorithms and methodologies has been applied in the literature to solve or at least optimize the CPP problems.

The following sections is focussed on the different algorithms and methodologies applied to solve or at least optimize the coverage path planning problem. In a first time, the watchman route problem is introduced to highlight the origin of the CPP problem and the relation with the cameras positioning for optimal coverage. In a second time, the more popular solutions are discussed.

2.4.1/ AGP TO WATCHMAN ROUTE PROBLEM

Before to explore the solutions proposed for the CPP is essential to understand the role of the Watchman Route Problem (WRP). The WRP is closely related than the CPP and the WRP has impacted the research for the CPP. The following sections are focused on the definition of the WRP, the solution proposed in order to solve it, and finally a fast discussion of the limit of the WRP is proposed.



(a)

Figure 2.12: Watchman route problems answer illustration.

2.4.1.1/ DEFINITION OF THE WATCHMEN ROUTE PROBLEM

The Watchman Route Problem is introduced for the first time by Chin and Ntafos in 1987 [66]. The problem of the WMP can be summarized in one sentence :

"How to calculate a shortest route contained inside a polygon such that any points inside this polygon is visible from at least one point of the route?".

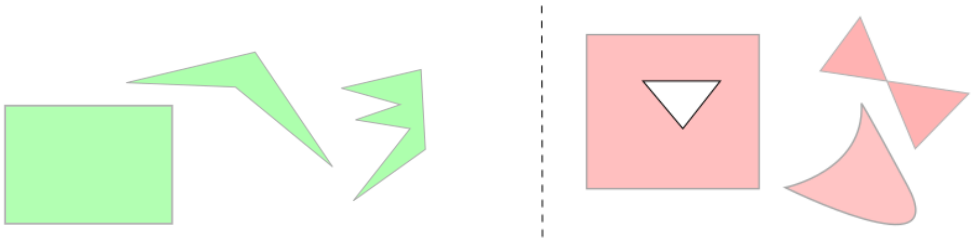
The guard has to cover an area represented by a polygon (see the illustration of the problems in Figure 2.12). The guards is considered as perfect with no restriction in the field of view (the guard can see at $360°$) and no restriction in the depth field (the guard can see form on extremity of the room to another. The guard can see the opposed wall excepted if an obstacle obstruct the view). The guard ability are directly inspired by the AGP (see 2.1.3) The shape of the polygon is primordial and affect greatly the possible answer and the complexity to solve the WRP. The WRP problem is by many aspects closely related than the AGP. The AGP (see Section 2.1.3) is commonly considered as the root of the WRP. In fact, the WRP is not only focussed on standing position, but on finding an optimal path. The path has to be optimized to cover all the points which compose the polygon and the path have to be shorter as possible.

The next section introduces the possible method and algorithm usable in order to solve or atleast optimize a solution for a WRP.

2.4.1.2/ SOLUTION

The WRP problem can be solve in some condition. The solution to solve it are applicable only if the polygon is simple. A polygon is considered as simple, when the boundary of it are composed of continue straight lines that do not intersect between them. To complete this definition, it is important to precise the simple polygon does not have a hole (see Figure 2.13).

To solve the WRP few algorithms were developed. Step by step the algorithms proposed in the literature allow a faster solution. The first interesting algorithm for the WRP with the simple polygon is the early work of Tan, X [67]. In Tan et al.[67] propose an algorithm for a fast computation time. The solution proposed work in the simple polygon composed by n vertices in a polynomial time $O(n^5)$.



(a) Example of simple polygons. (b) Example of non simple polygons.

Figure 2.13: Few example to illustrate the simple polygon.

Others algorithms proposed to go a bit further. Dror et al. [68] assume to have a better time complexity. In fact, in [68] the solution proposed for the WRP in a simple polygon is working in $O(n^3 logn)$. The deterministic solution applied is also usable in closely relate problem of the WRT as the zoo-keeper problem and safari problem. This two algorithms [67, 68] are usable only in the case of simple polygons to delivered the optimal solution which is the shortest path for a full room coverage.

To have a more general solutions for WRP, the proposition is to have an efficient algorithm working also with complex polygons. A solution is to look for an efficient approximation. An efficient approximation means a path short enough. Due to the computation method the optimality of the path can not be certified as the best. Several efficient approximations has been proposed as in [69] and [31].

In Packer [31] the algorithm proposed is based on splitting the problem into two sub-problems. The first sub-problem is to find a set of points with can be good enough to cover all the area despite a restricted visibility range. These points are called waypoints. Once the set of waypoints to cover all the polygon are found, the second sub-problem is to create a path passing by all these points. This second sub-problem is similar than a classic Travelling Salesman Problem (TSP). The TSP try to answer the questions asks by a travelling salesman "**What is the shortest path passing by each city only one time and return to the starting city?**". In the TSP, the cities are the nodes on the interconnected map and the roads are the connexions between them. The TSP is a well known as NP-hard and NP-complete in some conditions as described in [70].
Finally the solution used for the TSP can be applied for the second sub-problem of WRT. The TSP and the algorithms proposed to optimize it are discussed more in detail later in the Paragrapher 2.4.2.1.

In faigl [69] a similar method than the one proposed by Packer [31] has been proposed to approximate the shorter path as possible. The problem is also split into two sub-problems with the first is to optimize the position of the waypoints and the second is to schedule the position in order to create a path planning (directly inspired by the TSP). Furthermore, the solution proposed by Faigl in [69] is applicable to a watchman with a restricted visibility range. Equivalent then a $360°$ field of view with a restricted depth of field. This restriction affects greatly the waypoints positioning. Thanks to this constraint more waypoints has to be place to cover an area. Once the set of waypoints to cover all the polygon is found, the second sub-problem is to create a path passing by all these waypoints. The algorithms proposed to do it, are also inspired by the solution given for the optimize the TSP.

2.4.1.3/ LIMIT AND CONSEQUENCES

The method proposed to solve the WRP are really interesting and give a good solution in some specific conditions. These conditions are also the restriction about the methods proposed. In fact the optimal solution, that mean the absolutely shortest path which cover all the area (polygon) are usable only if the polygon respect some rules.The polygon must be simple and take in count the guard ability (no viewing constraints).
When the polygon begin to be more complex the optimal solution cannot be reached. The method applied to solve the WRP for complex area are interesting. Especially by the splitting the problem into sub-problems. Despite this interesting aspect the solution proposed are most of the time limited due to the area representation (must be a polygon composed by vertex). This representation associate to the geometric methodologies to find the waypoints give an crucial importance to the number of vertex in relation then the number of cameras. More the number of vertex is important more the first sub-problem risk to be difficult and long to solve. Consequently this method is not the more appropriate for the vast and complex outside area which would require numerous vertex to describe it.
The biggest limit of the WRP is the ability of the watchman. In fact in the original problem, the watchman is considered has perfect visibility. The solution proposed for answer to the WRP are not usable with the add of new visibility constraint. In Faigl [69] the WRP begin to be extended by add a constraint on the visibility. In this condition the problem is slightly muted to self-organizing map in the way to became a coverage path planning. But despite the add of a restricted depth of field the solution based on geometric heuristic do not allow to add more constraints.

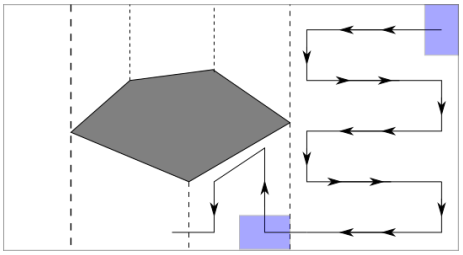
The AGP then WRP has greatly impacted the vision and the problem design of cameras coverage and coverage path planning. Among the solutions discussed in the precedent section numerous articles are based and make references to the AGP and WRP. Our work is also inspired by its and will propose to extend it.

2.4.2/ CPP SOLUTIONS

To optimize the CPP problems many solutions were proposed. The different algorithms and methodologies proposed are discussed in the following sections. The methodologies proposed are split in several branches. The more important branches to optimize the CPP is the use of sweep associate to a cellular decompositions.

2.4.2.1/ CELLULAR DECOMPOSITION AND SWEEP

To solve the CPP problem the solution the most common is to use the cellular decomposition. The cellular decomposition is by some aspect inspired by the methodologies presented from the WRP. To remember one of the interesting solution for the WRP is to split the problem in two sub-problems and optimize its independently. The first sub-problem is to find the best waypoints and the second sub-problem is to find the shorter path passing by all the waypoints (as a TSP). The Cellular decomposition also split the problem of CPP in two sub-problems. The first sub-problem is focus on the decomposition of the complex area in several cells. Each cell has to be a simple polygon (basically a rectangle or latter any quadrilateral polygon). Inside each cell a sweeping has be applied in order to cover all the area (see Figure 2.14). The second sub-problem is to find the



(a)

Figure 2.14: Illustration of simple cell decomposition with sweep.

path which can connect each simple polygon. The problem has to take in account, the sweep start and end to find the global optimized shorter path. Where the global optimized shorter path take in account the sweep trajectory of each simple polygon and the path between the simple polygons.

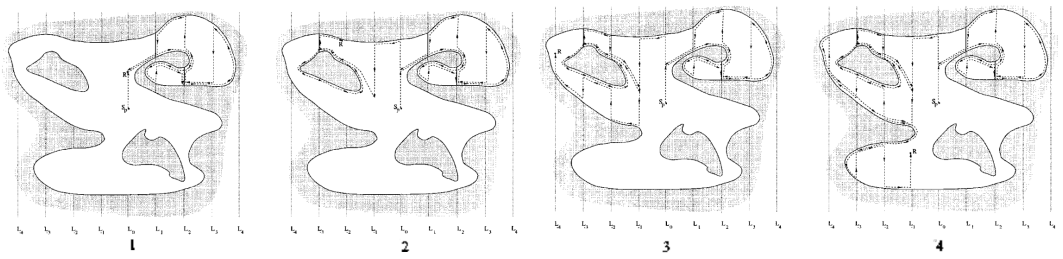
Finally the cellular decomposition is made by the decomposition of the complex area, choose the appropriate sweep and find the shorter path to connect the cells.

Decomposition

The decomposition in cells, has to objective to split a complex area in several sub-area. The sub-area has to be simpler in term of shape in order to can applies a sweep. Since the 90s numerous algorithms has been developed for the cellular decomposition. Among the algorithms for cellular decomposition, 3 types of decomposition has been made. As it is explained in the survey of Choset [71] and Galceran and al [53].

- Approximate: An approximate decomposition is based on a discretization of the area. The free space of the area is representative by a set of cases (gird). Each case of the grid has to be cover by the mobile robot. A case is considered completely cover if the robot is on this position. That mean the frequency of the grid is defined by the covered area of the mobile robot. The approximate decomposition by cases is fast to describe the area but is greatly limited due to the low sampling frequency of the gird and the limited trajectory possible.

- Semi approximate: The semi approximate decomposition is by part based on the discretization of the space. The idea is to create a set of large cells. The width of the cells is fix and the height is relative then the area boundary (see in [71]). The semi approximate cell decomposition allow to have cells with two parallel sides with a fix size (at the right and left) and the two other sides adapted to the boundary (at the up and down. See the illustration of a semi approximation in Figure 2.15). The width of the cells are chosen depending then half of the focal length of the camera mounded on the mobile robot. The area is covered by following the boundary of the cells with sweep to pass cells after cells. The advantage of this semi approximate decomposition is the facility to describe a vast area and the low computation required. Additionally, this decomposition can be made on-line by the robot and do not require a hight level of knowledge of the area. The disadvantage is the non optimized path due to the cells decomposition. The simplicity of this decomposition



(a)

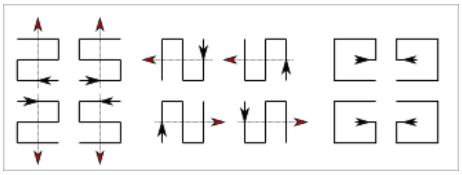
Figure 2.15: Illustration of a semi approximate decomposition issue of the survey of Choset [71]. The 4 figures illustrate the trajectory of the robot in the time.

can gender case, where the robot have to path many time by the same place. The semi approximate cellular decomposition do not allow to have a well optimised path planning.

- Exact cellular decomposition: The exact cellular decomposition describe the area by create juxtaposed geometrical region. The size of the regions (or cells) are not dependent then the robot ability despite the previous decomposition. The large size of the cells allow the mobile robot to do back and forth motion to cover all the cells (also called sweep as in Figure 2.16). The cells have to be ordered to have an efficient and short path passing by all the cells. The global path distance has to be reduced by an appropriate path passing by each cells The exact cellular decomposition became popular and numerous algorithms has been proposed. The algorithms proposed can be; a faster decomposition or with a more appropriate shape depending then the basic rectangle cells. Among the numerous exact cellular decomposition the trapezoidal decomposition for this simplicity and historic importance, the Boustrophedon decomposition and Morse-based Cellular Decomposition for their importance in the construction of other exact cellular decomposition has to be cited. These algorithms and other has been clearly summarized in the survey of Carreras and Galceran [53]. The exact cellular decomposition has continue to be studied since the survey of Carreras and Galceran [53] and upgraded, as in [72] for propose a decomposition for concave or multiple polygons.

Sweeps

The back and forth or sweep is an essential element of the CPP by cellular decomposition. The sweep has to cover the entire cells. The cells are splitting the area in relatively simple polygons (as see in the paragraph 2.4.2.1 and the Figure 2.14). The sweep has to be adapted then the shape of the cells and also must start and finish in a appropriate position for passing to the next cells. For that the starting point and the ending point of the sweep are crucial for the global path planning. In Torres et al. [72] the sweep is function then a direction and can go clockwise or counter-clockwise to have a start and end in the appropriate position for the transition. This allow to have 8 different applicable sweeps at each cell depending then the more appropriate start and finish according to the computed path. The sweep are adapted depending then :



(a)

Figure 2.16: Different sweeps and spirals

- The turn sides (clockwise or counter-clockwise)
- The directions (horizontal or vertical)
- The finishing positions (start and stop form the same side or opposite side)

The sweep can be also alternatively switch for a spiral as proposed in Jimenez et al. [73]. The proposed sweep and spiral in [73] are adaptable depending then the context. The sweep can have two directions and for the spiral two turn sides (clockwise or counter-clockwise). The different sweeps and spirals are illustrated in the Figure 2.16.

To have an adapted sweep the footprint has to be defined depending then the camera ability and the objectives. In [72] the size of the sweep is dependent then the area cover by a camera and a sufficient amount of overlaps for the 3D reconstruction (see also[22]). In Li et al. [7] the foot print of the camera with a small pan is took in account. Due to the pan the camera projection is not a rectangle but a trapezoid. Consequently the sweep size is adapted by taking the larger side of the trapezoid for the sweep dimension. Also for the grid decomposition the sweep size is directly related the coverage ability of the mobile robots (as example in [57, 74]).

One extra element to take in account for the sweep can be in some case the external element as the wind. In [57**?**]the external condition are taking in account in the cost function and influence the sweep.

To summarize the sweeps has to be adapted then the area and the relation between cells. To choose the appropriate sweeps the following element has to be taking in account:

- The size of sweep have to be defined depending then the ability of the camera (foot print size).
- The direction (horizontal or vertical).
- The starting and finishing position (start and stop form the same side or opposite side)

To can have the more appropriate sweep is primordial to know by advance the cells scheduling when is needed. The cells scheduling are a crucial part of the path planning.

Path planning

The aims of the path planning is to find the short path passing by all the cells. Find the best path planning passing by all the waypoints or in this case all the cells is a complex

problem which can be formulated as a TSP. The path planning became even more tricky in the case of exact cellular decomposition. Or for any decomposition which requires a sweeping inside each cells. In fact in this case, the goal is to have the shorter path planning with taking in account each sweep and transition between cells. In the precedent Paragraph 2.4.2.1 the different sweeps and spiral has been detailed. The start and the end of the sweep are the crucial elements to take in account during the path planning computation. Obviously, to find the best scheduling between each cell the algorithm form the TSP are commonly used.

When the area is decomposed in cells the goal is to schedule the priority of passage from one cell to another in order to create an efficient path passing by all the cells. The scheduling of the cells can be optimize by using the paradigm of TSP with those algorithms to have the best path planning. The TSP is well known problem and is deeply studied form long time ago. First is essential to remember the TSP is an NP-complete and NP-hard problem as proofed in Karp in [70].

To optimize the TSP numerous algorithms was tested and some of them has been specially applied to schedule the cells to have the shorter path. In An et al. [75] two algorithms was tested before to develop a third. The algorithms used are based on branch and bound. The algorithms developed in [75] is called Novel Previous-Next Waypoints Coverage Constraint (PNWCC). The algorithms presented in [75] propose at same time a schedule of each cells with also a smooth trajectory without sharp edges usable for non holonomic driving robots.

In the survey of Carreras et Galceran [53] the solutions proposed to solve the TSP is to computes an exhaustive walk trough the adjacency graph. This solution is workable only for compute small adjacency graph. The GA is popular to optimize the TSP and is commonly use to evaluate the influence of this parameter as in [76] [77]. About the CPP problem the GA is also used to optimize the TSP part as in Jimenez et al. [73]. Where the GA is applied to find an optimized schedule after an exact cellular decomposition of a complex polygon.
The GA is announced well appropriate to have an optimized solution for the TSP. In some situation a TSP is not realistic and some external constraint can be added as the wind effect, the turbulences, or holonomy constraint as in [78, 79, 53]). The add of external constraints may become more complex the scheduling of the cells.

2.4.2.2/ OTHER SOLUTION

Among the algorithms developed for the CPP some interesting methods have to be studied. Some algorithms has been discussed in Choset [71] as approximate solution (see Section 2.4.2.1).

The approximate cellular decomposition was briefly approached in the precedent Section 2.4.2.1. The approximate cellular decomposition was considered as not interesting due to the low sampling frequency of the grid. Conceptuality the works of [56, 58, 57] going further by using a higher sampling frequency of the area to cover. The solution based on regular grid discretization which may be compared to approximated cellular decomposition with a high sampling frequency as in [23]. Due to the relatively fine grid decomposition which engender an increased size of the case number, the algorithms to optimize the path passing by all the cases of the grid has to be adapted . Consensually new navigation strategy has been developed.

In Luo et al. [56] the goal is to have a complete coverage by visiting all the cases of a grid. To visit all the cases of the grid a neural-neighborhood analysis and neural dynamic programming approaches are adapted to have an efficient CPP for the robots. In Simon et Luo [58] they propose an similar method with a neural network solution for dynamic and non planar area.

The work of Lee et al. [57] propose another solution based on smooth spiral path. The idea is to propose to follow the boundary of the room to fully cover and use a smoothed spiral to cover the area. The solution proposed is compared to other method. The main advantage of these method is the on-line ability and the smoothed trajectory(see also [74]).

