UNIVERSIDAD DEL NORTE
Department of Industrial Engineering
Barranquilla, Colombia

BAYESIAN SEARCH UNDER DYNAMIC DISASTER SCENARIOS

BÚSQUEDA BAYESIANA BAJO ESCENARIOS DE DESASTRE DINÁMICOS

by

José Daniel Betancourt Reyes

A thesis submitted in partial fulfillment of the requirements for the Degree of

Master of Science in Industrial Engineering

Advisor: Gina Galindo, Ph.D.

Juries:	Dr. Rajan Batta	State University of New York
	Dr. Yezid Donoso	Universidad de Los Andes
	MSc. Santiago Nieto	Universidad del Norte

Barranquilla, January of 2017

Dedicated to

God

Who gave me the opportunity to live this wonderful experience and the strength to not give up. He is simply the greatest of my life.

My lovely family

Whose support and encouragement were exceptional aids to me through this path. My grandparents Vicente and Gloria, my mom and her husband Julia and Luis, my uncles Juan and Jorge, my aunt Zaida and her amazing children Sara and Josue, are all extraordinary people that I will always carry in my heart.

My girlfriend Juliette

With whom I have been so fortunate of sharing this stage of my life. Her company along this journey has filled my life with thousands of fantastic experiences which I treasure in my heart.

Acknowledgments

First of all, I want to thank God for giving me all the required resources to complete this thesis. I will always recognize that all I achieve is primarily thanks to Him.

I feel so happy of been directed by Gina Galindo for the development of this research. Her advices, corrections and instructions were crucial for the satisfactory completion of this thesis. I recognize the valuable opportunity that she offered to me of working with her in two research projects where I learned a lot and earned economic resources to support my studies.

I specially recognize the support provided by my family. They have motivated and inspired me to give my best at each stage of my career.

I also want to acknowledge the unconditional support that my girlfriend provided to me. I have been fortunate of counting with her as a friend and also as a colleague able to understand all the crazy things that I spoke during the execution of this thesis.

Finally, I would like to thank a very special group of people that I have encountered on the road. The truth is that it is not a very big group of people, but their ability to serve others makes them so special humans.

Abstract

Search and Rescue (SAR) is a hard decision making context where there is available a limited amount of resources that should be strategically allocated over the search region in order to find missing people opportunely. In this thesis, we consider those SAR scenarios where the search region is being affected by some type of dynamic threat such as a wildfire or a hurricane. In spite of the large amount of SAR missions that consistently take place under these circumstances, and being Search Theory a research area dating back from more than a half century, to the best of our knowledge, this kind of search problem has not being considered in any previous research.

Here we propose a bi-objective mathematical optimization model and three solution methods for such purpose: (1) Epsilon-constraint; (2) Lexicographic; and (3) Ant Colony based heuristic. One of the objectives of our model pursues the allocation of resources in most risky zones. This objective attempts to find victims located at the closest regions to the threat, presenting a high risk of being reached by the disaster. In contrast, the second objective is oriented to allocate resources in regions where it is more likely to find the victim. Multi-objective decision making techniques where implemented in this thesis.

Furthermore, we implemented a receding horizon approach oriented to provide our planning methodology with the ability to adapt to disaster's behavior based on updated information gathered during the mission. A second set of experiments evidenced the validity of our search planning methodology supported on a receding horizon scheme.

The search planning methodology proposed here has the capability of distributing available resources over the most risky zones and the location where it is most expected to find the victim. Our methods are suitable for a fleet of heterogeneous agents, making it appropriate for application in real search contexts, where different kind of agents belonging to different emergency management and research agencies take part.

This thesis was unanimously nominated to be honored with a *Cum Laude* distinction by the members of the committee, based on the excellence of the results and products, the quality of the defense and the diverse opportunities of future research.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Search and Rescue (SAR) is an increasingly important operation in many military and humanitarian contexts, given its primary objective of saving lives. This operation often involves complex decision making scenarios where search planners need to allocate available resources strategically in order to increase the chances of finding the victim(s) opportunely. Unfortunately, SAR takes place under many different scenarios and there is not any generic search strategy that can be broadly applied to all scenarios. For example, the strategy implemented in a search operation underway for survivors of a plane crash over the ocean should be different from the one implemented if a person became lost on a wilderness area. In the former case, the possible location of the victims should be considered. A belief map for their position could be made based on crash coordinates and progressively updated according to the expected displacement of people due to sea motion. Then, a strategy could be to focus search efforts on the most likely zones at each moment. In the second case, there could be a total lack of information about victim's location. Under these circumstances, a maximum coverage-minimum time strategy could be a good choice.