Before to conclude about some of the algorithms usable to optimise the coverage path planning the more elementary has to be discussed. In fact, among the algorithms proposed until here the full random was not discussed. In Liu et al. [80] a really basic methods is developed to cover an area with an mobile robot (hover robot) for on-line path planning in a dynamic environment. The algorithm proposed is based on random direction. The basic idea is to move forward until the detection of an obstacle (ultrasonic or bumper). When an obstacle is detected the robot turn randomly. This solution is really basic and is based on the idea then if the room is not too big or not specially complex with enough time the mobile robot will cover all. Obviously this solution is not optimized and not acceptable in many cases.

3

GENETIC ALGORITHMS

Contents

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 it is important to go back to the origin. The following section is focussed on the fundamental theories of the natural selection and its 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.

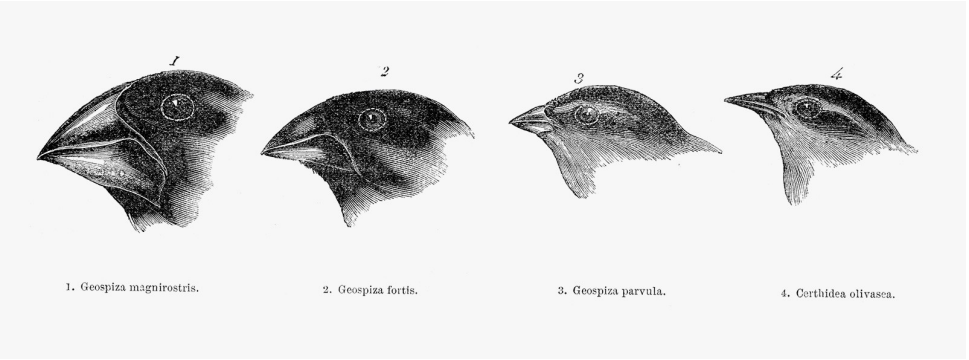


Figure 3.1: Finches from Galapagos archipelago extract od The Origin of species by C.Darwin .

The origin of species details is called the theory of evolution. 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 birds, 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 birds 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). Therefore 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 contains all the useful informations growth, 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 biology, every living element is composed by cellular. Inside each cellular the DNA code is stored. 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).

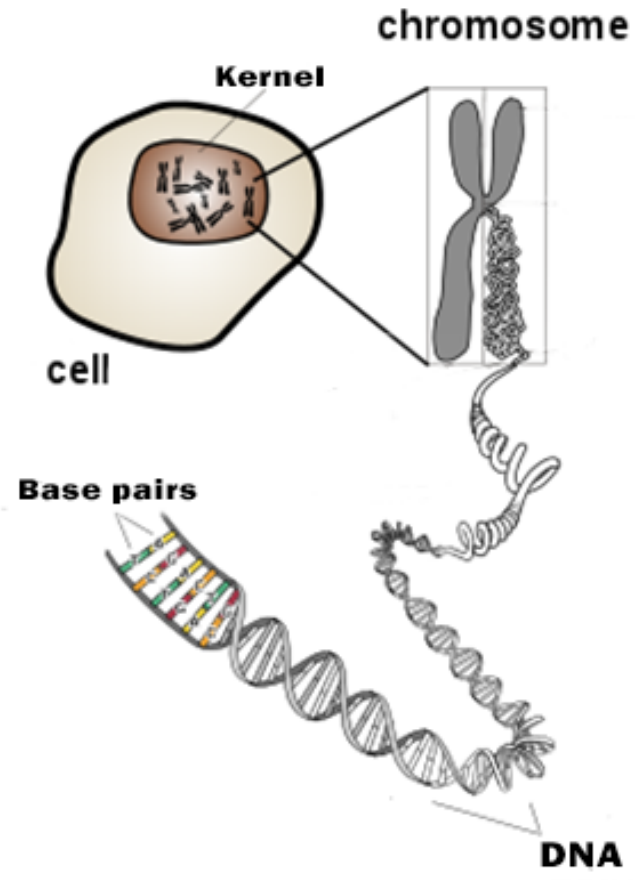


Figure 3.2: Biologic representation, cell to chromosome until DNA.

As introduced earlier, the evolution is possible by a natural selection. The selection is done by survival and reproduction ability of the better individual. Better mean 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 makes the geopiza with the bigger beaks in better position to reproduce and the DNA of the individuals with biggest beak is transmitted to 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 from 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.

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

Mutation. Contrary to the crossover which allows only the ability preservation from one generation to another, the mutation introduces some “anomalies”. The anomalies 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 genes are affected.
The mutation changes the little piece of DNA code to introduce variety in the genetic code by the small “anomalies”.
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 as an 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 become 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 given 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 algorithms and they include many meta-heuristic used in the field of optimization and artificial intelligence.
The evolutionary algorithm are inspired by the biologic mechanism to design meta-heuristics. The origin of the inspiration can be varied as the genetic, insect work, animal 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 has most of this attribute :

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: List of some basic EA.

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

These definition are not the strict characteristics for all the EA. They are rather the most common element of the major part of the vast EA family.

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. [81] in 1951. More commonly, the beginning of the EA are on the late 50s with the works of Bremermann [82], Friedberg [83], Box [84]. They propose different algorithms based on the evolving solutions to optimize any given problem.
During almost the three next decades, 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.
Despite of this difficulty, the fundamental works of Holland [85] 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. About the application of the EA in the engineering field (examples in [86]) and the

multiplication of the conferences in EA, allowed the democratization of this family of algorithms.
The EA took 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 was one of the premises of machine learning and classification.

- **The evolution strategies**, especially the work of Rechenberg ([87]), which propose a strategies based on deterministic selection and random mutation. The goal was to solve difficult experimental problems with discrete or continuous search space. Rechenberg applies in particular the evolutionary strategies for aerodynamic profile design.

- **The genetic algorithm**, is probably the most polyvalent and tunable from the EA methods. The GA propose an adaptive processes to optimize a solution. The detailed GA precisely in the Section 3.3.

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) [88, 89, 90]. The MOEA are the logic extension of the EA to answer of problem more and more realistic which require to manage different objectives and constraints potentially contradictory. The MOEA is discussed in the Section 3.4

3.2.2/ General formulation

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 given problem, in order to obtain the best solution as possible. The possible solution must be formalised as a input vector $\vec{x}$. Where each element of the vector ($\vec{x}$) is one input or a dimension (x_i) to optimize the given problem.

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

Where n the number of dimension to optimize (or number of inputs), and Ω is the search space of the problem. The search space represents all the possible solution of the problem. Some time, depending then the problem the elements in the vector $\vec{x}$ are ordered. In this case the value of x_i is important as this position i in the vector $\vec{x}$. Consequently the search space is even bigger.
Bigger is the search space, more the solution risk to be long to found, in term of time computation.
The goal of the EA family is to optimize an solution ($\vec{x}$), in order to have the best solution

possible for the problem, where the best solution is defined by a cost function ($f(\vec{x})$). The objective is to maximize the value of the cost function regarding a input solution ($\vec{x}$).

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

The EA maximize work towards to the global optimum solution $\vec{X}$.

$$\max f(\vec{x}) \leq f(\vec{X}) \tag{3.3}$$

Where $\vec{X}$ is the global optimum solution.
Optimize a solution $\vec{x}$ is not always enough to solve efficiently the problem. The proposed solution has 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.

$$E = \{e_1, ...e_m\} \tag{3.4}$$

Where e_j is the j^th constraint of the set E. To evaluate if a solution $\vec{x}$ respect the constraint e_j the function $e_j(\vec{x})$ can be use. Thus, that mean the solution $\vec{x}$ must minimize the set of constraint E and maximise the value of the cost function $f()$.

$$\max f(\vec{x}) - \min(\sum_{j=1}^{m} e_j(\vec{x})) = \max F((\vec{x})) \tag{3.5}$$

F is the cost function, which 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 algorithm applied. 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 necessarily 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 infinitely. To avoid the infinite optimisation the end has to be fixed depending then some stopping criterion.

3.2.3/ STOPPING CRITERIA

To determine a stoping criteria the proposed solutions in the following part are more especially adapted to the algorithms like GA, PSO, ant colony, mimetic and other EA working with a population to optimize.

The EA are mostly efficient in the problems with many local minima. Nevertheless, due to numbers and size of the local minima it can be difficult to ensure if a solution is the global optimum. To know if the global optimal is reached, the method applied must be sure that no other solution can be 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 if it is applicable as for convex problem. The convex problem has only one global optimal and no local minima.
The EA optimize a solution to be better as possible (not the optimum). Because the uncertainty of the EA optimization it is impossible to call the solution founded as global optimum.

Consequently, how to determine when is time to stop the optimisation. Because at some point, it is useless to continue the optimization, the best solution has been founded or the optimization is lock in some deep local optimum.To determine the stopping criterion three possible way are communally used:

- **fixed time criterion**: is to stop the algorithm after a fixed numbers of iterations or time limit. The limit is measured in term of time computing, also the numbers of iteration must be fixed by the user. The main interest of this method is to control the computation time of the problem which require a solution in a determined time (as for a real time). The risk of this method is to stop before finding an efficient solution.

- **Update criterion**: is to stop the algorithm (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 stopping criteria is to can stop before the convergence with a solution close or identical then the one reached at the convergence. The inconvenient is to stop too early. In fact, if the update criteria is not properly chosen the optimization may stop at the beginning.
  To use this stopping criterion a correct number of iterations has to be selected. It 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).

- **Convergence criterion**: is to stop the algorithm by waiting the convergence point. The convergence is reached when the actual population is composed by a set of identical solutions. 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 best one. In this case, the population has been converging to an optimized solution.

The stopping criteria presented can be a combination of the three methods to have an efficient and flexible solution as presented in the following example :
The mixed stopping criteria has to combine a fix time criteria and the update criteria. The advantage of the fixed 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 to reach the complete convergence and before to reach the time criteria limit.

The combination of these stopping criterion, always provide 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 called Genetic Algorithm (GA) and was introduced for the first time by Holland in 1962 [85].
The genetic algorithm (GA) is one of the fundamental algorithms of the EA. The GA became popular at the late 80s and early 90s, particularly with Goldberg works [91].Since the 90s, many details were redefined and explored in the knowledge of genetics to have a huge set-up and operator available for the GA.

The following section will try to list the more interesting aspects of the GA mechanisms.

3.3.1/ CHROMOSOMES

As in the Biologic field the chromosomes contain properties (with the genes 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, taking into account of different aspects of it.

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 parts 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: consist in direct mapping of the gene corresponding to the elements of the solution. Using the direct coding may simplifies the output of the optimization by returning it back to an element directly proper to use.

- 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 gene 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 [92, 93] where it is used to solve a complex scheduling problem with many constraints.

To define the chromosome the direct or indirect coding is not the only element to take into account. The coding structure is also important to the chromosomes design. The coding structure has to encode the solutions in the chromosome like presented in [93] and [94]. Among the possible coding structure 4 main categories can be considered as basic coding structure, which are the combinatorial coding, binary, the real coding (also alphabet) and tree coding. Moreover the choices for the chromosomes must also take in account the wanted solution in term of number of dimension to represent and their boundary.

Binary Coding The binary coding offers to format the chromosomes as a bit string in which every gene 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. [97] and also used in other EA like PSO in [98]. 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 [95]. 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 gene in 2 pack of bit.

The combinatorial The combinatorial coding is commonly applied in some specific problems where the goal is to order all the element. In this case, the position of the gene in the chromosome is primordial as in [99]. It is characteristically used to the problem as TSP [77] (Travelling salesman problem). When the combinatorial coding is used for problem the aim is to optimize the order the element of the chromosome. With a combinatorial coding the each chromosome already has all gene of the answer. An example, the TSP the aim is to order various cities (in the problem of TSP every gene 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 combinatorial traditional coding is to can add and remove some gene, that obviously affect the size of the chromosome, for example when the goal is to find the shortest distance in the tree [100].

Real Coding Real coding or integer coding is considering every gene 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 [76], which the operators are adapted to look for close neighbours.

Tree coding 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 combinatorial representation. An example in [93], 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 [93] 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 [92]. The survey [92]

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 [101] and also in the chapter 5.3.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. [88]. In Zhou et al. [88] 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 [88] 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 [102].
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 [96], 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 [92]). 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 [103]).
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 [100] 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 [103, 102, 104, 105, 97, 106]. 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 [102], 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 [104].

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 [107], 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 [102]. 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 [106]. 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 [104, 105].

3.3.4/ 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 [94]). 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 [108, 103] to estimate the efficiency of convergence using this selection mode or in [90] 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 [109]. 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

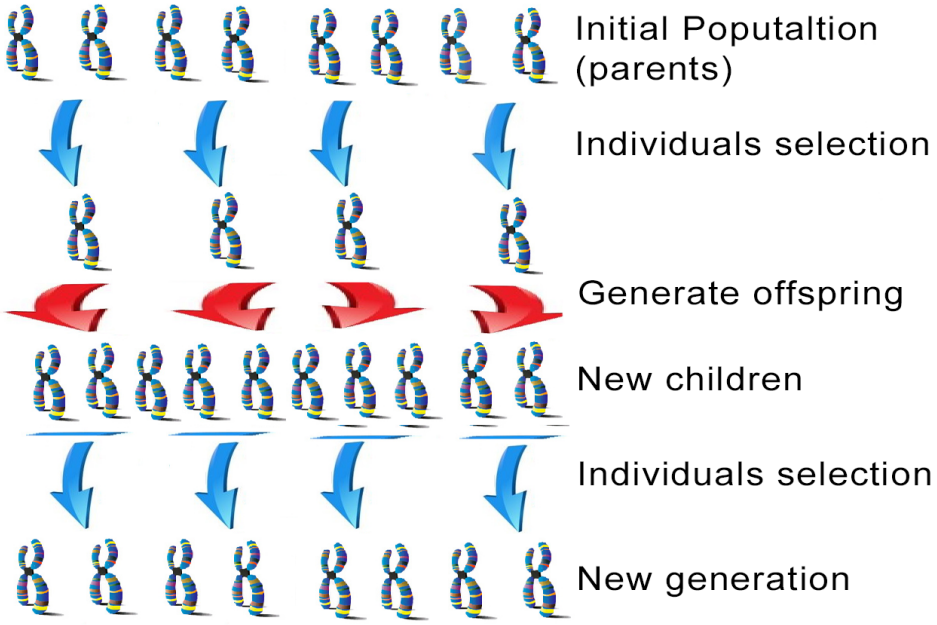


Figure 3.3: The 2 time for the selection mode with the GA mechanisme.

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 [103].
Obviously the size of the pool has a really big influence on the convergence as is studied in ,[103, 96]. 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 population 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 [100].

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 [100]. The mutation mechanism affect only rare gene in all the population. The gene mutated is randomly selected and the is will be based on random.

Customized operators This 2 principal 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 [76].
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 genes 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 [110] 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 gene 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 [95, 104, 107].

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 [77, 96] [100]. It is appearing one good static configuration to keep diversity [103] 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 [99, 107, 111]

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 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 [104]. 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 [95, 104, 107].

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.

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 [112] for the Simple Genetic Algorithm (SGA) are a good point to understand the progress before 1994.

In this article [112] 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 the customized GA to satisfy the multi objectives problems [88].
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 Sorting GA II (NSGA-II) [108]. These algorithms are using an elitist selection to optimize efficiently the problem without having to sort the different solutions depending to the objectives. Some other MOEA as QGA for Quantum-inspired GA [108, 113, 114], Non-dominated Sorting GA (NSGA) or BMPGA for Bi-objective Multi Populations GA, are examined on this survey[108].
The GA have been studied for different objectives and optimization problems. These surveys give a fast view of the GA formulation and specific customizations for GA application field, as the following example.

- In [115] are interested on the problem of clustering and use GA for have a non-supervised clustering. Also in [115] show the different implementations and customization of the GA, adapted to the problem of clustering.
- In [116], the GA 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 [94], the GA is applied into 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 decades the GA has been used for multi objectives problems, but its popularity has decreased in favour of algorithm which requires less configuration or other algorithm more oriented on learning with memorization.

4

PROBLEM MODELING

Contents

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 maybe 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 constraints as for example using a camera mounted on UAV.

The following sections are 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 points have to be clearly defined:

- The area himself. How to represent the area.
- The camera definition. How to estimate the cameras projection.
- The constraints added to the systems (constraints and secondary objectives).

This three 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 section is focused on the different grids design, based on the literature. The use of grid to describe the area to cover is common and several design has been invented and used depending then the constraints and objectives.

4.1.1.1/ SAMPLING FREQUENCY

The grid map is used to discretize the area to cover. The discretization of the area may 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.

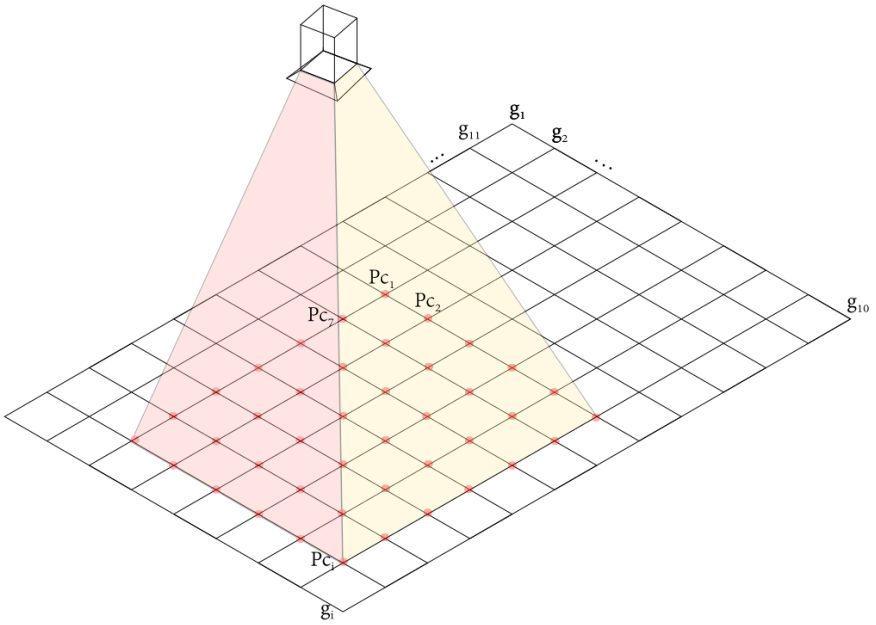


Figure 4.1: Camera projection onto a grid. The grid G is placed on the floor to discretize the area covered with numerous grid point g_i. The point cover by the camera Pc are noted in red.

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 [14] 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 test to estimate the coverage of each point of the grid, for a given 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 the 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, higher is the sampling frequency of the area more freedom has to be given to the pose estimation of each camera.

	Advantage	Disadvantage
High sampling frequency	Best estimation of the area to cover	Time consuming
	Give more precision on the cameras poses	
Low sampling frequency	Faster computation	Bad coverage estimation

Table 4.1: Sum-up of the low and high sampling frequancy adaventage or disadvantage.

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 [48]. In Zhou et al [48] a small value of m is chosen to have a real time solution for small area with just few cameras (up to 20). In the 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 points of the grid. This black hole can be too small to be detected. Due to the low density of points. 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.