There is a large number of studies focused on the construction of search plans under different scenarios. Some versions of the problem are constituted by combining levels of the following parameters: number of search units (one or multiple), SAR team configuration (homogeneous or heterogeneous), SAR units' sensing reliability (infallible or fallible), target state (stationary or mobile), among others. Furthermore, previous researches considered some typical constraints on SAR, such as those concerning to: searchers' motion (degrees of freedom, speed and acceleration), fuel available for the search team and maximum coverage of transmitters. Despite the wide variety of configurations of the search problem that have been studied, there exists a special SAR scenario which has not been tackled yet, at least to the best of our knowledge. To describe such a scenario, let us first define a non-instantaneous disaster; in this thesis, a non-instantaneous disaster refers to a natural disaster that spreads or progress through a region taking some hours, days or even weeks. Some examples of non-instantaneous disasters are volcanic eruptions, hurricanes and wildfires. In contrast, landslides are examples of instantaneous disasters while they occur suddenly and last for a few minutes or even seconds. The aftermaths of some type of disasters can be also considered a non-instantaneous disaster. For instance, consider the potential collapse of a damaged building few minutes after the occurrence of an earthquake. Such kind of threatening phenomenon will also be considered a non-instantaneous disaster in this thesis.

Having defined what a non-instantaneous disaster is, we can describe the scenario that we address in this thesis. Consider a search mission for a missing person, where the search area is being affected by a non-instantaneous disaster. The phenomenon is advancing through the region and has the potential to kill anyone in its way. In such a scenario, conventional search strategies are not sufficient as they do not consider the fact that there is a phenomenon progressively exhausting feasible search zones. In other words, they do not take into account the distribution of the risk over the region and its dynamic behavior.

In this thesis, we present what to the best of our knowledge is the first mathematical model for search planning in presence of a non-instantaneous disaster. Such a model is capable of coordinating a team of heterogeneous agents by defining the search path for each one of them and the number of explorations to be performed at each location of the search region. We equipped our model with a recursive Bayesian filter which updates the belief map after each exploration of an agent, allowing to decide how long to remain exploring at each location. According to our literature review, the mathematical model proposed in this thesis is one of the few path planning models for search missions capable of making such an update. As a result, our model builds reactive search sequences, adapted to expected changes in the belief map. Several previous path planning models found in the literature, only update the belief map at the end of each planning window.

In contrast to several models found in the literature, our model does not impose any limitation in time or space to the displacements executed by search agents. Instead of discretizing the time as is commonly done in previous studies, we designed an event-based model, which allows agents to move among any pair of cells at any time. This characteristic provides our model with a higher representative capacity, compared with models that discretize the time.

As expected, the computational requirements of solving our model result prohibitive for real size instances. In this respect, we designed and implemented a multi-objective Ant Colony based algorithm namely Pareto Multi-Agent Ant Colony Optimization (PMAACO), which is capable of effectively solving the problem under study. Our experiments show that our PMAACO is able to provide good quality solutions in a relatively short time. The applicability of PMAACO algorithm in real size instances was tested in a rolling horizon fashion that allows to keep an updated track of the disaster through the mission.

1.2 Organization

In this thesis, we introduce the Optimal Search Path with Effort Allocation Problem under Dynamic Disaster (OSPEAD). We further provide three solution methods for it; two of them are based on the exact solution of the mathematical formulation of the problem and the third one is an efficient heuristic method, able to deal with the trade off between computational requirements and solution quality. The steps on the development of this research and the corresponding results are exposed in the next chapters as follows: in Chapter 2 we extend the description of our research problem. In that chapter, we also justify the execution of this thesis and state our research objectives. In Chapter 3, we state the theoretical basis supporting our research. The chapter starts with a literature review of 83 articles published at different moments through the history of Search Theory (ST). Our survey not only provides a panorama of historical developments in ST, but it also introduces a novel taxonomy for studies in that research line, based on the identification of the main types of search problems and the typical variants of the problem setting. In Chapter 4 we introduce our mathematical formulation for the OSPEAD, which is in principle a Mixed Integer Nonlinear Program. The separability of our model allowed us to implement the Piecewise Linear Approximation method, leading us to a linear version of our model, corresponding to a Mixed Integer Linear Program. Accounting for the dynamic behavior of the disaster, we propose the solution of the OSPEAD under a receding horizon scheme working as follows: at each time window, a forecast of disaster future state (location and severity) is generated based on updated information collecting by sensing platforms and imagery collection devices. This information is then used as an input of our model, which prioritize explorations in most danger locations for the next time window. The whole search path for each agent results from the joint of individual search paths built at each window. In Chapter 5, we present three solution methods for the OSPEAD which are then tested and compared in Chapter 6. Finally, in Chapter 7 we offer our conclusions and suggest a set of promising future research lines.

Chapter 2

Problem Description

2.1 Problem statement

Imagine that a wildfire is in progress in a large outdoor area such as a natural reserve. Authorities evacuated the area, but a person was reported missing. A SAR mission should be performed soon because this person could be in risk of being harmed by the fire. SAR units are ready to launch the mission, but which search strategy should they apply in order to increase the chances of finding the missing person safe? If there is available some information about target's location, a strategy could include some smart use of this information. Based on it, the mission could start by investing efforts on exploring over the most likely zones where the person could be, and then progressively move to less likely zones until the person is found. If there is not any clue about the location of the sought person, the strategy could be focused on achieving the maximum possible coverage within the minimum possible time. However, an overriding element that should be taken into account is that the fire is advancing and each zone that becomes exhausted, represents a lost opportunity of finding the missing person alive. Accordingly, this thesis aims to answer the following question: which could be an efficient search strategy for a single missed person, on an area hit by a non-instantaneous disaster that progressively consumes parts of the search region? To answer this question, we applied mathematical modeling and algorithm concepts on the construction of a methodology for conducting a search mission in presence of an ongoing non-instantaneous disaster. This methodology will increase the effectiveness of a SAR mission under stated circumstances. Our methods are not limited to a given type of disaster, but they concern about any non-instantaneous natural phenomenon that could progressively consume the search region.

The dynamic behavior of the threatening phenomenon makes it necessary to systematically adapt the mission to phenomenon's evolution. The lack of an adaptation scheme may conduct to unfeasible or poor solutions. To illustrate this idea, imagine that the emergency response manager is going to apply a non-adaptive scheme in a search mission in presence of a hurricane. In such a scheme, a set of forecasts for hurricane's future state is generated at the beginning of the mission and then, those forecasts are used as inputs for a single run of the search planning model. No future updates of the forecast are done. The weakness of the non-adaptive schemes is that they do not consider the natural uncertainty associated with non-instantaneous disaster forecasts. This issue may introduce the following two main types of imprecision in the search plan: (1) the scheduling of explorations over certain locations at moments when such locations will already have been exhausted by the phenomenon; and inversely (2) the avoidance of scheduling explorations at certain locations, at moments when forecast tags them as exhausted, given that they will actually remain feasible to explore at those moments.

To mitigate this issue, we propose the adoption of a receding horizon scheme where the search planning model is solved iteratively, providing the search plan for the next time window each time that it is solved. At each time window, an updated forecast for disaster's behavior until the next horizon is generated and taken as input for the next run of the search planning model. We are aware that such a methodology do not completely remove the two types of distortion in the search plan, mentioned in the previous paragraph. We will be forced to deal with those issues whenever the forecast is subject to uncertainty. Fortunately, our receding horizon scheme reduces such uncertainty by using shorter forecasting horizons than a non-adaptive scheme and keeping updated forecasts through the whole mission.

Recent execution of applied research projects as Deployable SAR Integrated Chain with Unmanned Systems (Chrobocinski, Makri, Zotos, Stergiopoulos, & Bogdos, 2012) and Modeling Crisis Management for Improved Action and Preparedness (Daou et al., 2014), suggests that it is time to start merging decision support models and methodologies belonging to Search Theory with the rising technology of robotics automation. Our methodology is not limited to a specific type of agent, nonetheless, we consider interesting its application making use of automated robotic agents, particularly unmanned aerial vehicles (UAVs). The advantages that robotic agents offer in comparison with humans include:

- They do not require the exposition of human agents to hazards present in the search region during the exploration,
- They facilitate the coordination within the search team,
- They allow a more precise execution of the search plan as it is designed, given the possibility of standardizing their kinematics and detection profile.

By the other side, UAVs offer the possibility of making large displacements in a short time, allowing the search team to rapidly cover the search region. Additionally, an unmanned quadrotor results much cheaper than a manned aerial vehicle such as a helicopter or a plane, which means that the usage of UAVs permits to include a greater number of search agents to take part on the mission in a lower cost than if it was performed by means of manned aerial vehicles. The fulfillment of this assumption will contribute to make it feasible to implement our protocol in real-world missions; however, once again we want to make it clear that our methods are not limited to any specific type of agent.

Real search mission often involve heterogeneous search teams. Heterogeneity is typically originated in the adoption of different types of agents, the uneven aging of homogeneous agents and the execution of a cooperative mission where different emergency management agencies or research groups take part. Seeking for the applicability of our methodology in real scenarios, we considered agents' travel times, exploration times and reliability as sources of heterogeneity within the search team.