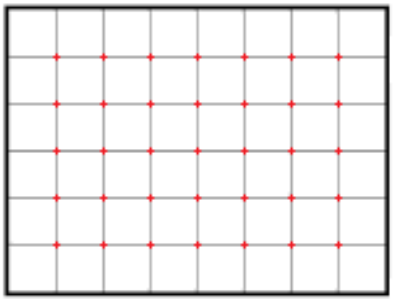
Low or high sampling frequency

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 sampling frequency in the solution has also consequences, 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 [6] is to have a progressive refinement by increasing the grid density.

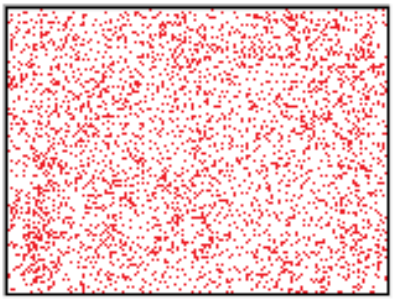
Zhao et al [6] despite a low density of points at the beginning, the number of points are increased slowly to have a better density and refined results. Also the increasing points frequency is applied to refine the solution and add more cameras at each step of the optimisation to avoid the black hole.

Cameras pose precision.

The frequency sampling is often linked to the pose precision of the camera as discussed previously. Therefore increase the number of points in the grid will also increase in proportion the number of possible positions 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.



(a) Grid with uniform and regular distribution.



(b) Random distribution for represent the map.

Figure 4.2: Map representation using random distribution or uniform grid.

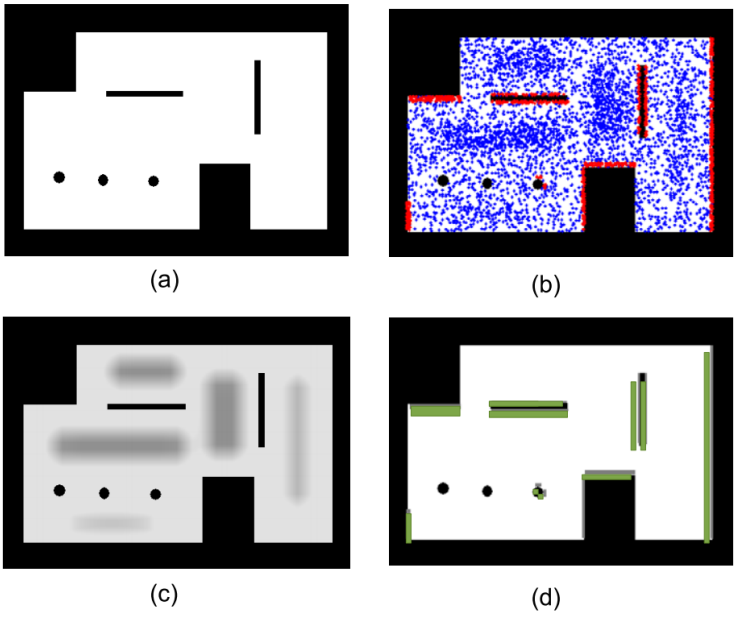


Figure 4.3: Example of map to cover with a randomized area sampling. Illustration form Horster et al [14]
(a) Area to cover.
(b) Area with the random points of the grid to cover
(c) Importance weighting of the area (zone of the interest.
(b) allowed position for cameras position.

4.1.1.2/ DISTRIBUTION

The distribution is an important factor to manage for design an occupation grid. The distribution is how the points of the grid are placed on the area to cover. Different distributions 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 [46, 14] the random distribution is used to describe the area to cover.
In Hoster et al [14] the points of the grid are randomly distributed to describe all the area to cover. The advantage of this article [14] 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 [46] 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 zones in the area. Based on this observation hengel et al [46] decide to use the uniform grid notably because of its simplicity of implementation. The ”random distribution” is less popular but in [6] a hybrid distribution is proposed. The idea 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 the grid a random selection is used.

4.1.1.3/ SPATIAL MODELLING (3D OR 2D)

To design the occupation grid, one other element should be taken in consideration. The spatial modelling, has an important impact on the grid design. After deciding the useful density and the distribution of the grid, the position in the space of the points g_i which compose the grid has 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 [54, 41, 48, 35, 14, 6], but depending on the context the grid have interest do describe a 3D space. In Hegle et al [46], calculates the visibility, where it is relevant: for example, on the upper torso or head of the possible target rather than the floor. In this case a 2D grid is proposed, but inside the 3D space in order to characterize properly the volume by placing the grid at a specific height.

In [15] the grid is formalized in the full volumetric space by numerous “control points” 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 [98] the grid are formulated in the volumetric space by using a uniform 3D grid distribution. The formulation proposed shown the complexity even more important to use a 3 dimensional occupation grid. 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 difficulty of the optimization process. Nevertheless some solution was discussed [42, 46] 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 Van Den Hengel et al[46] 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 informations by adding a layer at each level of the building.

In Akbarzadeh et al[42] 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 (hill and valley).

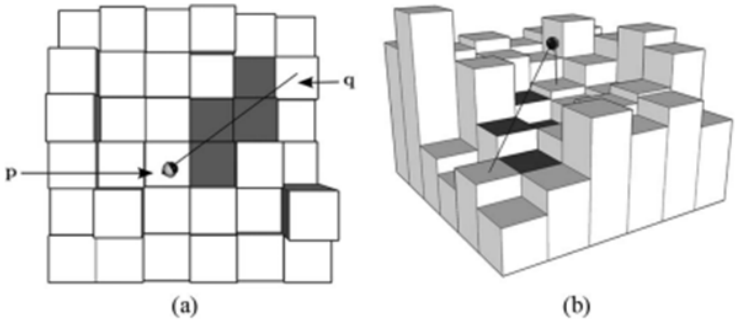


Figure 4.4: Relief grid use to discretize an area with taking in account the relief.

To estimate properly the area to cover a grid has been placed following the altitude of the relief as showed in Figure 4.4. The spatial modelling is traditionally a 2D grid at fix altitude more or less equal to the floor ground. The other model are commonly more greedy in term of point to describe the area. The 2D grid model is the easiest and can be adapted for numerous problems with allow to reduce the number of points and thus the computation.

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 zones.
- The priority zones.
- Non-interesting zones.

The multi coverage zones The aim of the multi coverage zones, is to have on a specific zone of the area controlled by numerous cameras. Multiple coverage may be called k-coverage as in [26]. 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 the grid 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, excepted for the multi coverage zones, where they have to be covered by k cameras as in [15].

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 ways.
One way to express the priority is with a grid uniformly distributed, where a weighting on the points of the grid which need to be covered in priority is applied, as in [42, 5]. This method was implemented in [42] to optimize the position of the camera on the road passing through the area to cover.
Another way to express the weighting of the priority zone for the randomly distributed points, is to increase the sampling frequency of the desired zones. Increase the sampling frequency of the priority zones is simplified due to the use of a random distribution as in Horster et al [14] and as Illustrate in the Figure 4.3b. 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 which have to be cover in priority.

Non-interesting zones. 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 [42]. For uniform grid distribution in the non-interesting zones are removed to the list Pc. This method is currently used like in [6, 35, 42, 14, 5] .

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

Finally, these methods use to design the zones of interest are not fully independent and can be associate with the same model as in [42, 14, 5]. 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 [41]. 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

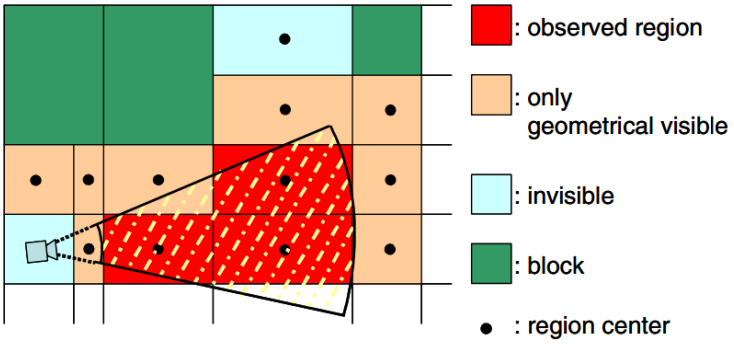


Figure 4.5: Observed regions from camera candidate. Grid representaiton form [35].

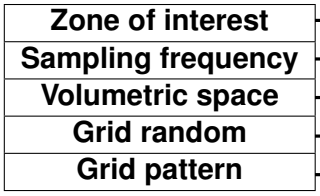
to the resolution wished but by the sensor range. Indeed the size between the points of the grid has been defined by the size of the minimum sensors range. The distance of the minimum sensor range are used as nodes for a topological relation.
Another atypical solution is to develop a grid composed by 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 [35]. This method is adapted to the area with few obstacles as is shown in the Figure 4.5.

Sum-up

The following paragraph surmise the different ways presented to design the map of the area. The area represented has to be covered by set of cameras. The grid map may be used to represent different objectives and constraints of the problems. The map representation take an crucial role in the area coverage. The design aborted previously are summarized here in this Table A.2. This table show the more important aspect of each grid representation in different paper.

4.1.2/ OUR APPROACH

Based on the designs studied 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 adopted 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.



[48] zhou2011	✔			For real-time	✔
[6] zhao2008	✔	✔		Incremental	✔
[15] chrysostomou2012	✔		3D grid		✔
[46] van2009	✔	✔	2D grid at each level		
[98] morsly2012	✔		Superposition of 2D grid	Adaptable	
[42] akbarzadeh2013	✔		2D grid on relief		✔
[41] chakrabarty2002	✔			Topologies sensor	
[54] valente2013	✔		overlap by shifting of z		
[35] yabuta2008	.			For zone segmentation	✔
[14] horster2006		✔			✔
[26] zhang2016					✔

Table 4.2: Sum-up of the grid map.

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: 4.6 with $k = 1$ and $p = 1$.
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 cameras 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 examples [38, 20, 48, 15, 6]. Other model of cameras or vision sensor can be used as for example omnidirectional with a $360°$ of field of view as [28, 41, 26]. 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 a box (or a 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 objects 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 (the back side of the inside 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. To estimate the projection just

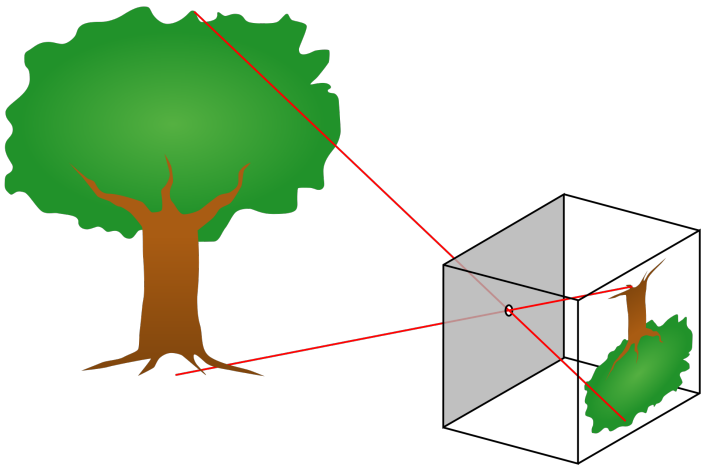


Figure 4.6: Pin hole camera model.

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

view element has to be knew:

- Three degrees of freedom for the camera position : (x, y, z);
- Three degrees of freedom for the camera orientation orientation: with the the rotation in pan, tilt, and roll angles: (α, β, γ)
- Optical parameters including: the focal length f of the lens, the sensor size $Sw \times Sh$, u_0 (the digital sensor or the film) 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:

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

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

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 [29, 43, 13]. In [29, 43, 13] 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 6 Degree of Freedom (DoF) in [29] and 5DoF in [43]). Both are used to estimate the 2D projection of the camera onto the floor.
In [29], 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 [43], 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 [13], 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 [13, 43, 42] include the object occlusion. To detected the part of the area occluded by an object, the solution commonly proposed (as is well explain in [43]) 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 by one object in the scene, the point g_i cannot be considered any-more as covered.

To go further other model and formulation inspired by the pin hole model has been proposed as [98, 42, 7, 50]. These models are inherited for the camera projection and adapted to fit their problems.
In [98, 50] 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 [42] 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 [7] 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 [98, 42, 7, 50, 6, 20, 29, 43, 13] is the computation of a camera projection on to a grid. The computations necessary to estimate, if a point of the area is covered 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 calls 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 than the computation of the coverage estimation, is primordial due to the numerous calls.

4.2.1.2/ COVERAGE ESTIMATION OPTIMIZATION

To 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 related to 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 a given altitude the area covered by one camera, based on these given parameters. Especially the focal length f, the altitude of the cameras A and the sensor size $Sw \times Sh$.
In this model the camera projection is always a rectangle as described in Figure 4.8. In fact the camera is placed on an UAV and have a direct look to the ground floor. the looking direction of the cameras is perpendicular then the floor, thus the rectangle projection. 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.
Based on the useful parameters form the camera $(f, Sw \times Sh, A)$ the computation to estimate the size of a camera projection (Wr,Hr) is :

$$\begin{aligned} Wr &= (\frac{1}{f}.Sw).A \\ Hr &= (\frac{1}{f}.Sh).A \end{aligned} \tag{4.11}$$

The model of camera projection is greatly simplified by the UAV assumption (fixed pan, tilt and focal length). That permits to consider the camera projection as a simple rectangle with a size directly related to the altitude. The altitude is assuming to be a coefficient of the Equation 4.11
All these simplifications helps the cost function to be fast and efficient. In fact in the Equation 4.11 the parameters focal length and the sensor size are fixed and do not need to be recomputed for each camera position only the altitude A has to be updated.

4.2.2/ PARAMETERS TO OPTIMIZE

By dint of the simplification presented previously the vector of parameters (see Equation 4.8) can be simplified. To summarize, the computation of one camera projection onto the floor, where the floor is represented by a grid of the area to cover, and the camera is in altitude with a looking direction orthogonal to the floor. Just few parameters are

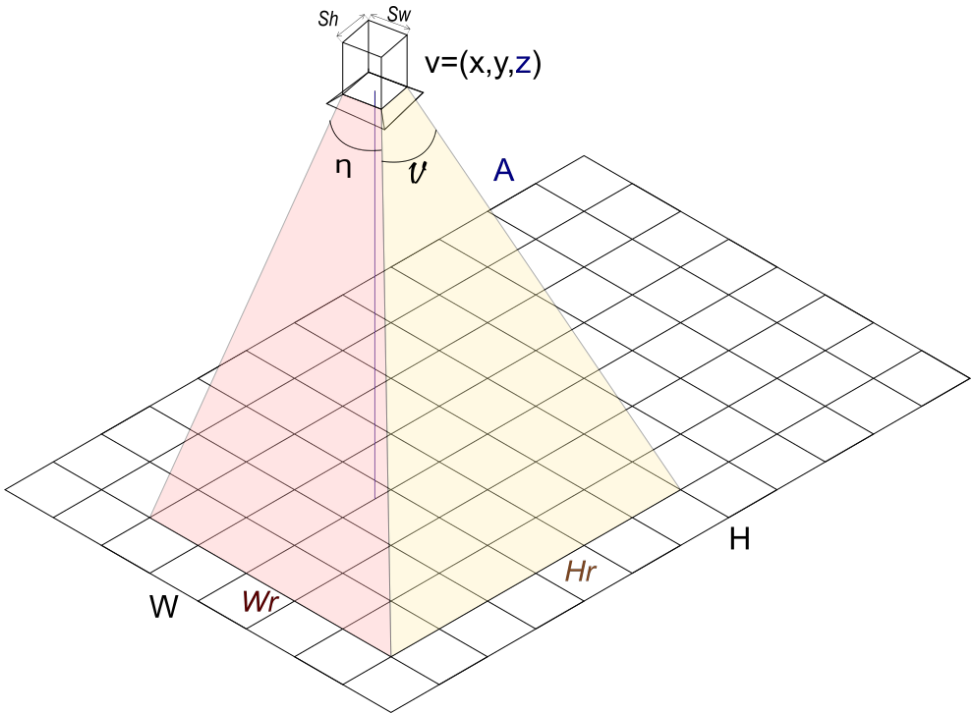


Figure 4.8: Camera projection onto a grid. The grid is placed on the floor to discretize the area covered.

necessary as showed in equation 4.11. Thanks to this design the Equation 4.8 can be reduced with keeping only the position of the camera and the roll as:

$$v = (x, y, z, \gamma) \tag{4.12}$$

Reducing the number of parameters, passing to the equation 4.8 to 4.12 are really usefully to optimize the problem. The reduction of the number of parameters 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 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 cameras. 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.

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:

$$V = \{v_i\} \ , \ \forall i = [1; n] \ , \ n \in \mathbb{N}^* \\ \text{and } v_i = (x_i, y_i, z_i, \gamma_i) \tag{4.13}$$

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, in 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 related 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. E represent the set of constraint (the set E is defined more in detail latter).

So that the V should respect the constraints of the set E (see Eq.4.14). 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 :

$$Vs = V \text{ , iff } E(V) = \begin{cases} 1, & \text{iff } E_i(V) = 1 \text{ , with } i = 1...Nc \\ 0, & \text{otherwise} \end{cases} \tag{4.14}$$

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 constraints have to be defined depending on the problem and the goal.

4.3/ COST FUNCTION

The cost function has to evaluate the quality of an given answer. To estimate the quality of an 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 constraints depending then their impact on the problem two types of constraints are introduced in the following sections before to discuss about.

4.3.1/ CONSTRAINT LIST

The constraints 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 constraints 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 [6] 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. Thanks to this constraint the optimization has to focus on the precise cameras positioning.

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 difficulty by reducing the possible search space (see section 4.4). It is also useful for other assumptions, as a cameras on the celling or for a submarine [53]. This constraint is an optional constraint and is not always used in the experiments presented in the following sections.

The altitude boundaries When the altitude is not fixed some limits must be chosen to avoid the extremely high and low altitudes. The highest altitudes 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 users under the UAV. In practice the altitude boundary is defined with :

$$\inf A \leq A \leq \sup A \tag{4.15}$$

Where $\sup A$ is the maximal altitude of the camera and $\inf A$ the minimum altitude. A is the altitude between the camera and the grid.