2.2 Justification

Human lives are at stake during SAR operations. The search plan will have an impact on the probability of finding the missed person on time. In many practical cases, a wrong decision could imply a lost life. Unfortunately, this decision is not easy, since there could be millions of possible search plans suitable for the same mission. Just consider the number of possible paths that a team of SAR units could use to explore a certain area or the infinite combination of exploration times that could be assigned to each zone of the search region. Furthermore, the presence of a non-instantaneous disaster on the search area considerably hampers the decision to be taken. As stated before, there is not a decision making model, suitable for the conditions of our research problem. This lack of supporting tools is a weakness of current SAR protocols which reduces the chances of saving lives during SAR missions. Through the execution of this thesis, we developed a solution for this issue. The resulting methodology might contribute to reduce the risk of losing human lives during SAR missions under dynamic disaster scenarios.

2.3 Objectives

2.3.1 General objective

To design a protocol for conducting a search mission in presence of a non-instantaneous disaster, where such a disaster is progressively exhausting parts of the search region.

2.3.2 Specific objectives

- To build a mathematical model that provides the search paths for a team of heterogeneous agents considering the forecast of disaster's behavior.
- To select a solution method for our mathematical model which balances solution quality and processing time.

Chapter 3

Frame of reference

In this section we provide a literature review on Search Theory, which is the research area which closely matches to our research problem. Such review allowed us to identify research gaps present in current literature, which are fulfilled though the execution of this thesis.

3.1 Search Theory: A Taxonomic Literature Review

We conducted a review in Search theory (ST), which is the discipline that studies how to search for an entity in order to increase the chances to find it using limited resources. Search theory began with the researches performed by Dr. Koopman, B. O. and his team, during World War II, as a part of Antisubmarine Warfare Operations Research Group (ASWORG)(B. O. Koopman, 1946). This discipline grew stronger through years and nowadays it is an important and well-known field in Operations Research. One of the main applications of search theory is in SAR[1], where it concerns about the management of limited resources to find harmed or lost people. Our research problem fits in this research area since we are interested in the construction of search methodology for cases of missing people. Nonetheless, as it was stated in previous sections, this review is not exclusively focused on SAR scenarios because there are studies related to other search contexts, belonging to ST, that can be adapted to our purposes. The literature review presented here is identical in content to a paper co-authored by José Betancourt and Gina Galindo, which has been submitted to the academic journal Computers & Operations Research, and is currently under review for publication.

3.1.1 Search methodology and scope of the study

In this section we present the search methodology and the boundaries of our review. The following databases were used for our search: ScienceDirect, IEEE Xplore, ProQuest, SpringerLink, Wiley Online Library, Taylor & Francis, INFORMS PubsOnline, JSTOR, SPIE Digital Library and Google Scholar. The latter directed us to articles on the following academic websites: ResearchGate, Dudley Knox Library - Naval Postgraduate School, Cornell University Library, American Institute of Aeronautics and Astronautics,

[1] Search and Rescue

Scientific Research Publishing, MDPI AG, ISRN Mathematical Analysis, The Operations Research Society of Japan (ORSJ) and SIAM Journal on Applied Mathematics. Additionally, results from Google Scholar led us to the academic websites of two researchers, namely, Dr. R. Batta (Personal website at UB - University at Buffalo), and Dr. G. Di Caro (Personal website at IDSIA - Istituto Dalle Molle di Studi sull'Intelligenza Artificiale). We used *Search Theory*, *Search and rescue*, *Bayesian Search*, *Optimal Search path* and *Search effort allocation* as keywords in the aforementioned databases.

The scope of our survey limits to articles in Operations Research that clearly situate their problem on the ST field without specifying a particular time window. We did not consider studies that did not present any numerical or analytic approach to solve a search problem.

After screening the first set of articles, we performed backward and forward reference search. This procedure lead us to a second sample of items. We repeated the process multiple times, ensuring that all studies in the sample were meeting our acceptance criteria. For this purpose, we explored the abstract and the methodology section (if available) of each new item looking for the presence of numerical or analytic methodologies to solve search problems. The resulting collection of items constitutes by no means an exhaustive bibliography on ST. Nonetheless, we find our sample to be sufficiently diverse in year of publication, problem setting and solution approach, to make it suitable for our purposes of characterizing main ST types of problems and their variants.

As a result of our search process, considering the boundaries stated above, we incorporated a total of 83 items to our survey, including theses, book chapters, complete papers and conference papers. Major article producers identified by us were European Journal of Operational Research (7 items), Operations Research (5 items), Computers & Operations Research (4 items) and Naval Research Logistics (4 items). As expected, those four journals are part of the top in Operations Research and Management Sciences. The remaining journals accounted for only one article in our sample. As it is evident, there is not a clear preference for publishers in ST literature yet. This behavior is partially explained by the fact that only the 55.4% of our sample correspond to journal papers. The remaining 45% was distributed among conference papers (41.0%), technical reports (1.2%), book chapters (1.2%) and theses (1.2%).

3.1.2 Previous bibliographies and surveys

We are aware of five previous bibliographic reviews in the field of ST, these are Enslow (1966), Dobbie (1968), Stone (1989), Benkoski, Monticino, and Weisinger (1991) and Frost and Stone (2001). This group of articles presented the following characteristics:

1. Only two of the five studies presented conclusions and identified research gaps. However, only one of those two studies proposed future research lines. Unfortunately, that study was specifically concerned about the state of ST research on the US navy and their suggested research lines may not be applicable to ST research in general.

2. One of the studies is limited to provide the abstract for a list of 75 articles in ST and Reconnaissance Theory.

3. The classification scheme in tree of the studies appears to be oriented to the natural grouping of the publications rather than the natural decomposition of the ST matter. A fourth article classifies the literature based on ST main eras. The fifth article implements a classification scheme that relates research in ST with the definition of search planning doctrine and the development of computer based search planning decision support tools.

4. The four articles where published within the period 1966 to 2001, inclusive.

It is evident that previous surveys in ST have adopted a dissimilar perspective in the exploration of the literature. Our study is not an exception. In this review, we identified two typical purposes followed by search studies: (1) Path planning; and (2) Effort allocation. Furthermore, we noticed that ST literature was decomposed in studies solving search problems under planning and control schemes. The main difference between both approaches is that a single run of a planning model provides the decision for multiple actions of the search team, while a control algorithm guides the team step by step thorough the mission. The intersection between the purposes of the studies and the solution schemes lead us to the decomposition of ST literature in four types problems: (1) Search Effort Allocation Problem (SEA); (2) Optimal Search Path with Effort Allocation Problem (OSPEA); (3) Optimal Search Path Problem (OSP); and (4) Probabilistic Search Control Problem (PSC). Table 3.1 illustrates the characteristics of each type of problem. A formal definition of each type of search problem and a discussion on the developments for each one of them is presented in Section 3.1.4.

Table 3.1: Main types of search problems

Problem	**Objective**		**Sol. scheme**	
	Path planning	Effort allocation	Planning	Control
SEA		x	x	
OSPEA	x	x	x	
OSP	x		x	
PSC	x	x		x

Additionally, we identified a set of factors that can be adopted as variants of any of the four types of problems introduced above, some of them are: agent, target and scenario features such as the number of agents (one or multiple), the constitution of the search team (homogeneous or heterogeneous), the target state (stationary or moving), the time and space modeling (discrete or continuous) and others. We refer to this set of elements as the problem setting.

The identification of studies according on the type of problem and problem setting, allowed us to constitute a novel taxonomic decomposition ST literature. Our classification scheme allows researchers and practitioners to identify studies of interest for their particular purposes as part of a benchmark procedure. The classification by type of problem helps to discard the inspection of studies that present a different purpose (path planning/effort allocation) or solution approach (planning/control) than the one that the researcher is facing. Furthermore, the inspection of the problem setting of those articles will aid the identification of the studies that closely match to the characteristics of the problem under consideration and may also evidence unexplored problem settings.

3.1.3 Key SAR concepts

Before delving into the exploration of literature, this section provides the fundamental notions involved on a typical search mission. The concepts addressed here provide the basis for the models and methodologies found in both, theory and practice in ST. Consequently, it is highly recommended to gain full comprehension of these concepts before exploring any methodology related to ST. Fortunately, the majority of those concepts are very intuitive and can be easily understood by association with daily searches that we all make for misplaced objects such as keys or pens. The following subsections describe the tasks required for the construction of an optimal search plan. This scheme was structured based on Frost (1998) and Abi-Zeid and Frost (2005).

Task 1: Delimitation of the *possibility area*

Every search mission starts with the definition of the region where the mission will be conducted. Such a region is known as ***possibility area*** and can be defined as the smallest region that contains all possible object locations. The delimitation of the possibility area strongly relies on the exploitation of available information about the missing object and the environmental conditions. The ***last known position*** (LKP) of the object, its intended route (if available) and its physical features, combined with the topographic description of the area, its vegetation and weather, are all sources of information often implemented by search planners in the definition of the possibility area. In this thesis we will also refer to the possibility area as search region.

Task 2: Construction of the probability map

Once the possibility area has been delimited, it is discretized in a finite collection of sub-regions often referred as ***cells***. Such a discretization may reflect natural partitioning of the region (e.g. by rivers or buildings) or virtual sectioning defined by SP to aid analytic methods and assign specific tasks to ***agents*** through the mission (Chung, Kress, & Royset, 2009). Those sub-regions are typically represented as squared or hexagonal cells on ST literature. However, such cellular decomposition is often adopted for clarity in illustration and the majority of models are independent on the particular decomposition frame implemented by the SP.