The map boundaries The map boundaries, 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:

$$\begin{aligned} 0 \leq x \leq W \\ 0 \leq y \leq H \end{aligned} \tag{4.16}$$

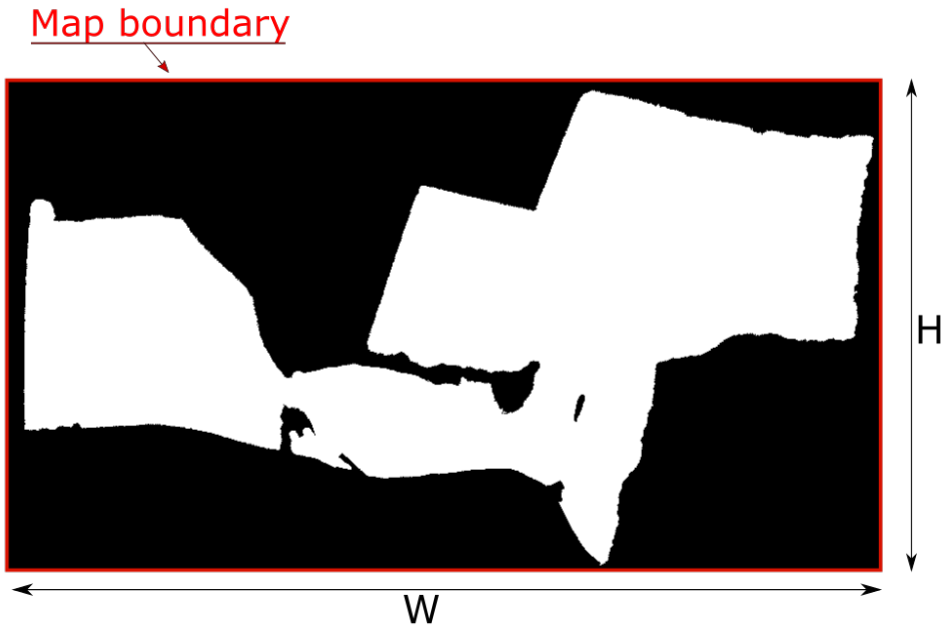


Figure 4.9: Map to cover with the map boundary in red (W and H size) in black the sub-part have no interest to be covered.

By associating the altitude boundaries (from Eq4.15), a cube boundaries limits the position of the camera in the 3 dimensional spaces.

$$\begin{aligned} 0 \leq x \leq W \\ 0 \leq y \leq H \\ \inf z \leq z \leq \sup z \end{aligned} \tag{4.17}$$

Non rectangle map with possible hole. Despite the rectangle boundaries of the area, the map to cover can be much more complex than a simple rectangle and may 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 point of the grid to cover. 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 camera 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).

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 (or objective) 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. The altitude of the set of cameras as to be minimized, the average of the altitude is used as index for estimate the resolution ($\min \frac{\sum_{i=1}^{n} z_i}{n}$). In order to manage this trade off, the average altitude of the cameras is included in the cost function and has to be minimized(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. These hard constraints is directly used during in the optimisation process to prohibit any solution to be out of these boundaries. These hard constraints has to be integrate in the optimization process in order to cannot generate a solution with does not respect it. Consequently the hard constraints may some times slow down the generation of the individuals due to the specific generation and test required.
For example, the 3D boundary as defined in Equation 4.17 is a hard constraint. Each cameras position must be inside the 3D boundary. Otherwise if one of the camera are out all the solution must be rejected.

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 $E = \epsilon + \varepsilon'$.

$$\max f(Vs) - \min \epsilon \ \forall Vs \subset \varepsilon' \tag{4.18}$$

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 complex shape of the map thanks to 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 constraints as the complex shape of the map by removing the points out of the area to cover.
The grid modification allow the cameras position to cover the area already covered and removed form the grid. The consequence of it are to reduce the coverage rate possibility. The optimization 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.

$$C(Vs) = \frac{\sum_{i=1}^{N} Pc_i}{m} \tag{4.19}$$

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 constraints. 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 Section 4.3.1). A criteria must be added in the cost function formula of the Equation 4.19. The average of the altitude z is used and have to be included in the cost function.

$$\bar{z} = \frac{\sum_{i=1}^{N} z_i}{N} \tag{4.20}$$

If the resolution is strongly related then the altitude the average of it $\bar{z}$ can be considered as a part of the soft constraint (ϵ) in the Equation 4.18 and the Equation 4.19 may be updated as :

$$C = \frac{\sum_{i=1}^{N} Pc_i}{m} - \frac{\sum_{i=1}^{N} z_i}{N} \tag{4.21}$$

The Equation 4.21 is used in the cost function to add the resolution constraint. Consequently, the optimization try to minimize the average altitude and maximize the coverage with no priority. Concretely by just applying this Equation 4.21 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 4.21. 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, is to link the weight of the resolution criteria with the coverage rate.

$$\sigma \times \sum_{i=1}^{N} Pc_i \times \frac{\sum_{i=1}^{N} z_i}{N} \tag{4.22}$$

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:

$$C(Vs) = \frac{\sum_{i=1}^{N} Pc_i - \delta \times \sum_{i=1}^{N} Pc_i \times \frac{\sum_{i=1}^{N} z_i}{N}}{m} \tag{4.23}$$

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, priorities, and constraints. The one presented here is the more equilibrated.
Despite that some of the work presented in the following sections are made with the basic cost function from the Equation 4.19. In this case, the element which compose v can be reduced as only $v = (x, y, \gamma)$ depending then the need of the experimentations.

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. Where G include the soft constraints as the room shape. The constraint of resolution is added by using the average altitude in the equation 4.23. 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 and search space

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 its. This number of position can be estimated for each camera as follows.
Each camera defined by the position on x, y, z and γ can be set anywhere in the search space of a cameras named Sp :

$$Sp = (W \times H \times (\max(z) - \min(z)) \times 2) \tag{4.24}$$

Where W and H are the size as width and height of the area to cover, $\max(z) - \min(z)$ is the range of possible altitude. Two is to define the roll γ, as the rectangle projection is horizontal or vertical (landscape or portrait).

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

$$\binom{n}{Sp} = \frac{Sp!}{n!(Sp-n)!} = |Vs| \tag{4.25}$$

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 of one camera (as Eq. 4.24) associate to the set of n cameras (as Eq.4.25) 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. 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. The complexity for the problem of camera positioning is at least NP-hard has the AGP. In fact if we considering the camera positioning as inherited from AGP with some extra constraints (as field of view and depth of field). Also the literature is unanimous about the NP-hard complexity as discussed in [28, 31, 51, 6, 117].

The size of the search space is even more important then the method applied. In fact due to the problem complexity and the numerous local minima the classic optimization algorithms is not appropriate. The stochastic algorithms are efficient in this case. Due to the wide search space that increase the difficulty to converge. The stochastic algorithms is widely based on the optimized random solutions and bigger is the search space, harder is to get the best solution.

5

WAYPOINTS POSITIONING EXPERIMENTATION

Contents

To remind, the main objective is to compute an efficient path to cover an area with complex shape using a camera mounted on an UAV. The solution proposed here, is to focus in a first time on optimizing the position of a cameras set, to fully cover a vast and complex area. When the set of optimized cameras pose is found the positions 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 the waypoints positions.

5.1/ WAYPOINT POSITIONING CONTEXT OF EXPERIMENTS

During the previous sections, the problem was discussed as an optimization problem. The formulation of the problem was presented and also the complexity of the problems was discussed in the Section 4.4. The precedent sections has introduce all the useful elements to solve the problems of camera positioning in a complex environment.
This chapter is dedicate to the different experimentations done to test the efficiency of the proposed methods. For the first experiment, different environment are proposed. The parameters of the environment has been adapted and carefully chosen.
The environment have been designed to have a various shape and size. The rooms shape are used to estimate the impact of the shape on the proposed solution. The shapes of the rooms are designed to estimate the exact number of cameras in order to have a ground trough (for the first set of experiments). Moreover the rooms propose few sizes. The sizes of the rooms may have two main impacts on the optimization. First, the increased size of a room, will increased in proportion the size of the search space. On the other hand a bigger room allow to place more waypoints. More the number of waypoints to place is important more the number of dimensions to optimize raise-up. The increasing number of dimensions is used to test the robustness of the algorithms.
The room finally used for the experimentations are proposed in the Figure 5.1.

In addition of the rooms parameters, the number of dimensions is also tested by added various possible altitudes for each waypoint. The waypoints can be in the simplest version placed only on x and y coordinate. In the second time z coordinate can be optimized inside a given range of altitudes. Two sets of scenarios has been tested one with a fix value of z and another set of scenarios with a range of z. This is to evaluate the impact of the altitude on the optimization process.

5.2/ OTHER ALGORITHM USED

In order to compare carefully our proposed method, other algorithms has to be introduce. The following sections show the two algorithms used for the comparison to the proposed GA. The environment introduced in 5.1 is used to make the comparisons. The particles swarm optimisation (PSO) and a random selections are described in the following sections. The PSO is used due to this importance in the literature for similar problems. The RS for this simplicity to evaluate the optimization, being a reference point.

5.2.1/ PARTICLES SWARM OPTIMISATION

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 [118]. 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 birds flocking, fish schooling and swarming theory. These animals are working in group (or swarm) 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 foods 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 [118] 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 from the swarm 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} :

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

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 factors to search away from the particles. These factor are also called learning factors and push to learn or discover (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 (more exactly in the number of iterations). Thus 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 particle. 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 it. In fact, the PSO needs at minima few elements to works 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 decades.

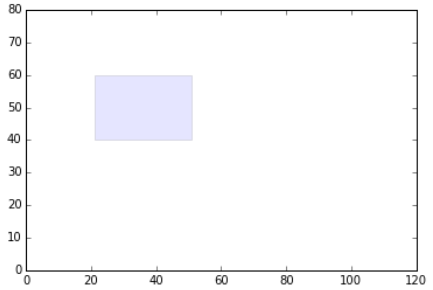
5.2.2/ RANDOM SELECTION

The Random Selections (RS) is a very basic algorithm. It serves as a reference point for the comparison of algorithms. The RS does not take a complex meta-heuristic and is perfect to compare the efficiency of the other algorithms.

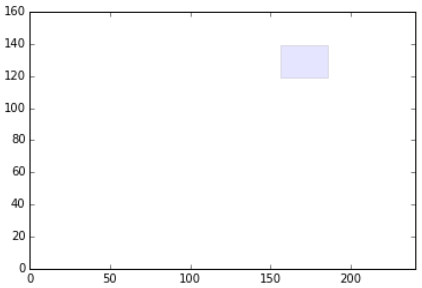
The random selections works by randomly generate numerous solutions. Among the solutions randomly generated the best solution is kept as the optimized global solution. The RS allows to look through the search space by randomly try many possible solutions. The search space exploration by the RS is only 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 calls, 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. To be efficient for the comparison several RS has to be made to can have an average optimized value of RS solutions.

5.3/ ALGORITHM COMPARISON

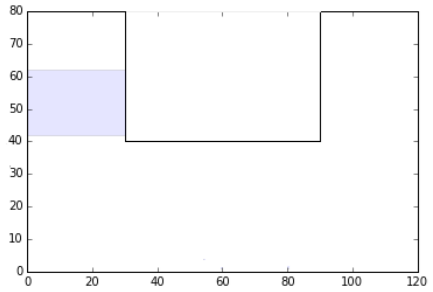
To solve the problem of cameras positioning (or waypoints positioning) the usable algorithms are varied as that was discussed in the Chapter 2 (see the sum-up Tables 2.1 and 2.3.4). Among the algorithms studied in the literature the EA family appear as the more suitable to have an appropriate answer 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 Table 2.3.4). The PSO gives a good and fast result in many case. In the EA 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 [20, 48, 5, 49, 29, 50]).
The conscientiously comparison of GA and PSO was proposed by Boeringer et al in



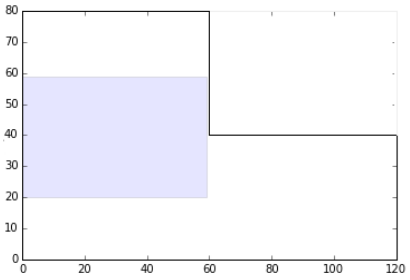
(a) Simple room.



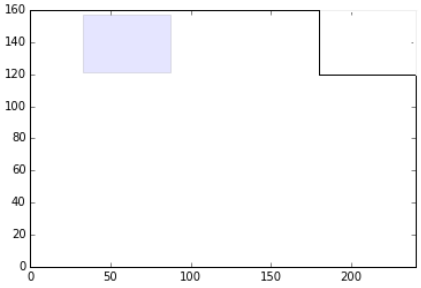
(b) Big room.



(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).

[119]. The conclusions of the comparison in [119] is relatively open and highlights the similarity of results between the two algorithms and the importance of evaluate them for each new problems. Consequently an set of experimentations has to be done to find the best algorithm for the problem of cameras position in a complex environments.

5.3.1/ DESIGN OF EXPERIMENT

To find the best solution to optimize the coverage of an area, 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. To compare and evaluate their performance, we tested them in different scenarios. Each algorithm was tested before to have an appropriate set-up of their parameters. The comparison is applied on different scenario as that was introduced in the section 5.1 and listed below.

- 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 constraints as the fix altitude and various 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.

z=1		**GA**		**PSO**		**RS**	
		GT	**NC**	**GT**	**NC**	**GT**	**NC**
Room	**120x80**	16	20	16	20	16	20
	240x160	64	70	64	70	64	70
Room U	**120x80**	12	20	12	20	12	20
z=2		**GA**		**PSO**		**RS**	
		GT	**NC**	**GT**	**NC**	**GT**	**NC**
Room	**120x80**	4	10	4	10	4	10
	240x160	16	20	16	20	16	20
Room L	**120x80**	3	10	3	10	3	10
	240x160	15	20	15	20	15	20

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 waypoints required to fully cover a given area. The sizes and the shapes 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 until 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.

5.3.2/ Analysis of the results

After performing several experiments (see Table 5.1), it appears that the GA and PSO algorithms are close in performance in numerous cases. Among several experiments of the DoE some particularities appears despite the globally close results of GA and PSO. Also as expected the RS are always the worst solutions. In the following subsection just

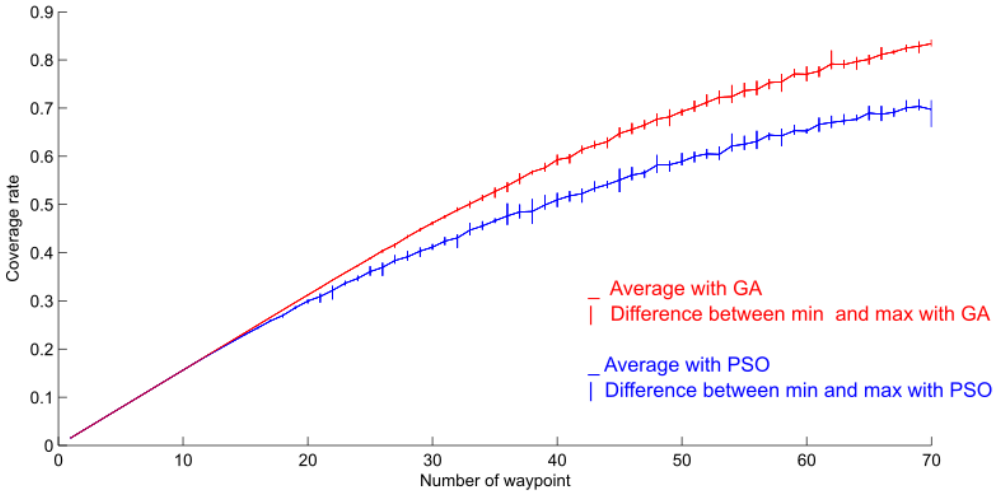


Figure 5.2: Comparison of eight solutions given by the GA, with eight solutions given by PSO algorithms with a fixed altitude (z equal to 1) in the big room 240×160. The ground truth for this room equals to 64.

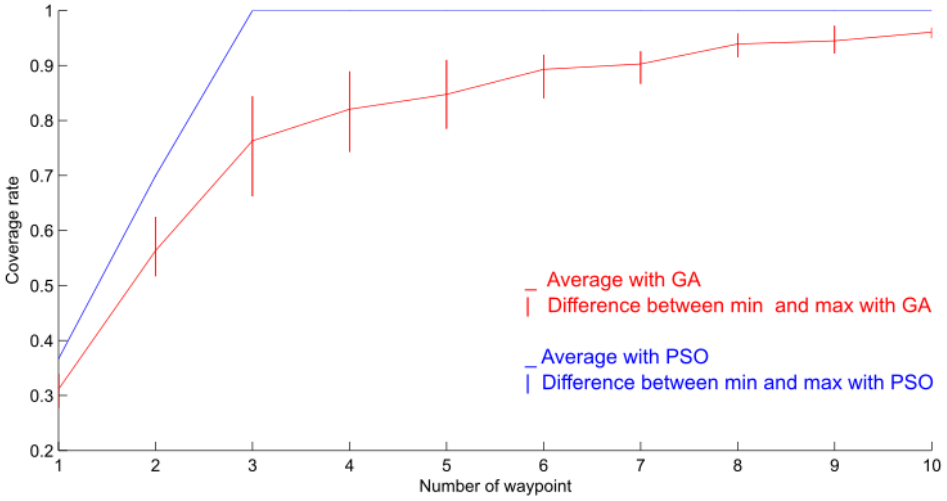


Figure 5.3: Comparison of eight solutions given by the GA, with eight solutions given by PSO algorithms with a Z between $[1/2; 2]$ in the room with L shape 120×80 and ground truth equal to 15.

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 these cases (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 relatively bad results obtained during the experiment in the small rooms. The variety of the GA negatively affects the accuracy of the solutions and may require a further optimization step to refine.

5.4/ HYBRID GA PSO

Thanks to the experiments done and presented in the previous sections (see Section 5.3) 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 dimensions to optimize. When PSO is efficient to refined faster the solution. The hybridization can be the key to optimize the cameras positions in all the condition. The aim of the hybridization is to exploit the better of both algorithms, trying to further refine the solutions.

5.4.1/ THE DIFFERENT HYBRIDIZATIONS

Different hybridizations of the GA and PSO can be made. Each hybridization of GA and PSO has advantages and disadvantages. This following section is focused on the main hybridisation of GA and PSO.
In Premalatha al et [120] propose three different solutions 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 iterations. 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 problem of cameras positioning in a vast and complex area. The experiments made until now (see Section 5.3.1) 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. In the other hand, the PSO ability 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 [105] 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 [105] 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 [105] 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 problem. As that was disused in Shi et al [105] numerous tests have to be done to find its.

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 slightly higher.

5.4.2/ EXPERIMENTATIONS

To compare the efficiency of the hybridized GAPSO to the GA, one experimentation is proposed. The experimentations followed the rule fixed during the comparisons as in Table 5.1. It appear 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 situations, 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 reducing the inertia. Before to reduce the inertia of the PSO few tests were 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).

5.4.3/ RESULTS AND COMMENTS

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.

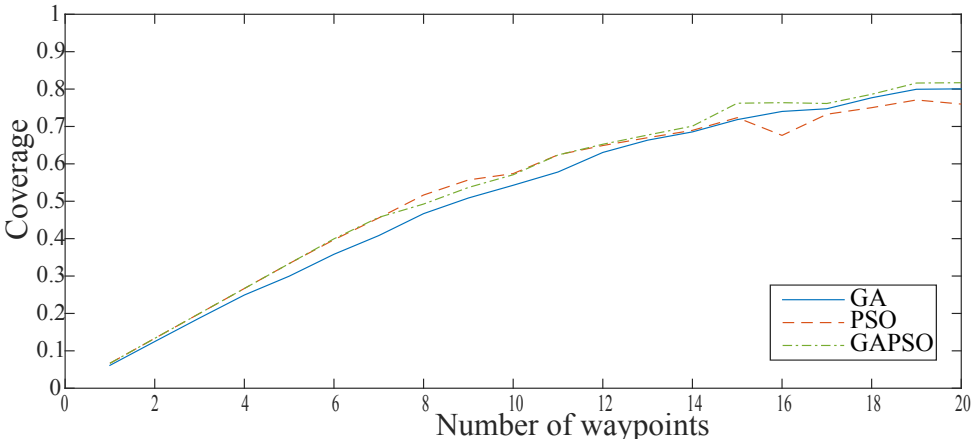


Figure 5.4: Comparison between GA PSO and the hybridization of GAPSO.