Based on the same information that leaded to the definition of the possibility area, the SP subjectively assigns a probability to each cell, indicating the expectancy of finding the target there. The probability assigned to each cell is variously called ***probability of containment*** (POC), while the probability distribution allocated over the whole possibility area is referred as ***probability of containment map*** or ***probability map*** (POC map). It is always desired to be sure that the object is present in the possibility area (i.e. that the sum of the POC over the whole search region is one), however, this condition is not always true, due to limitations in time, fuel and other types of limited resources. This fact will be detailed later in this section.

As stated by Dr. L. Stone, a pioneer in ST research, "devising a probability distribution for target's location is an art rather than a science" (Stone, 1977). Fortunately, the incidence of the POC map on the effectiveness of a search plan is widely recognized in the practice and there are available multiple programs for support in its construction.

Task 3: Search team constitution

In order to build effective search plans, the SP must explicitly consider the particular characteristics of different types of agents. Indeed, it must also consider differences between agents of the same type. Humans, dogs, trucks, planes, helicopters and ships, constitute the traditional spectrum of search agents, all of them equipped with diverse types of ***sensing devices*** and ***recognition mechanisms***. Unmanned Aerial Vehicles (UAVs), Ground Vehicles (UGVs), Surface vehicles (USVs) and Underwater Vehicles (UUVs) have recently joined the list in the last decade.

Once again, available information about the object and the environment helps the SP to conform a convenient search team for the particular characteristics of the case in progress. For its selection, each search agent competes with the others in the following set of specifications:

Kinematic properties: maximum speed, acceleration and deceleration are all features belonging to the kinematic profile of an agent. All of them will determine the times required for the agent to move over the search region while performing the search plan. The degrees of freedom of the agent are also considered in this category. A very fast aircraft with limited capability of turning around presents a very different kinematic profile to the one corresponding to a human agent, slower but capable of turning around at any moment if required.

Autonomy: this second specification is strongly related to the concept of ***search effort***, which refers to any limited resource necessary for the operation of a given agent. Energy and fuel are both examples of search effort. With this in mind, the autonomy of an agent can be defined as its expected operative time provided that it has been fully recharged/refueled.

Skills and capabilities: this describes the expected performance of each agent at different sub-regions of the possibility area. For that reason, it is useful to discretize the region in such a way that the conditions at different points within each cell remain approximately homogeneous. To exemplify this, consider the search region composed by rocky, woody and open areas. Figure 3.1 illustrates such a scenario.

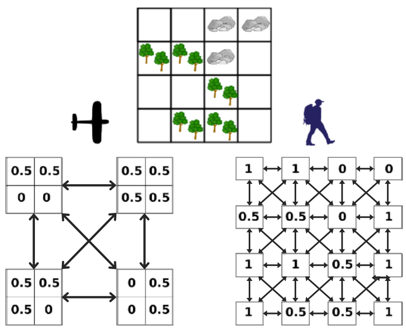


Figure 3.1: Capabilities based on environmental features
Retrieved from: Flushing et al. (2013)

It is expected to find different levels of performance coming from different types of agents at cells with different characteristics. For example, in Figure 3.1, aircrafts will not be able to effectively detect objects hidden in densely vegetated areas but will have the possibility of searching over rocky areas; in contrast, human searchers may not be able to search over rocky areas but will have the possibility to enter on the wooded areas and explore with a relatively high level of confidence. The strategic selection of the agents that will take part on the mission will be determinant on its success and therefore it is a critical step on search planning.

Reliability: this last specification is closely related to the previous one. ***Reliability*** of an agent is an indicator of its expected level of performance when exploring a region with certain conditions. Alternatively, it can be seen as a measure of how well an area was explored by a given agent (Frost, 1998). Agent reliability is commonly associated with a scalar between 0 and 1 representing the probability of detecting the object given that it is in the area searched. This feature accounts for the fact that real agents have rarely perfect detection and recognition capabilities.

Table 3.2: Sensor visibility coefficients

Environment	**Sensor**					
	1	**2**	**3**	**4**	**5**	**6**
Forest	0.4	0.5	0.6	0.8	0.5	0.1
Water	0.9	0.1	0.1	0.1	0.3	0.5
Town	0.3	0.1	0.4	0.6	0.5	0.2
Rugged land	0.2	0.7	0.8	0.2	0.4	0.6
Very rugged land	0.1	0.6	0.7	0.1	0.3	0.5
Flat land	0.8	0.9	0.1	0.7	0.6	0.2

Retrieved from: Simonin et al. (2009)

Table 3.2 illustrates an example where six agents were modeled with heterogeneous sensing reliabilities. As in the example illustrated in Figure 3.1 of heterogeneous agent capabilities, agents possess environment-dependent sensing reliabilties.

The longer an agent remains exploring at a the area where the object is hidden, the higher the probability that the agent will detect the object. This concept is captured by a quantity known in ST as ***probability of detection*** (POD). The POD represents the probability of detecting the object given that it is in the area searched, as a function of the effort that the agent expends exploring there. Hereinafter we will denote the POD mathematically as:

$$POD_i^k = POD(\alpha_k, z_i^k) \tag{3.1}$$

being α_k the reliability coefficient of agent k and z_i^k the amount of effort invested by agent k, exploring at cell i.

Task 4: Allocation of available effort

Whenever it is available a POC map and the reliability profile corresponding to each agent, the SP is able to estimate the expected benefit of performing an exploration on a given cell. Such benefit is known in ST as ***probability of success*** (POS) and can be defined as the joint probability of: (1) the missing object being in the area scanned by the agent; and (2) the agent effectively identifying the object hidden in scanned area. As a joint probability of two independent events, POS can be computed by the product between the probability of the two events as follows:

$$POS_i = POC_i \ x \ POD_i^k \tag{3.2}$$

where i is the index of the scanned cell and k is the index of the agent.

The construction of an optimal search plan involves the strategic distribution of the effort available for each agent f^k in such a way that one or multiple criteria are maximized or minimized. Every time that POC map and POD profiles are available, the maximization of ***cumulative probability of success*** is a natural and common criteria to be maximized. The resulting optimization problem is here referred as the ***general search problem*** (GSP) and can be formulated as follows:

$$\begin{aligned} \text{Max} \quad & POS = \sum_i \sum_k POC_i \ x \ POD_i^k \\ \text{subject to} \quad & \sum_i z_i^k \leq f^k, \qquad \forall k \end{aligned} \tag{GSP}$$

The GSP is the basis of any search problem existent in ST literature.

Task 5: Updating of probability map

The last but not least important task required for the construction of an optimal search plan is the updating of the POC map after each exploration performed by an agent. Given that a certain agent k performs a single unsuccessful scan on a given region, its POC should immediately be updated to represent that it is now less expectancy of finding the target there. The updated POC can be expressed as the probability that the agent is on cell i given that agent k said that it is not there, and corresponds to a conditional probability. The computation of the updated POC can be performed as follows:

$$POC_i^1 = \frac{POC_i^0 \ x \ (1 - POD_i^k)}{POC_i^0 \ x \ (1 - POD_i^k) + (1 - POC_i^0) \ x \ (1 - \delta_k)} \tag{3.3}$$

where δ_k represents the probability that the agent will erroneously detect the object given that the it is not present in the area and superscripts 0 and 1 in POC indicate if the value correspond to the POC before or after the exploration, respectively.

In cases where agents are virtually false alarm free, δ_k becomes 0 and Equation 3.3 is simplified to:

$$POC_i^1 = \frac{POC_i^0 \ x \ (1 - POD_i^k)}{1 - POC_i^0 \ x \ POD_i^k} \tag{3.4}$$

Case of moving target

In the case of a moving target, an additional step is often required. Here we explain the computation for the specific case of a Markovian neutral target. We mean by neutral that the target is neither evasive, nor non-evasive, its decisions are just independent of searcher's actions. Let M be a stochastic matrix describing a Markov chain over the discretized search region R, where M_{ij} denotes the probability that the target will move from cell i to cell j on a single time step. Each time that an agent performs an exploration, the POC map can be updated by the following two steps:

1. Implement Equation 3.3 to find the updated POC of each cell $\widehat{POC}_j^1$ according to the searcher's actions:

$$\widehat{POC}_j^1 = \frac{POC_j^0 \ x \ (1 - POD_i^k)}{POC_j^0 \ x \ (1 - POD_i^k) + (1 - POC_j^0) \ x \ (1 - \delta_k)} \tag{3.5}$$

2. Update POC according to the motion model of the target by means of Equation 3.6:

$$POC_j^1 = \sum_i M_{ij} \widehat{POC}_i^1 \tag{3.6}$$

This procedure is suitable for cases where the target is not able to leave the search region and also for case when it is.

3.1.4 Representative search problems

Through the story, the GSP have been adapted to a large variety of scenarios. In this section, we classify derived versions of the GSP in four main sub-problems, based on the characteristics of their solutions. The first type of problem namely SEA, consists on determining the best allocation of available search effort. The second type of problem, namely OSPEA is concerned not only about effort allocation but also about the routing of search agents. The third type of problem, namely OSP is only focused on the routing of search agents, and the search effort allocation is managed implicitly by allowing multiple visits to the same location. Those three types of models are solved under a planning scheme, where a single run of the decision support model accounts for multiple decisions of each agent. By its side, the fourth type of problem, namely PSC has the same objective of the OSPEA problem, but it is solved under a control scheme, where an algorithm guides search entities step by step through the mission.