In the Figure 5.4 it is appearing the hybridization of GA, PSO increases slightly the percentage 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 these experiments for each 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 at almost any time the best solution and some time slightly better by combining the advantages of both algorithms. The GAPSO can reduce the limitations 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 advantages (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.

5.5/ GOING FURTHER, MORE EXPERIMENTS

The previous sections 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 areas relatively simple the next step is to increase the difficulty of the scene. The increased complexity is made by adding:

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

The following sections present the results obtained by increase step by step the difficulty. Each step presented chronologically in the following sections 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 steps has been presented.

5.5.1/ RECTANGLE OBSTACLES

Thanks to the results obtained (see Section 5.4) an set of experiments is made using a slightly bigger size than the big room (in Figure 5.1). The environment of these experiments try to be more realistic, consequently the room is designed with more obstacles. 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 5.1). 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.

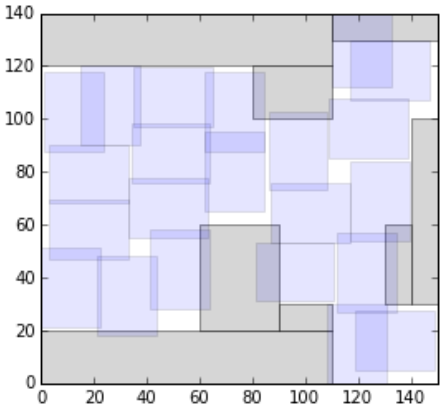
The simulated room is 15×14 m^2 which corresponds more or less to a large lecture hall. The areas in red (see Figure 5.5a) represent the zones which do not require coverage. Every camera can cover a 4×3 m^2, when z is equal to one. The z factor can be equal at $[0.5, 1, 1.5]$, and the cameras can turn 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 (No limit in the number of cost function calls).

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 Section 5.3.1). As aspect, despite the increasing number of obstacle and increased size of the search space. This confirm, again the ability of adaptation of the EA optimization and more exactly the single GA. These results encourage us to go further.

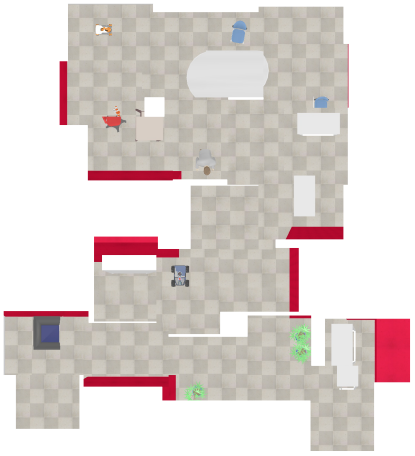
5.5.2/ RECTANGLE OBSTACLES WITH HOLES

The previous experiments shown the efficiency of the single GA despite numerous obstacle and slightly bigger room. The following experiments try to push a bit more the optimization process using a single GA. The increased complexity of map was made by adding much more obstacles 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

(a) Room in V-rep.



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



(c) Reconstructed room based on the waypoints positions.

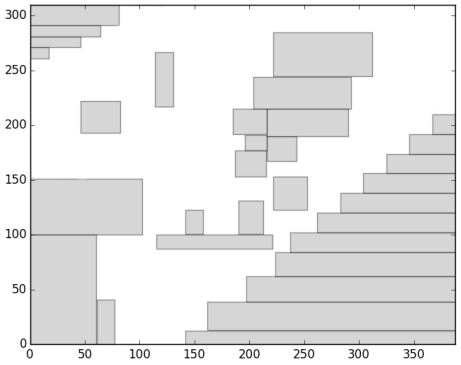
Parameters	Value
z	[0.5;1;1.5]
γ	portrait or landscape
Size of the area	150x140 px
Maximum size of camera projection	30x22 px
Number of waypoints	22

Figure 5.5: Indoor area coverage using V-rep to simulate a realistic environment.

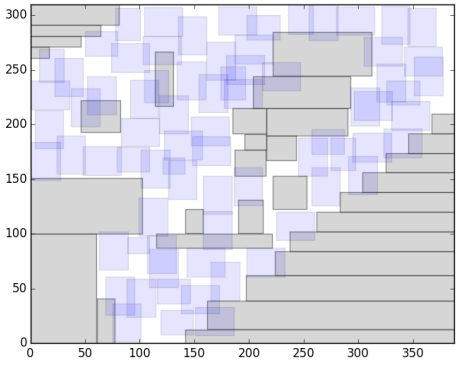
estimation. To simulate a realistic environment the map is designed manually based on a satellite images (see Figure 5.6a). Each rectangle obstacle has been placed to reproduce the buildings as in the satellite images.

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 dimensions to optimize (75 due to the add of z and γ) and the increased size of the area (twice bigger

(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.

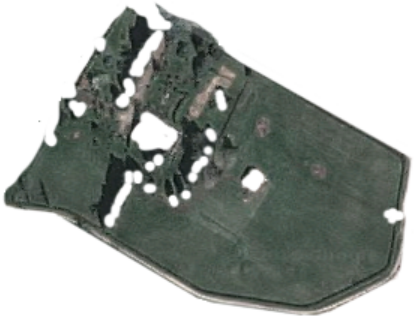
Parameters	Value
z	[0.5;1;1.5;1.75]
γ	portrait or landscape
Size of the area	380x310 px
Maximum size of camera projection	42x54 px
Number of waypoints	75

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

then previously) mark a 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 area,...).

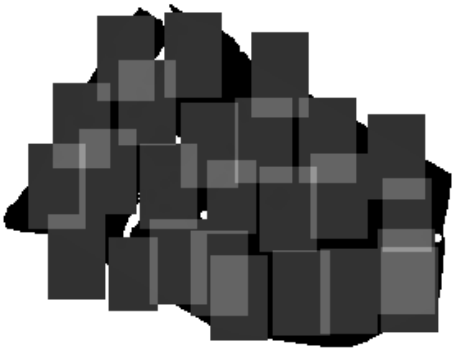
5.5.3/ USING MASK TO DESCRIBE THE AREA

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 Section 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.

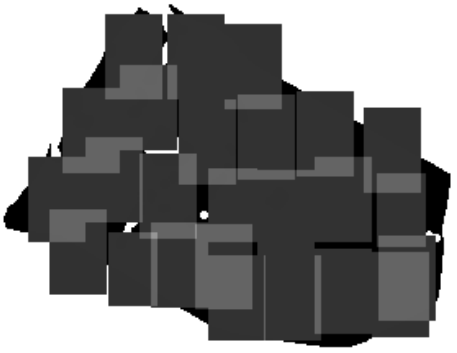


(a) The area to cover taken form satellite images.

(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.

The solution chosen for the flowing experimentations are 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 zones (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 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 (see Figure5.7b). 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 results are visible in Figure 5.7c. The optimized waypoints positioning, cover 90.76% of the area with 25 waypoints 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 experiments presented until now (see Sections 5.5.2 and 5.5.1) only the single GA was employed for the optimization. On the last experiments (see Sections 5.5.2) 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 holes and overlaps (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 compensated it. The experiments allows also to evaluate the improvement made by the GAPSO. The GAPSO hybridization is robust and flexible despite the strong constraints due to the non geometric area.

Thanks to these 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.

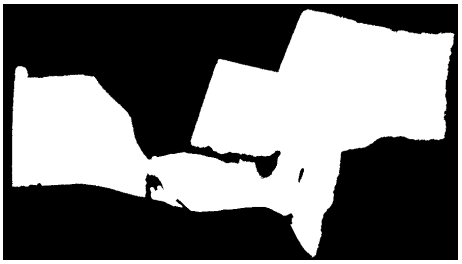
5.5.4/ USING MASK FOR BIGGER AREA

Based on the last experiment (see Section 5.5.3) 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.

5.5.4.1/ VAST AND COMPLEX OUTDOOR

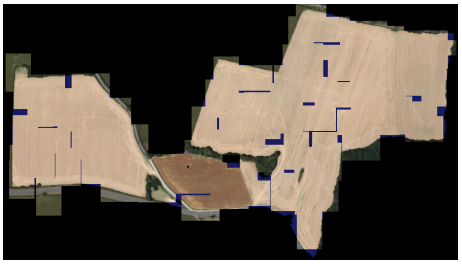
In this experiment, a much bigger and complex area is presented. A much bigger area involves the increasing of the search space and consequently the increasing of the difficulty. In the following example Figure 5.8b the satellite images are used to define the area to control (in white). The size of the area has been increased to have a grid composed by almost half million of points with nearly 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 spaces and holes has been selected.
During this experimentation, a high coverage rate is required. We aspect more than 98% 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

(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.

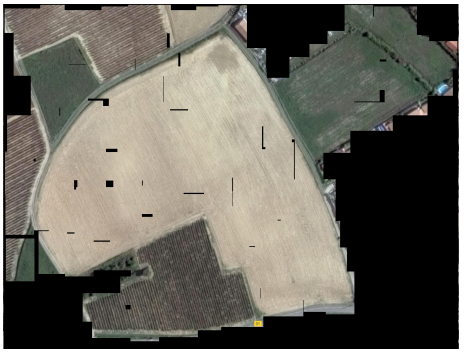
long time before to converge. The convergence time is due to the number of dimension to optimize $(80 \times 4)$and thus the size of the search space.

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 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.

(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 waypoints 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 waypoints position, (d) is the representation of the black hole.

5.5.4.2/ BIGGEST MAP WITH NUMEROUS WAYPOINTS

The precedent experiments (see Section 5.5.4.1) shown the efficiency and flexibility of the GAPSO to the big map with lots of waypoints 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 boundaries.

For these experiments, the area proposed here is the biggest never tested yet (among the one presented). The grid is composed by almost 1 million of points, with more than 616 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 among the more important compared to the literature for examples, in [48, 20, 15, 49, 4, 25, 13, 29, 43, 3]. 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 generations 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 generations and consequently an important time of computations (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 required 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. Reducing the coverage rate required and the consequently the number of camera allow a much faster optimization. For example a similar experiment with 100 waypoints reach with the GAPSO the 92.659% of coverage after only 16 545 generations instead the 170'501 generations.

5.5.5/ WAYPOINTS POSITIONING LIMITATIONS

Among the numerous experiments done, the GAPSO appear as a good solution to optimize the positions and orientations 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 are restricted.
During the different experiments proposed some limitations 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 to have a high coverage rate.
To illustrate this phenomenon the 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 generations. This 200 generations for the GA and a similar number of iterations 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

COVERAGE PATH PLANNING PROBLEM

Contents

The following sections are devoted to the Coverage Path Planning (named CPP) problem. Until now, the work presented was focus on optimizing the position and orientation of a fix number of cameras to cover a vast and complex areas. The method introduced, use GA or a more flexible (but slower) GAPSO. These algorithms give an efficient result to estimate the pose of a set of fix cameras. To find a solution for the CPP problem we propose for the first time to use the algorithms developed for the cameras positioning to find the set of useful waypoints. Instead to use the conventional method based on sweep and spiral in simple polygon. The optimized poses of the cameras are considered as waypoints of a complete path. The detail of the proposed solution to optimize the CPP problem and few experiments made are presented in these sections.

6.1/ SEQUENTIAL METHOD

To optimize the CPP problems a simple and innovative method has been developed. The method proposed here can be decomposed in 3 principal parts all interconnected.

- Number of waypoints : A crucial step is to estimate the number of waypoints. A wrong estimation of a too high or low number of waypoints will cause a bad area coverage or a too complex computation.

- Waypoints positioning : The waypoints positioning optimization is the more crucial step. As already discuses numerous solutions has been studied to optimize the pose of the cameras depending on several constraints. Based on it and the experiments made until now, an efficient algorithm has been chosen and adapted to pose estimate each waypoint (see Section 5.4).
- Path plan computation : When the number and the positions of the waypoints has been computed the last step is to find the shorter route passing by all the waypoints. The route must start and finish at the same position (as for the TSP see Paragraph 2.4.2.1 and detailed in Section 6.1.2).

The proposed method having the advantage to optimize independently all the waypoints and the path plan. This method is also not systematically based on sweep and allows to have a shorter path adapted to a complex map. Moreover the proposed CPP having with the same starting point and ending point. In practice the return to the starting point is meaningful.

6.1.1/ NUMBER OF WAYPOINTS ESTIMATION

It is difficult to estimate properly the minimum of waypoints which are necessary to cover a complex area. To do so, a two-steps procedure has been implemented. The procedure is based on the pose optimization for a fixed number of waypoints introduced in the precedent Chapter 5. The first step is to find the minimum number of waypoints depending on the area to cover like formulated in the Equation 6.1.

$$\frac{A_{room} - \sum_{i=1}^{n} A_{walli}}{A_{cam}} \times \text{Threshold Rate} = \text{NWayPoint} \tag{6.1}$$

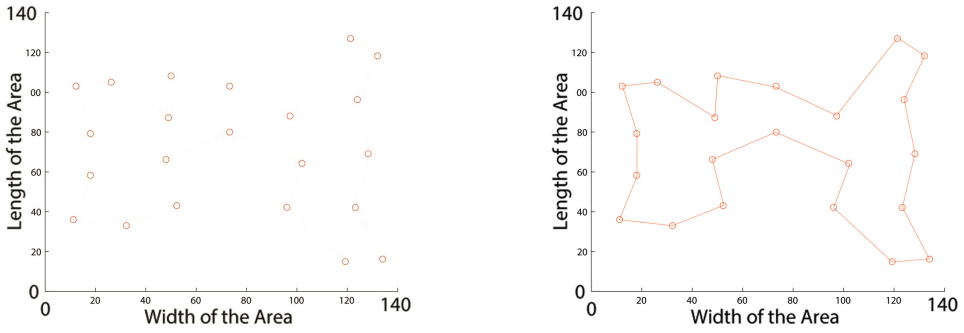
- A_{room} : area of the Room (length $\times$ width)
- A_{Wall} : area of the obstacle like wall (length $\times$ width)
- A_{Cam} : area cover by the camera in the maximum size of z
- NWayPoint : number of waypoints
- Threshold Rate : objective threshold rate
- S : one solution of waypoints set
- $evalCost$: cost function

The second step is to compute GA optimization until the threshold is reached. At the convergence of each GA, if the threshold rate is not reached one more waypoints is added and a new GA optimization start with one more waypoints. The algorithm used to estimate the number of waypoints is explained in the ”Algorithm 1 Estimation of the number of waypoints”.

At the end of these steps, we have the number and a good set of waypoints poses from the last GA convergence. The waypoints poses can directly be used, but can also be refined using the adapted PSO (as in the GAPSO see 5.4). Once the number and the efficient poses of the waypoints founded the next step is to compute the path planning. The path planning has to pass by all the waypoints found before to return to the starting point.

Algorithm 1 Estimation of the number of waypoints

procedure NMBWAYPOINT(A_{room}, A_{Wall}, Threshold Rate, A_{Cam})
  $S \leftarrow 0$
  $NWayPoint \leftarrow \frac{A_{room} - \sum_{i=1}^{n} A_{walli}}{A_{cam}} \times$ Threshold Rate (Equation 6.1)
  while $evalCost(S) \leq ThresholdRate$ **do**
    $S \leftarrow GA(NWayPoint)$
    $NWayPoint \leftarrow NWayPoint + 1$
  return $NWayPoint$



(a) Every pose after the optimization of the waypoint positioning.

(b) Path compute with GA multi objective for TSP.

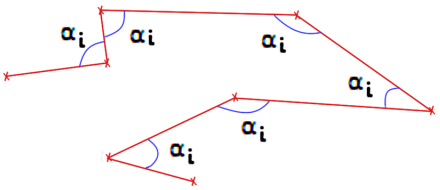
Figure 6.1: Optimization of the path planning.

6.1.2/ SORTED WAYPOINTS AND PATH PLANNING.

In the previous sections, the method to obtain the list of waypoints positions to have a desired coverage has been detailed. Now, the list of waypoints has to be sorted, in order to compute an efficient path with the shorter travelling distance passing by all the waypoints. In order to create an efficient path passing by all optimised waypoints the problem is formalized as a TSP. The TSP was quickly introduced in the Paragraph 2.4.2.1 and more detail about is provided in the following sections.

TRAVELING SALESMAN PROBLEM.

The sorted path can be formulated as Travelling Salesman Problem (TSP). To remember the TSP is inspired by a question asked by a salesman "**What is the shortest path passing by each city only one time and return to the starting city? When i know a list of cities and the distances between each pair of cities.**". The TSP problem is a well known NP-Hard and NP-complete problem (see in [70]) and different solutions exist to optimize it depending of the context. Ponnambalam et al [121] propose using a multi-objective GA to optimize the TSP. Moreover then, the GA used Ponmambalam et al [121] provide the GA set-up (the mutation rate, population size,...). Davies et al [78] propose also to use the GA to solve the TSP applied on robotic with obstacles constraint.
Based on the literature, GA since adapted and efficient enough to optimize the TSP problems. The GA has to be set-up properly depending then the specific TSP problem. The GA set-up was discussed in the Section 3.3.6. The know the more appropriate set-up to optimize the TSP, several studies has been made as [76, 77, 122]. The conclusion of



(a)

Figure 6.2: Extraction of the curve angle in the trajectory.

these studies is to use a simple GA for combinatory problem. That mean the operator has to be adapted (using swap for example) with a high mutation rate and a very elitist selection. To illustrate the GA ability one example using an adapted GA (high mutation rate, elitist selection and combinatory formulation) solution is provided in the Figure 6.1.

COMPLEXITY OF TRAJECTORY.

Once the waypoints position estimated and the shorter path passing by all the waypoints computed it is interesting to can evaluate the trajectory complexity. The trajectory complexity can be a great clue of the path reliability. It also be useful to compare our solution to the more classical sweep trajectory.

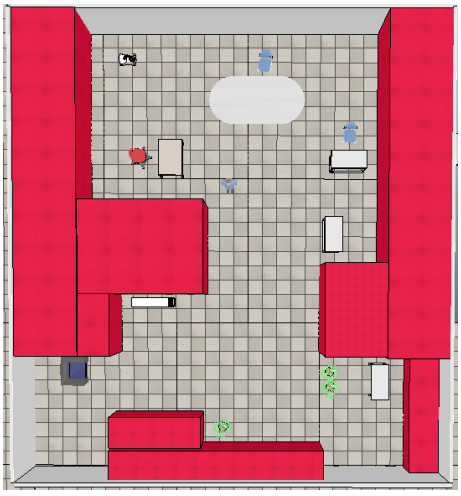
To estimate the trajectory, two indicators are used to evaluate properly the path complexity. The first indicator is the distance of the trajectory. The distance allows to evaluate basically the optimization of the path planning. This indicator is directly included in the optimisation process as discussed before (see Section 6.1.2). Evaluate the trajectory only by using the distance indicator is not enough and another indicator must be associate to can evaluate quickly the trajectory complexity. To estimate the complexity in terms of curve for the UAV evolving in the 3D space the angles at each node is studied. The complexity of trajectory indicator is computed as follow:

$$\text{Trajectory complexity} = \frac{\sum_{i=1}^{size(\alpha)} 180 - \alpha_i}{size(\alpha)} \tag{6.2}$$

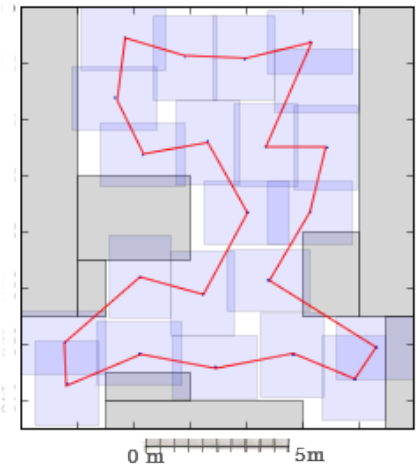
Where α_i is an angle of curve in the trajectory as in Figure 6.2.
$Size(\alpha)$ is the number of curves in all the trajectory.
This method can give an idea of the global trajectory complexity despite this simplicity. The two indicator presented (the distance and the trajectory complexity) are used during the next sections to evaluate the gain of the proposed method.

6.2/ EXPERIMENTS

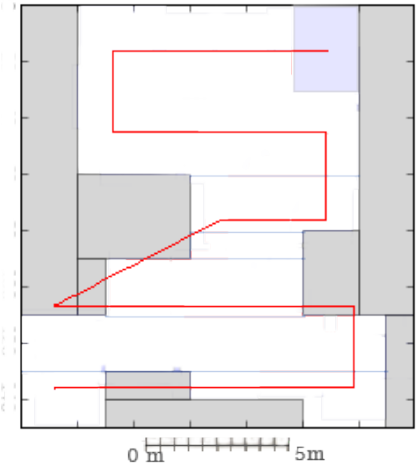
The proposed method for the problem of CPP was tested during different experimentations. The experimentations brings out the advantages and the limits of the developed method. The experiments are structured in sections with show the method and algorithms in more and more complex experimentations.



(a) Room Coverage



(b) Pose of every images with the path planning



(c) Room in V-rep

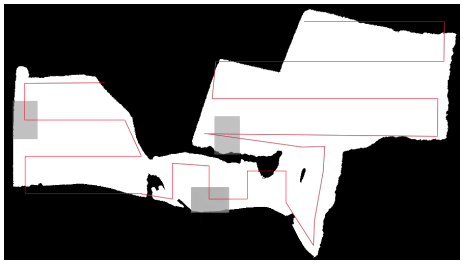
	Coverage rate	**Path length (in px)**	**Trajectory complexity (Eq.6.2)**
Pattern Method	100%	1137px	97.81°
Ours Method	90%	881px	65.21°

Figure 6.3: optimisation of the path planing

6.2.1/ RECTANGLE OBSTACLE

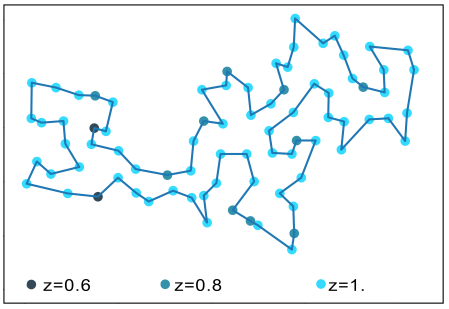
In order to experiments the proposed method for CPP a simple indoor experiment is proposed to begin. The experiments proposed are based on the one discussed previously on Section 5.5.1. To remind the experiments are made in a simulated room $(15 \times 14m^2)$. The area to cover is shown in the Figure 6.3a. The camera parameters are the same, $4 \times 3m^2$ when z is equal to one and the z factor can be equal at $[0.5; 1; 1.5]$. On the other hand, the sweep is computed with the camera at the maximum altitude to have the biggest area coverage possible.

Thanks to this first experiment the path planning proposed appear much more appropriate (see Figure 6.3). The path planning obtained by our method, proposed a shorter path,



(a) Path planing using the pattern method.

(b) 80 waypoints for 98.28% of coverage.



(c) Path with 80 waypoints for a distance of 4015.6 px.

	Coverage rate	**Path length (in px)**	**Trajectory complexity (Eq.6.2)**
Pattern Method	100%	4362.66px	82.14°
Ours Method	98.28%	4015.6px	65.78°

Figure 6.4: Experimentation of coverage path planning in the outside area.

881.6 pixel length when the path by sweeping is at 1137 pixel length. The path proposed by the sweep is longer notably due to the return to the starting point, but not only. The junction between the start and finish of the swept path is 245 pixel long, with mean the path useful to fully cover the area is around 892 pixel long. Moreover the proposed path planning in addition then the shorter path proposes also a better trajectory complexity with 65.22° instead to 97.81° for the path with sweep. Finally the proposed solution allows a better path planning thanks to this optimized waypoints. Despite a better path plan efficiency few points (g_i) in the area are not covered in contrary than the traditional swept method.

6.2.2/ Using Mask to describe area.

The gain for the indoor area is important. The gain is mostly due to the well optimised position of the waypoints. The experiment must be extended for a bigger and complex area with a higher coverage rate requirement.

6.2.2.1/ Vast and complex area.

The CPP and more precisely the path planning part is compared to a standard method (the sweeping). We compare the CPP in term of coverage, path plan distance and

trajectory complexity. The standard method used for CPP uses an adapted pattern to cover an area. The patterned method application is based on several articles as [72, 22, 60, 53, 123] (see more about the cellular decomposition and the sweeping methods in Section 2.4.2.1).

To focus on a bigger area the experiment for outdoor waypoints positioning in vast and complex area using a mask is reused as in Section 5.5.4.1. This experiment has the advantage to be vast and complex enough for the comparison between the sweeping method and the proposed method.
To compute the sweep, the area is firstly decomposed in cells to apply an appropriate sweep. In the second time, the appropriate sweep are placed in each cell. The size of the sweep has been designed for the fix altitude depending then the size of the camera projection on to the floor at the highest altitude spite the resolution. The path plan using a sweeping method is visible in the Figure 6.4a. In order to cover completely the area several overlap and un-interesting region (the black region in Figure 6.4a) has to be covered too.

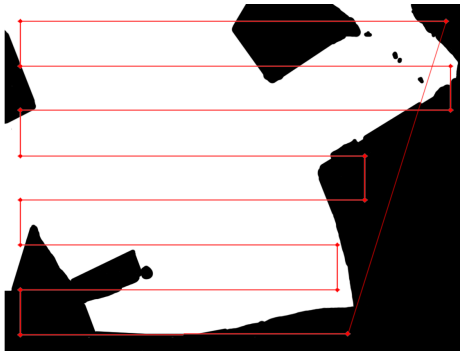
The solution proposed by optimizing in a first time the waypoints position and in second time the path passing by the waypoints is applied. The optimization is made for a set of 80 waypoints which must cover over the 98% of the area. This threshold is reached after a 3101 generations (see in 6.4b). Once the waypoints optimized (in $x; y; z$ and roll) the path plan is computed using the method based on the TSP (presented in the Section 6.1.2). The final CPP is visible in the Figure 6.4c with the different altitudes represented in colors.

One more time and despite the increased size and complexity of the map (with increase greatly the complexity and the number of the waypoints optimization) the proposed solution give a better result. The path plan is shorter 4362px for the pattern method versus 4015px for our method. The path is also easier to compute in term of trajectory 82.14° versus 65.78°. In fact, the optimization of the waypoints despite a not completely covered area offer good waypoints for the path planning. In fact the coverage rate over the 98% of coverage can be considered more then enough depending then the application.

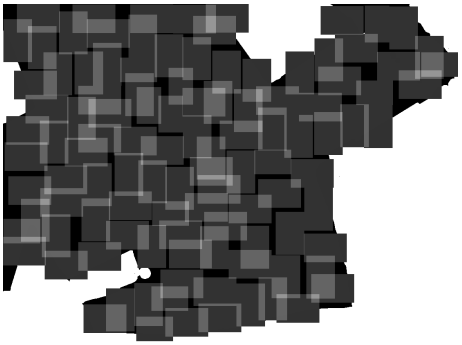
6.2.2.2/ BIGGEST MAP WITH NUMEROUS WAYPOINTS

To continue the experiment a biggest area can be selected with require much more waypoints. The big area selected for test the coverage path planning is the same area then in Section 5.5.4.2. The simple sweep path is computed based on the camera projection at the highest altitude (see in Figure 6.5a). Our method is computed with 110 waypoints to estimate the poses. After 3634 generation for a coverage area at almost 95% (see Figure 6.5b) the path can be computed (as explained in the section 6.1.2). The final CPP is shown in the Figure 6.5c.

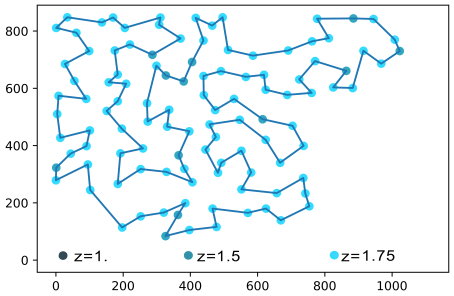
The solution proposed give a shorter path compare than the swept method (8173px versus 8582px) for a high level of waypoints coverage. But the difference between the 2 paths are not so important in term of distance. But the difference of trajectories complexity gives an important advantage of the method proposed.
Thanks to these experiments the limit of our method begins to appears. The principal limitation is due to the difficulty to optimize numerous waypoints (as that was discussed in the Section 5.5.4.2). The increased number of waypoints associate to a high level of coverage rate required, push the GAPSO optimization to this limits. Numerous generations are



(a) Path planing using the pattern method.



(b) 110 waypoints for 94.862% of coverage.



(c) Path with 80 waypoints for a distance of 4015.6 px.

	Coverage rate	**Path length (in px)**	**Trajectory complexity (Eq.6.2)**
Pattern Method	100%	8582px	90°
Ours Method	94.86%	8173px	69.75°

Figure 6.5: Experimentation of coverage path planning in the outside area.

necessaries before to converge. On the other hand, the path plan computation appear efficient despite the increased number of waypoints and globally the solutions proposed give a better CPP in term of distance and trajectories complexity for a high coverage rate then the classical sweep.

6.2.2.3/ CAMERA CONTRAINT BY THE TRAJECTORY

The previous indoor or outdoor simulations, assumed to have an UAV with a camera stabilization for the roll, in order to stabilise the roll (portrait or landscape) independently then the UAV trajectory. During this simulation we assume the UAV is not able to control the roll independently of the direction and the UAV is sufficiently agile to follow the path. The solution proposed in this section is adapted to the UAV trajectory to constrains the camera directions. In the following experimentation the roll and the altitude are fixed for the waypoints positioning. The camera size is reduced as a scare shape projection onto the ground 40x40px, corresponding to the smaller side of the camera projections. The camera is represented as a reduced scare is used only for the optimization process. The reduction of the camera estimation, the fixed roll and also a fix altitude premiss us to make a coverage at minima of the area. The optimisation of the waypoints are computed (as in

(a) 115 waypoints poses after optimisation for a coverage of 94.3%.

(b) Coverage path planning for 99.71% of coverage with a distance of 4785px.

	Waypoints coverage rate	**Stream coverage rate**	**Path length (in px)**
Our Method	94.3%	99.7%	4785px

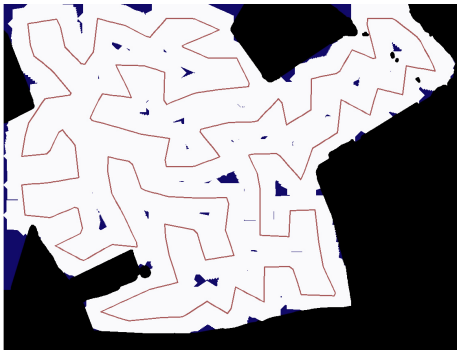
Figure 6.6: Outdoor simulation of the coverage path planning. The camera projection FOV is $40 \times 60px$ and fix altitude(with a scare projection of 40px wild for the positioning optimisation).

Chapter 5) with these new constraints. Once the waypoints positioned the path planning can be computed as presented in Section 6.1.2 to have the sorted waypoints.

To estimate the final coverage of the area with a fixed camera on the UAV, the UAV direction have to be estimated before. Based on the path computed with the TSP formulation as Section 6.1.2 the UAV orientations are deduced by estimating the trajectory to go one waypoint to another. The next step is to estimate the coverage of the area with the camera stream record and the cameras FoV oriented depending of the trajectories (see Figure 6.6b). The final stream coverage estimation is done with the camera size 60x40 pixel.
Obviously the final coverage path planning is better than the estimation did it during the waypoints positioning, see in the following example Figure 6.6a. The camera projection for the optimisation is 40 pixel by 40 pixel with 115 waypoints for coverage pose estimation of 94.3% and 99.71% for stream coverage of the full room with the simulation of path planning and a cameras projection to 40x60 pixel (see Figure 6.6b). This gain is not taken into account during the phase of waypoints positioning optimization and in consequence not for estimate the number of waypoints (see Section 6.1.1).
To confirm the gain of stream coverage an other experiments has been made with a similar condition but in a different maps. The experiment is made with the map introduced in Section 5.5.4.2 with the camera projection of 70 pixel by 70 pixel with 110 waypoints for a coverage estimation at 88.38% (see Figure 6.7a) and 98.64% (see Figure 6.7b) for the full room after the simulation.

This experiment demonstrates the ability of our proposed method to do coverage path planning in a large area with a non convex shape with different constraints. The solution proposed being adaptable and flexible to the different conditions. This experiment also shows the non negligible gain of coverage between the waypoints positioning optimization and the final CPP with the stream coverage of the area.

(a) 110 waypoints for 88.38% of coverage fix altitude .



(b) Path planing using the pattern method Coverage path planning for 98.6% of coverage with a distance of 8114px.

	Waypoints coverage rate	**Stream coverage rate**	**Path length (in px)**
Our Method	88.38%	98.6%	8114px

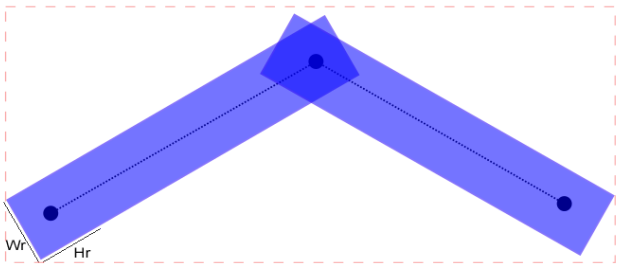
Figure 6.7: Outdoor simulation of the coverage path planning. The camera projection FOV is $40 \times 60px$ and fix altitude(with a scare projection of 40px wild for the positioning optimisation).

6.3/ CPP GLOBAL OPTIMIZATION ATTEMPT

The experiments made previously were successful despite the different optimization steps. In fact to can compute an efficient CPP several optimization has been used (see example Section 6.2). First number of waypoints has to be estimated and the waypoints poses has to be optimised. When the coverage rate is reached the optimized waypoints can be refined with a PSO to have an appropriate coverage. The final optimization is to use the TSP paradigm to compute a path passing by all the waypoints (as explained in Section 6.1).
Despite the success of this greedy method a logical improvement clue maybe to reduce the number of successive optimizations. In the experiments proposed previously the waypoints positioning optimization shown some beginning of limit due to the too hight number of dimensions to optimize (see Section 5.5.5). In the experiment made for a camera constrain by the UAV trajectory (see Section 6.2.2.3), the streamed coverage area increase significantly compared to the waypoints coverage. This increased area covered is not taking in account until now. The idea is to try to limit the number of waypoints by using a global method to optimize the CPP problems.
The method explored in the following sections is to merge the optimisation of the waypoints positioning and the sorted waypoints process to have only one global optimization for the CPP problem.

6.3.1/ ADAPTED FORMULATION

In order to find a solution for the problem of CPP, by one optimization of the waypoints positioning and the path planning, the problem has to be re-adapted. Until now, the waypoints positioning and the path planning computation were made by using two different



(a)

Figure 6.8: Covered area between waypoints following the path plan. Where Wr and Hr are the size of the camera projection as defined in the Section 4.2.1.2.

GA during two different optimizations process. The idea here is to combine the cost function from the waypoints positioning and the path planning computation to have an appropriate and global cost function. To create this cost function few things are necessary.

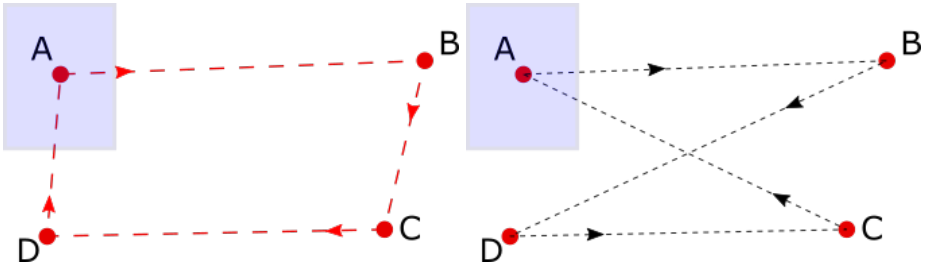
- First is to estimate the area covered. The area has to take in account not only the area covered by each waypoints but also the covered area during the path (stream coverage). In contrary then the coverage estimation used in the previous sections with are discretized (only at each waypoint).
- Second is the path distance. The path distance was already used to estimate the path passing by all the given waypoints during the path computation (as in Section 6.1.2). In this case that imply to have beforehand the sorted waypoints for the path.

Coverage path plan estimation

To formulate properly the CPP problem is essential to can estimate the area covered by the UAV during the path passing by all the optimized waypoints. For that the waypoints and the path must be knew. Once the path knew the coverage estimation can begin.
The idea is to consider all the points of the grid G (see Section 4.1 for the grid definition) between two waypoints as covered. To can compute the points cover during the fly over two waypoints the camera fixed on the UAV as to be known in term of projection size onto the floor and the direction. In fact, the size of the camera projection and the direction will change the coverage. The direction is directly deducted from the waypoints positions and the size of the camera projection form the camera parameters as this altitude (see Section 4.2.1). Based on the projection size onto the floor the area cover between the waypoints are deduced as illustrate in Figure 6.8.

That mean in this case the roll of the camera is not optimized but taken in account depending then the trajectory to follows. Moreover in this case the altitude of the UAV is fixed in order to simplify the optimization computations and simplify the cost function.

The methodology proposed to estimate the area covered by the path require to have a set of waypoints ordered. In the previous algorithms the TSP paradigms with a GA optimisation was used to order the waypoints, to have the shorter path passing by the set of optimized waypoints. In this new method, the path as to be compute in the same optimization then the waypoints positioning. To do that the solution is to priories the chromosomes order (or particles). The idea is to use the position of each each genome inside



(a) ABCD (b) ABDC

Figure 6.9: Influence of the waypoints order in the new cost function

the chromosome as important in the estimation. Where in this case a genome is the set of parameters of a waypoint. That mean the same genome at different position in the chromosome has not the same impact in the cost function. Consequently the optimization is not only focus on finding the best coverage for a set of waypoints but also find the sequence of waypoints position. This formulation add some combinator constraint inside each chromosome and particle. To illustrate, an example is provided in the Figure 6.9 : The solution ABCD (see Figure 6.9a) where A; B; C and D are the optimized positions of the waypoints and another solution ABDC (see Figure 6.9b) which has found the same waypoints but not in the same order. In the precedent proposed method only the coverage at each waypoint was used to estimate the coverage. Therefore the cost function estimate the two solutions as identical. In this new estimation of the coverage by taking in account not only the coverage of a set of waypoint, but the entire path plan coverage the order of the position give the path to follow. Consequently the solutions are completely different, ABCD (see Figure 6.9a) is much better due to this shorter path for the equivalent coverage.

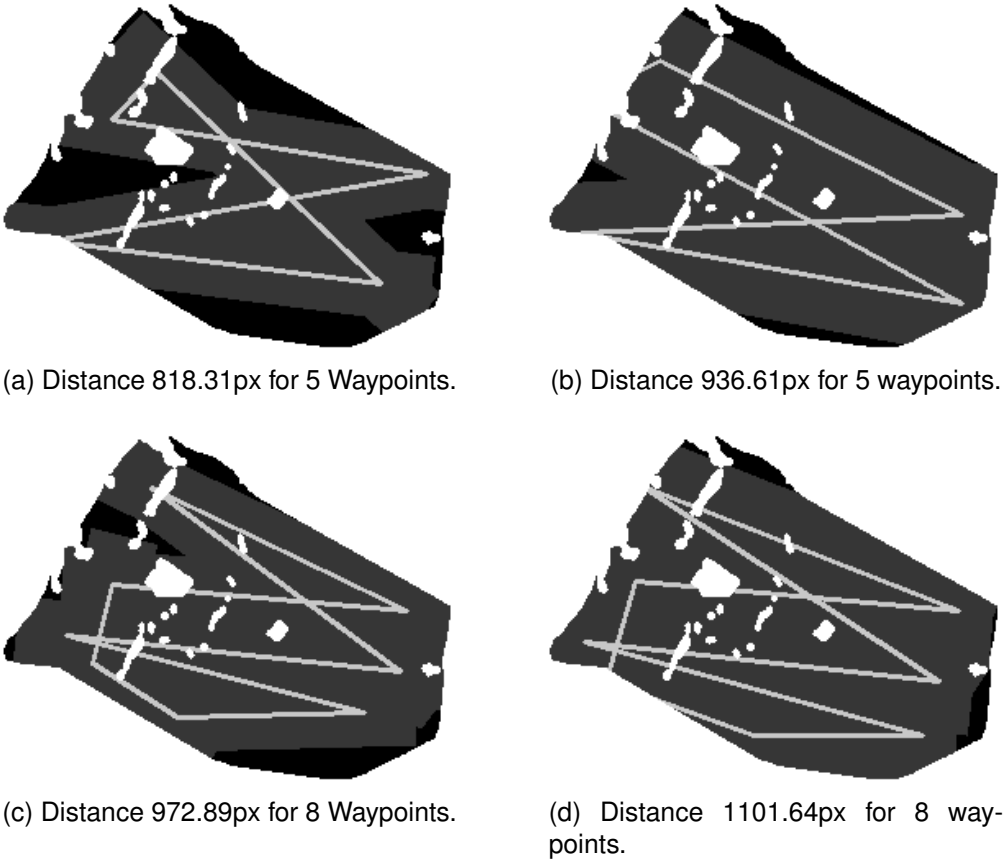
Finally the new cost function will integrate the covered area by the path passing by all the waypoints and the distance of this path. The goal is to have a cost function appropriate to maximize the area covered with the shorter path at same time. The cost function dedicated to quantify the CPP for the optimization of waypoints positioning and path planning is :

$$f = \sum_{i=1}^{n} Pc_i + \frac{\sum_{i=1}^{n} Pc_i}{(\frac{Distance}{N}) \times 10} \tag{6.3}$$

Where $\sum_{i=1}^{n} Pc_i$ is the number of points cover by the path plan (as discussed in the paragraph 6.3.1 and Section 4.2) and the $Distance$ is the size of the path. The 0.5 is an empirical coefficient to minimize a bite the distance importance in favour to the coverage.

Optimization

Once the answer quality evaluate thanks to this redefined cost function, it is time to discuses about the optimization process. One more time the GA is used associate to the PSO. If the EA (especially the GA) are used here is for few main reasons. The GA was the best algorithms to solve the TSP problem and also well adapted to the waypoints positioning (even more with an GAPSO). The last reasons are the knowledge grab about the GA PSO until know make the implementation fast and easy. The solution adopted is



(a) Distance 818.31px for 5 Waypoints.

(b) Distance 936.61px for 5 waypoints.

(c) Distance 972.89px for 8 Waypoints.

(d) Distance 1101.64px for 8 waypoints.

Figure 6.10: Path plan for maximize the covered area with an cameras projection equal to 40x60. The area is covered with 5 and 8 waypoints

to use the GAPSO as introduced in 5.4 associate to the newly adapted cost function (see Paragraph 6.3.1).

6.3.2/ Results and consequences

Based on the new problems formulation and the use of an appropriate GAPSO directly inherited then the waypoints positioning (see Chapter 5) few experiments has been made. The experiments proposed here is based on the map presented in the Section 5.5.3. Once the map selected the optimization begin with three waypoints and the number of waypoints is slightly increased until reach nine waypoints. For each number of waypoints three optimizations has been made. The experiment proposed in the Figure 6.10 are the interesting and representatives then the experiments made.

Experiments and discussions

The experiments made allows to see the advantage and limit of this optimisation. The main advantage of this formulation is the beside to optimize the CPP problem in one unique optimisation is the reducing amount of waypoint compared to the waypoint positioning optimization. In fact for the map proposed only fife waypoints are required to cover

the area (see Figure 6.10b) compare then the waypoints positioning optimisation require 25 waypoints (see Figure 5.7d). Reducing the number of waypoints needed to cover an area is essential to have an better and faster optimization process. The reducing amount of the number of waypoints is even more important when the problem is complicated by the position order of the waypoints in a solution. On the other side, despite a good coverage of the area the path plan appear too long with many overflight (overlaps). Consequently, and despite the good coverage the path appear not so well optimized. About the optimization we can observe the average number of generations of the GA required is around 47 generation for a set of experiment between 3 to 9. The average number of generations are comparable to the number of generations needed to optimize the waypoints positioning (see in Section 5.5.3), where for 25 waypoints to pose estimate 67 generation was needed. The fast studies of these experiments, show some limits of using the unique optimization for the problem of CPP. In the other hand, the formulation despite the inconclusive, but promising result appear interesting and the cause of the inconclusive results are explored.

Why inconclusive results?

Despite inconclusive results during the first experiments made, it is an interesting track to follow which must be more explored. The inconclusive results of the experiment can be from :

- The algorithms: As that was introduced the GAPSO with a similar set-up than the previous experiment for the waypoints positioning (see Chapitre 5) are used. Few inconclusive tests have been done to slightly modify the set-up of the GAPSO by adding more mutation for example. The inconclusive preliminary result push to consider the GAPSO proposed previously good enough and the other reason has to be explored.

- The cost function: The cost function presented in the Equation 6.3 is the more appropriate due to our experiments. Despite that the cost function has some coefficients which they were found empirically. The coefficients could be an interesting track which must be thorough in order to reduce their uncertainty.

- Complexity and search space: The last potential reason of the inconclusive results obtained can be the increased size of the search space and complexity. The size of the search space was discussed in the Section 4.4. The search space estimation was made for the problem of optimizing the waypoints positions and can be reused by part for the problem of CPP for a unique optimization. First the Equation 4.24 can be simplified.

$$S\,p = (W \times H) \tag{6.4}$$

  Where W and H are the size as width and height of the area to cover. $S\,p$ is simplified due to the reduced number of dimensions optimized. In fact, the altitude is fixed and identical for all the waypoints as the pant and tilt. Regarding the roll depend then the path plane. Despite the important decreasing size of the search space for each waypoint ($S\,p$), the global search space for the CPP increase dramatically due to the number of waypoint and the ordered constraint. In fact due to the ordered

constraints and the global search space is re-defined as :

$$A_n^{Sp} = \frac{Sp!}{(Sp-n)!} = |Vs| \tag{6.5}$$

Where $|Vs|$ is the number of possible solution for a set of n cameras. Due to the importance of the waypoints order in the optimization process the number of possible solution increase greatly. Obviously due to the complexity of the problem (the complexity is comparable to the WRP see in Section 2.4.1) and the increased size of the global search space for CPP problem can be one of the important reason of the inconclusive result.

The conclusion is despite the low quality of the answer obtained with these methodologies the interesting track and possible gain must push the research to continue to explore the GAPSO optimization to solve the problem of coverage path planning problem as a one optimization problem. But caution the increased search space can become break of the optimization despite the reducing of the number of dimensions to optimize. In fact, the number of dimensions is reduced by fixing the altitude and the camera orientation and moreover with this unique optimization the number of required waypoints are greatly reduced. The main risk of using a unique optimization for the problems of CPP is the complexity and the associate search space. In fact, among the reason to explain the inconclusive results, the complexity due to the fusion of 2 NP-hard and NP-complete problems could be the cause.

BIBLIOGRAPHY

[1] Jose Luis Alarcon Herrera and Xiang Chen. Online configuration of ptz camera networks. In *Distributed Smart Cameras (ICDSC), 2012 Sixth International Conference on*, pages 1–6. IEEE, 2012.

[2] Cristian Soto, Bi Song, and Amit K Roy-Chowdhury. Distributed multi-target tracking in a self-configuring camera network. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1486–1493. IEEE, 2009.

[3] Chong Ding, Bi Song, Akshay Morye, Jay A Farrell, and Amit K Roy-Chowdhury. Collaborative sensing in a distributed ptz camera network. *IEEE Transactions on Image Processing*, 21(7):3282–3295, 2012.

[4] Jian Zhao, Ruriko Yoshida, Sen-ching Samson Cheung, and David Haws. Approximate techniques in solving optimal camera placement problems. *International Journal of Distributed Sensor Networks*, 9(11):241913, 2013.

[5] Yi-Chun Xu, Bangjun Lei, and Emile A Hendriks. Camera network coverage improving by particle swarm optimization. *EURASIP Journal on Image and Video Processing*, 2011(1):458283, 2011.

[6] Jian Zhao, Sen-Ching Cheung, and Thinh Nguyen. Optimal camera network configurations for visual tagging. *IEEE Journal of Selected Topics in Signal Processing*, 2(4):464–479, 2008.

[7] Yan Li, Hai Chen, Meng Joo Er, and Xinmin Wang. Coverage path planning for uavs based on enhanced exact cellular decomposition method. *Mechatronics*, 21(5):876–885, 2011.

[8] Christof Hoppe, Andreas Wendel, Stefanie Zollmann, Katrin Pirker, Arnold Irschara, Horst Bischof, and Stefan Kluckner. Photogrammetric camera network design for micro aerial vehicles. In *Computer vision winter workshop (CVWW)*, volume 8, pages 1–3, 2012.

[9] Stefanos Nikolaidis, Ryuichi Ueda, Akinobu Hayashi, and Tamio Arai. Optimal camera placement considering mobile robot trajectory. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 1393–1396. IEEE, 2009.

[10] Robert Bodor, Paul Schrater, and Nikolaos Papanikolopoulos. Multi-camera positioning to optimize task observability. In *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, pages 552–557. IEEE, 2005.

[11] Xiao-Shan Lu, Hai-Jun Huang, and Jiancheng Long. A bi-level programming model for network traffic surveillance of optimal camera location. In *Computational Sciences and Optimization (CSO), 2011 Fourth International Joint Conference on*, pages 1035–1039. IEEE, 2011.

[12] Robert Bodor, Andrew Drenner, Michael Janssen, Paul Schrater, and Nikolaos Papanikolopoulos. Mobile camera positioning to optimize the observability of human activity recognition tasks. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1564–1569. IEEE, 2005.

[13] Yibo Jiang, Jinwei Yang, Weijie Chen, and Wanliang Wang. A coverage enhancement method of directional sensor network based on genetic algorithm for occlusion-free surveillance. In *Computational Aspects of Social Networks (CASoN), 2010 International Conference on*, pages 311–314. IEEE, 2010.

[14] Eva Hörster and Rainer Lienhart. On the optimal placement of multiple visual sensors. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pages 111–120. ACM, 2006.

[15] Dimitrios Chrysostomou and Antonios Gasteratos. Optimum multi-camera arrangement using a bee colony algorithm. In *Imaging Systems and Techniques (IST), 2012 IEEE International Conference on*, pages 387–392. IEEE, 2012.

[16] Liang Liu, Xi Zhang, and Huadong Ma. Dynamic node collaboration for mobile target tracking in wireless camera sensor networks. In *INFOCOM 2009, IEEE*, pages 1188–1196. IEEE, 2009.

[17] Dalei Wu, Song Ci, Haiyan Luo, Yun Ye, and Haohong Wang. Video surveillance over wireless sensor and actuator networks using active cameras. *IEEE Transactions on Automatic Control*, 56(10):2467–2472, 2011.

[18] Katayoun Sohrabi, Jay Gao, Vishal Ailawadhi, and Gregory J Pottie. Protocols for self-organization of a wireless sensor network. *IEEE personal communications*, 7(5):16–27, 2000.

[19] Liang Liu, Xi Zhang, and Huadong Ma. Optimal node selection for target localization in wireless camera sensor networks. *IEEE Transactions on Vehicular Technology*, 59(7):3562–3576, 2010.

[20] Krishna Konda Reddy and Nicola Conci. Camera positioning for global and local coverage optimization. In *Distributed Smart Cameras (ICDSC), 2012 Sixth International Conference on*, pages 1–6. IEEE, 2012.

[21] Nirupama Bulusu, Deborah Estrin, Lewis Girod, and John Heidemann. Scalable coordination for wireless sensor networks: self-configuring localization systems. In *International Symposium on Communication Theory and Applications (ISCTA 2001), Ambleside, UK*, 2001.

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

[23] Wang Meiting, Tan Shili Ding Junjian, and Yan Liwen. Complete coverage path planning of wall-cleaning robot using visual sensor. In *Electronic Measurement and Instruments, 2007. ICEMI'07. 8th International Conference on*, pages 4–159. IEEE, 2007.

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

[25] Chang Wang, Fei Qi, and Guang-Ming Shi. Nodes placement for optimizing coverage of visual sensor networks. *Advances in Multimedia Information Processing-PCM 2009*, pages 1144–1149, 2009.

[26] Zhao Zhang, James Willson, Zaixin Lu, Weili Wu, Xuding Zhu, and Ding-Zhu Du. Approximating maximum lifetime k-coverage through minimizing weighted k-cover in homogeneous wireless sensor networks. *IEEE/ACM Transactions on Networking*, 24(6):3620–3633, 2016.

[27] Nabajyoti Medhi and Manjish Pal. Sixsoid: A new paradigm for k-coverage in 3d wireless sensor networks. *arXiv preprint arXiv:1401.0200*, 2013.

[28] Uğur Murat Erdem and Stan Sclaroff. Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. *Computer Vision and Image Understanding*, 103(3):156–169, 2006.

[29] Yi-Ge Fu, Jie Zhou, and Lei Deng. Surveillance of a 2d plane area with 3d deployed cameras. *Sensors*, 14(2):1988–2011, 2014.

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

[31] Eli Packer. Computing multiple watchman routes. *Lecture Notes in Computer Science*, 5038:114–128, 2008.

[32] Joseph O'rourke. *Art gallery theorems and algorithms*, volume 57. Oxford University Press Oxford, 1987.

[33] Marcelo C Couto, Pedro J de Rezende, and Cid C de Souza. An exact algorithm for minimizing vertex guards on art galleries. *International Transactions in Operational Research*, 18(4):425–448, 2011.

[34] Shachar Fleishman, Daniel Cohen-Or, and Dani Lischinski. Automatic camera placement for image-based modeling. In *Computer Graphics Forum*, volume 19, pages 101–110. Wiley Online Library, 2000.

[35] Kenichi Yabuta and Hitoshi Kitazawa. Optimum camera placement considering camera specification for security monitoring. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 2114–2117. IEEE, 2008.

[36] Mahdi Moeini, Alexander Kröller, and Christiane Schmidt. Une nouvelle approche pour la résolution du problème de la galerie d'art.

[37] Yuhao Wang, Ge Dang, Yang Si, and Huilin Zhou. An inversion propagation model using ga for coverage prediction of a single urban cell in wireless network. In *Communications, 2008. ICC'08. IEEE International Conference on*, pages 4894–4898. IEEE, 2008.

[38] Haluk Rahmi Topcuoglu, Murat Ermis, and Mesut Sifyan. Hybrid evolutionary algorithms for sensor placement on a 3d terrain. In *Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on*, pages 511–516. IEEE, 2009.

[39] Huan Ma, Meng Yang, Deying Li, Yi Hong, and Wenping Chen. Minimum camera barrier coverage in wireless camera sensor networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 217–225. IEEE, 2012.

[40] Raghavendra V Kulkarni and Ganesh Kumar Venayagamoorthy. 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, 2011.

[41] Krishnendu Chakrabarty, S Sitharama Iyengar, Hairong Qi, and Eungchun Cho. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE transactions on computers*, 51(12):1448–1453, 2002.

[42] Vahab Akbarzadeh, Christian Gagné, Marc Parizeau, Meysam Argany, and Mir Abolfazl Mostafavi. Probabilistic sensing model for sensor placement optimization based on line-of-sight coverage. *IEEE Transactions on Instrumentation and Measurement*, 62(2):293–303, 2013.

[43] Meizhen Wang, Xuejun Liu, Yanan Zhang, and Ziran Wang. Camera coverage estimation based on multistage grid subdivision. *ISPRS International Journal of Geo-Information*, 6(4):110, 2017.

[44] Na Li and Jason R Marden. Designing games for distributed optimization. *IEEE Journal of Selected Topics in Signal Processing*, 7(2):230–242, 2013.

[45] Bi Song, Cristian Soto, Amit K Roy-Chowdhury, and Jay A Farrell. Decentralized camera network control using game theory. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pages 1–8. IEEE, 2008.

[46] Anton van den Hengel, Rhys Hill, Ben Ward, Alex Cichowski, Henry Detmold, Chris Madden, Anthony Dick, and John Bastian. Automatic camera placement for large scale surveillance networks. In *Applications of Computer Vision (WACV), 2009 Workshop on*, pages 1–6. IEEE, 2009.

[47] Weicai Zhong, Jing Liu, Mingzhi Xue, and Licheng Jiao. A multiagent genetic algorithm for global numerical optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(2):1128–1141, 2004.

[48] Pu Zhou and Chengnian Long. Optimal coverage of camera networks using pso algorithm. In *Image and Signal Processing (CISP), 2011 4th International Congress on*, volume 4, pages 2084–2088. IEEE, 2011.

[49] KB Maji, Atreye Ghosh, R Kar, D Mandal, and SP Ghoshal. An evolutionary algorithm based approach for vlsi floor-planning. In *Science and Technology (TICST), 2015 International Conference on*, pages 248–253. IEEE, 2015.

[50] Xiao Fu, Ru-chuan WANG, Li-juan SUN, and WU Shuai. 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, 2010.

[51] Gopalakrishnan Vijayan and R-S Tsay. A new method for floor planning using topological constraint reduction. *IEEE transactions on computer-aided design of integrated circuits and systems*, 10(12):1494–1501, 1991.

[52] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

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

[54] João Valente, David Sanz, Jaime Del Cerro, Antonio Barrientos, and Miguel Ángel de Frutos. Near-optimal coverage trajectories for image mosaicing using a mini quad-rotor over irregular-shaped fields. *Precision agriculture*, 14(1):115–132, 2013.

[55] David W Casbeer, Derek B Kingston, Randal W Beard, and Timothy W McLain. Cooperative forest fire surveillance using a team of small unmanned air vehicles. *International Journal of Systems Science*, 37(6):351–360, 2006.

[56] Chaomin Luo, Simon X Yang, Deborah A Stacey, and Jan C Jofriet. A solution to vicinity problem of obstacles in complete coverage path planning. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 612–617. IEEE, 2002.

[57] Tae-Kyeong Lee, Sang-Hoon Baek, Se-Young Oh, and Young-Ho Choi. Complete coverage algorithm based on linked smooth spiral paths for mobile robots. In *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, pages 609–614. IEEE, 2010.

[58] Simon X Yang and Chaomin Luo. A neural network approach to complete coverage path planning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):718–724, 2004.

[59] Pablo J Zarco-Tejada, José AJ Berni, L Suárez, and E Fereres. A new era in remote sensing of crops with unmanned robots. *SPIE Newsroom*, pages 2–4, 2008.

[60] Haiyang Chao, Marc Baumann, Austin Jensen, YangQuan Chen, Yongcan Cao, Wei Ren, and Mac McKee. Band-reconfigurable multi-uav-based cooperative remote sensing for real-time water management and distributed irrigation control. *IFAC Proceedings Volumes*, 41(2):11744–11749, 2008.

[61] DS Long, SD DeGloria, and JM Galbraith. Use of the global positioning system in soil survey. *Journal of soil and water conservation*, 46(4):293–297, 1991.

[62] Antonio Barrientos, Julian Colorado, Jaime del Cerro, Alexander Martinez, Claudio Rossi, David Sanz, and João Valente. Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *Journal of Field Robotics*, 28(5):667–689, 2011.

[63] Camille CD Lelong, Philippe Burger, Guillaume Jubelin, Bruno Roux, Sylvain Labbé, and Frédéric Baret. Assessment of unmanned aerial vehicles imagery for quantitative monitoring of wheat crop in small plots. *Sensors*, 8(5):3557–3585, 2008.

[64] DS Long, SD DeGloria, and JM Galbraith. Use of the global positioning system in soil survey. *Journal of soil and water conservation*, 46(4):293–297, 1991.

[65] James F Roberts, Jean-Christophe Zufferey, and Dario Floreano. Energy management for indoor hovering robots. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1242–1247. IEEE, 2008.

[66] Wei-pang Chin and Simeon Ntafos. Optimum watchman routes. *Information Processing Letters*, 28(1):39–44, 1988.

[67] Xuehou Tan. Fast computation of shortest watchman routes in simple polygons. *Information Processing Letters*, 77(1):27–33, 2001.

[68] Moshe Dror, Alon Efrat, Anna Lubiw, and Joseph SB Mitchell. Touring a sequence of polygons. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 473–482. ACM, 2003.

[69] Jan Faigl. Approximate solution of the multiple watchman routes problem with restricted visibility range. *IEEE transactions on neural networks*, 21(10):1668–1679, 2010.

[70] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[71] Howie Choset. Coverage for robotics–a survey of recent results. *Annals of mathematics and artificial intelligence*, 31(1):113–126, 2001.

[72] Marina Torres, David A Pelta, José L Verdegay, and Juan C Torres. Coverage path planning with unmanned aerial vehicles for 3d terrain reconstruction. *Expert Systems with Applications*, 55:441–451, 2016.

[73] Paulo A Jimenez, Bijan Shirinzadeh, Ann Nicholson, and Gursel Alici. Optimal area covering using genetic algorithms. In *Advanced intelligent mechatronics, 2007 IEEE/ASME international conference on*, pages 1–5. IEEE, 2007.

[74] Young-Ho Choi, Tae-Kyeong Lee, Sang-Hoon Baek, and Se-Young Oh. Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 5788–5793. IEEE, 2009.

[75] Vatana An and Zhihua Qu. A practical approach to coverage control for multiple mobile robots with a circular sensing range. In *Robotic and Sensors Environments (ROSE), 2013 IEEE International Symposium on*, pages 112–117. IEEE, 2013.

[76] Heinz Mühlenbein. Parallel genetic algorithms, population genetics and combinatorial optimization. In *Workshop on Parallel Processing: Logic, Organization, and Technology*, pages 398–406. Springer, 1989.

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

[78] Trevor Davies and Amor Jnifene. Multiple waypoint path planning for a mobile robot using genetic algorithms. In *Computational Intelligence for Measurement Systems and Applications, Proceedings of 2006 IEEE International Conference on*, pages 21–26. IEEE, 2006.

[79] John Ware and Nicholas Roy. An analysis of wind field estimation and exploitation for quadrotor flight in the urban canopy layer. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1507–1514. IEEE, 2016.

[80] Yu Liu, Xiaoyong Lin, and Shiqiang Zhu. Combined coverage path planning for autonomous cleaning robots in unstructured environments. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pages 8271–8276. IEEE, 2008.

[81] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[82] Hans J Bremermann. Optimization through evolution and recombination. *Self-organizing systems*, 93:106, 1962.

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

[84] George EP Box. Evolutionary operation: A method for increasing industrial productivity. *Applied statistics*, pages 81–101, 1957.

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

[86] Jarmo T Alander. An indexed bibliography of genetic algorithms papers of 1996 (in proceedings). Technical report, 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, 1994.

[87] Ingo Rechenberg. Cybernetic solution path of an experimental problem. 1965.

[88] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.

[89] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.

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

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

[92] Simon Ronald. Robust encodings in genetic algorithms: A survey of encoding issues. In *Evolutionary Computation, 1997., IEEE International Conference on*, pages 43–48. IEEE, 1997.

[93] Godfrey A Walters and David K Smith. Evolutionary design algorithm for optimal layout of tree networks. *Engineering Optimization*, 24(4):261–281, 1995.

[94] Suhail Owais, Vaclav Snasel, Pavel Kromer, and Ajith Abraham. Survey: using genetic algorithm approach in intrusion detection systems techniques. In *Computer Information Systems and Industrial Management Applications, 2008. CISIM'08. 7th*, pages 300–307. IEEE, 2008.

[95] Alden H Wright et al. Genetic algorithms for real parameter optimization. *Foundations of genetic algorithms*, 1:205–218, 1991.

[96] Brad L Miller, David E Goldberg, et al. Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, 9(3):193–212, 1995.

[97] David Edward Goldberg. *Optimal initial population size for binary-coded genetic algorithms*. Clearinghouse for Genetic Algorithms, Department of Engineering Mechanics, University of Alabama, 1985.

[98] Yacine Morsly, Nabil Aouf, Mohand Said Djouadi, and Mark Richardson. Particle swarm optimization inspired probability algorithm for optimal camera network placement. *IEEE Sensors Journal*, 12(5):1402–1412, 2012.

[99] Agoston E Eiben, James E Smith, et al. *Introduction to evolutionary computing*, volume 53. Springer, 2003.

[100] HAJ-RACHID Mais, Christelle Bloch, Wahiba Ramdane-Cherif, and Pascal Chatonnay. Différentes opérateurs évolutionnaires de permutation: sélections, croisements et mutations. *LABORATOIRE D'INFORMATIQUE DE L'UNIVERSITE DE FRANCHE-COMTE*, 2010.

[101] Olivier François and Christian Lavergne. Design of evolutionary algorithms-a statistical perspective. *IEEE Transactions on Evolutionary Computation*, 5(2):129–148, 2001.

[102] Jaroslaw Arabas, Zbigniew Michalewicz, and Jan Mulawka. Gavaps-a genetic algorithm with varying population size. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 73–78. IEEE, 1994.

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

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

[105] XH Shi, YC Liang, HP Lee, C Lu, and LM Wang. An improved ga and a novel pso-ga-based hybrid algorithm. *Information Processing Letters*, 93(5):255–261, 2005.

[106] Raphaël Cerf. An asymptotic theory for genetic algorithms. In *European Conference on Artificial Evolution*, pages 35–53. Springer, 1995.

[107] Hans-Paul Schwefel. 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, 1984.

[108] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *International Conference on Parallel Problem Solving From Nature*, pages 849–858. Springer, 2000.

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

[110] Dirk Thierens and David Goldberg. Elitist recombination: An integrated selection recombination ga. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 508–512. IEEE, 1994.

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

[112] Mandavilli Srinivas and Lalit M Patnaik. Genetic algorithms: A survey. *computer*, 27(6):17–26, 1994.

[113] Kuk-Hyun Han and Jong-Hwan Kim. Genetic quantum algorithm and its application to combinatorial optimization problem. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 2, pages 1354–1360. IEEE, 2000.

[114] Kuk-Hyun Han and Jong-Hwan Kim. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE transactions on evolutionary computation*, 6(6):580–593, 2002.

[115] Rahila H Sheikh, Mukesh M Raghuwanshi, and Anil N Jaiswal. Genetic algorithm based clustering: a survey. In *Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on*, pages 314–319. IEEE, 2008.

[116] Yuan-Kai Wang and Kuo-Chin Fan. Applying genetic algorithms on pattern recognition: An analysis and survey. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 2, pages 740–744. IEEE, 1996.

[117] Wing Ning. Strongly np-hard discrete gate-sizing problems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(8):1045–1051, 1994.

[118] Russell Eberhart and James Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43. IEEE, 1995.

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

[120] K Premalatha and AM Natarajan. Hybrid pso and ga for global maximization. *Int. J. Open Problems Compt. Math*, 2(4):597–608, 2009.

[121] SG Ponnambalam, H Jagannathan, M Kataria, and A Gadicherla. A tsp-ga multi-objective algorithm for flow-shop scheduling. *The International Journal of Advanced Manufacturing Technology*, 23(11-12):909–915, 2004.

[122] Noraini Mohd Razali, John Geraghty, et al. Genetic algorithm performance with different selection strategies in solving tsp. In *Proceedings of the world congress on engineering*, volume 2, pages 1134–1139. International Association of Engineers Hong Kong, 2011.

[123] Howie Choset and Philippe Pignon. Coverage path planning: The boustrophedon cellular decomposition. In *Field and service robotics*, pages 203–209. Springer, 1998.

List of Figures

List of Tables

List of Definitions

I

ANNEXES

A

ANNEXES

A.1/ ACRONYM LIST

UAV - Unmounted Aerial Vehicle
PTZ - Pan Tilt Zoom
AGP - Art Gallery Problem
NP-Hard - Non Polynomial Hard
NP-Complete - Non Polynomial Complete
WSN - Wireless Sensor Network
WSAN - Wireless Sensor and Acutor Network
PSO - Particles Swarm Optimization
VSN - Visual Sensor Network
BIP - Binary Integer Programming
EA - Evolutionary Algorithm
SA - Simulated Annealing
MAGA - Multi-Agent Genetic Algorithm
HEA - Hybrid Evolutionary Algorithms
PCB - Printed Circuit Board
PI-PSO - Probability Inspired binary PSO
CR-PSO - Craziness based PSO
CPP - Coverage Path Planing
WRP - Watchman Route Problem
TSP - Travelling Salesman Problem
PNWCC - Novel Previous-Next Waypoints Coverage Constraint
DNA - DeoxyriboNucleic Acid
MOP - Multi Objective Problem
SGA - Simple Genetic Algorithm
MOEA - Multi Objective Evolutionary Algorithms
MOEA/D - Multi Objective Evolutionary Algorithms decomposed
NSGA - Non-dominated Sorting GA
NSGA-II - Non-dominated Sorting GA II
BMPGA - Bi-objective Multi Populations GA
QGA - Quantum-inspired GA
DoF - Degree of Freedom
RS - Random Selections
DoE - Design of Experiments
GT - Ground Truth

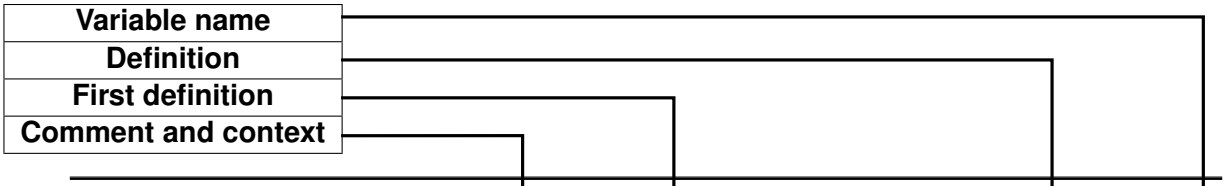
NW - Number of Waypoints
GAPSO - Genetic Algorithm Particles Swarm Optimization
FoV - Field of View

Computation to compute the η, υ. Where η, υ are the horizontal and vertical camera field of view.

$$\begin{aligned} \eta &= 2 \times \tan^{-1}(\frac{Sw}{(2 \times f)}) \\ \upsilon &= (\frac{Sw}{Sh}) \times \eta \end{aligned} \tag{A.1}$$

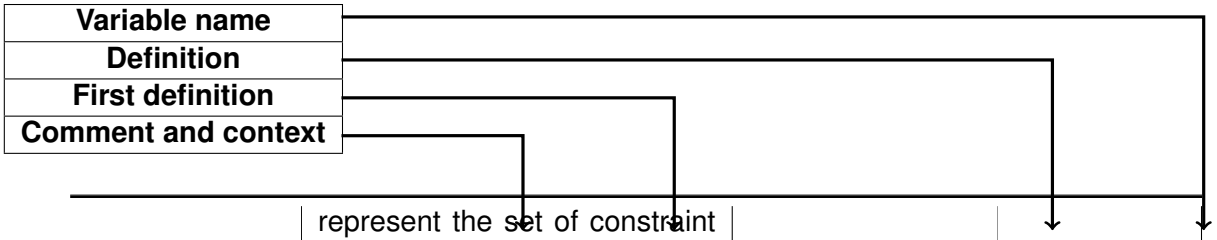
The Equation A.1 is to complete the Figure 4.8.

A.2/ EQUATION VARIABLE



k and k-coverage	Where k represent the number of cameras useful to cover some region of the area.	Section :2.1.2	All
P	Is a Polygon used to represent a art galery (room).	Sub-section :2.1.3.1	Only the Section 2.1.3
n	Number of vertice of P	Section :2.1.3.1	Only the Section 2.1.3
v_i	Is the i^{th} vertice of the polygon P	Section :2.1.3.1	Only the Section 2.1.3
x_i	Is the i^{th} guard (coordinate position in P)	Section :2.1.3.1	Only the Section 2.1.3
y	Is a point $y \in P$	Section :2.1.3.1	Only the Section 2.1.3
X	Is set with the minimum number of guards X useful to fully cover the polygons P	Section :2.1.3.1	Only the Section 2.1.3
g	Is the number of guard x need it to cover P to have a set $X = \{x_1 \dots x_i \dots x_g\}$	Section :2.1.3.1	Only the Section 2.1.3
G	Is an occupation grid of the area	Section :4.1	All
g_i	the i$^t h$ point of the grid G should be covered by a camera	Section :4.1	All
m	is equal to the number of points in the grid G	Section :4.1	All
Pc	is the list of the point g_i form the grid G which are covered	Section :4.1	All
p_i	is the weight of the point g_i on the grid G	Section :6.1.1	Only the Section 6.1.1
P	is the list of p_i which contain the weighting of the area	Section :4.1	Only the Section 4.1
U	the zones without interest to be covered noted with the set U. $U \in G$	Section :4.1	Only the Section 4.1 ???
(α, β, γ)	Three degrees of freedom for the camera orientation orientation: with the the rotation in pan, tilt, and roll angles	Section :4.2.1	All
f	the focal lens	Section :4.2.1	Only the Section 6.1.1
Obj_k	the k^{th} object in the scene.	Section :6.1.1	Only the Section 6.1.1
$f(...)$	The function is in charge to compute a camera projection	Section :6.1.1	All

Table A.1: Sum-up of the grid map.



E	represent the set of constraint (the set E is defined more in detail latter)	Section :2.1.2	All
V	represents a solution. V contain all individual positions and orientations of the set of cameras for a predefined focal length, sensor size and related map depending on the problem	Sub-section :4.2.2	Only the Section 4.2.2
n	the number of cameras in the network	Section :4.2.2	All
V_k	velocity of a particlule at the kth iteration	Section :5.2.1	Only the Section 5.2.1
P_i	best solution of a particle	Section :5.2.1	Only the Section 5.2.1
P_g	The best solution from the swarm	Section :5.2.1	Only the Section 5.2.1
ω	is the inertia of the swarm	Section :5.2.1	Only the Section 5.2.1
$b1$ and $b2$	is random value between 0 and ϕ_g for b_1 or ϕ_p for b_2	Section :5.2.1	Only the Section 5.2.1
α_i	is an angle of curve in the trajectory as in Figure 6.2	Section :6.1.2	All
!	the ith point of the grid G should be covered by a camera	Section :4.1	All
!	is equal to the number of points in the grid G	Section :4.1	All
Pc	is the list of the point g_i form the grid G which are covered	Section :4.1	All
A_{room}	area of the Room (length $\times$ width)	Section :6.1.1	Only the Section 6.1.1
A_{Wall}	area of the obstacle like wall (length $\times$ width)	Section :6.1.1	Only the Section 6.1.1
A_{Cam}	area cover by the camera in the maximum size of z	Section :6.1.1	Only the Section 6.1.1
NWayPoint	number of waypoints	Section :6.1.1	Only the Section 6.1.1
Threshold Rate	objective threshold rate	Section :6.1.1	Only the Section 6.1.1
S	one solution of waypoints set	Section :6.1.1	Only the Section 6.1.1
evalCost	cost function	Section :6.1.1	Only the Section 6.1.1

Table A.2: Sum-up of the grid map.

Document generated with LaTeX and:
the LaTeX style for PhD Thesis created by S. Galland — http://www.multiagent.fr/ThesisStyle
the tex-upmethodology package suite — http://www.arakhne.org/tex-upmethodology/

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