The four types of problems defined here correspond to an empirical classification proposed by us. Other classifications found in the literature may be perpendicular to our classification, meaning that a category of problem considered in other study may be associated with problems in any of the four categories defined by us. To illustrate us, consider the *Stationary target - discrete time and space* problem defined as a category in Benkoski et al. (1991). In our classification scheme, we may find studies considering stationary targets, modeled in a discrete time and space environment, in any of the four types of problem.

3.1.4.1 Search Effort Allocation (SEA)

This first version of the search problem consists on distributing the available effort of each agent f_k at the most convenient areas of the search region R. This problem is only concerned about the amount of effort that should be allocated at each point of R, irrespective of the path that must be followed by each agent to perform such allocation. In this sense, the SEA can be associated with a *resource allocation problem*.

The SEA problem has been tackled in both, the continuous and the discrete version. The continuous version was first introduced by Dr. Koopman in B. Koopman (1956a, 1956b); B. O. Koopman (1957). In those three articles, he presented a pure mathematical methodology for the solution of the continuous SEA problem, considering a single target. Despite the revolutionary nature of this work, it was first limited to the special case where the POC map comes from a bivariate normal probability distribution. Dr. Koopman later extended his methods to a more general set of probability distributions. The optimization model proposed by Dr. Koopman is presented in Formulation C-SEA with minor adaptions to our notation.

$$
\begin{aligned}
&\text{Max } \ POS(z) = \int_x \int_y POC(x,y)\ POD(x,y,z(x,y))\ dx\ dy \\
&\text{subject to} \\
&\int_x \int_y z(x,y) = F \\
&z(x,y) \geq 0, \qquad \forall x, y
\end{aligned}
\tag{C-SEA}
$$

where $z(x,y)$ is a search density function that indicates the amount of effort to be allocated at each point of the continuous plane constituted by R.

In the discrete version, the region R is divided into a finite collection of cells as described in Section 3.1.3. This version of the problem was later introduced by Charnes and Cooper (1958). Under this scenario, the SEA problem becomes to find the amount of effort that should be assigned to each cell so that cumulative POS is maximized, subject to a limited amount of search effort. Resulting formulation looks exactly like the GSP model illustrated in Formulation GSP.

The SEA problem was then extended by Dr. Stone, L., to the case of a single moving target (Stone, 1979). In this work, the author provided necessary and sufficient conditions for an optimal solution of both, the continuous and the discrete version of the problem. By these years, the improvement of computers began to gain presence in the Operations Research (OR) context and thus in ST field. In Brown (1980) the authors proposed an algorithm capable to build optimal solutions to the discrete version of the SEA problem proposed by Charnes and Cooper (1958). By their side, Hohzaki and Iida (2001) tackled a special case of the discrete SEA problem where the target has multiple options of predefined paths. Each path has a probability of being selected by the target, and the searcher knows both, the constitution of each path, and the probability of being selected by the target. The authors formulated the problem as a concave maximization program and proposed two algorithms for its solution based on Kuhn-Tucker necessary and sufficient conditions for optimality. Their methods was noticed to be more efficient than numerical methods developed in previous works for moving targets.

Hohzaki (2006) also considered a SEA problem where a moving target is able to chose its path from a predefined set of them. In this case, the author implemented a game theoretic approach to deduce equilibrium points for both, the continuous and the discrete version of the problem. The study by Hohzaki (2006) differs from previous works, not only in the solution approach occupied there, but also in the assumption that the target is evasive. The author proposed a theorem for the existence of equilibrium points in both versions of the SEA. Nonetheless, he reported the intractability of the continuous version solved by his method.

An important extension of SEA models was done by Iida, Hohzaki, and Sakamoto (2002) who relaxed the assumption of *local effectiveness of searching effort*, consistently imposed in previously published works. This assumption states that the search effort located on a certain point of R is only effective at this point and does not have any effect on a target located at any neighbor point. In many practical cases, this assumption seems to be unrealistic, since real agents typically possess a sensing range/field of view and targets within this range/field are detectable. The assumption of local effectiveness of searching effort is only valid in scenarios where the sensing range/field of view is relatively tiny with respect to the size of the search region. This could be the case of a vessel equipped with a sonar, searching a submarine on a large region in the ocean. This example provides some insights in the possible rationales that could lead ST pathfinders to adopt this assumption.

By their side, Dambreville and Le Cadre (2004) extended Dr. Koopman's formulation to account for multiple modes of detection. They tackled a generalization of the continuous SEA problem where each agent possess particular effort allocation function and reliability. As will be discussed later, heterogeneity within the search team is a very important feature in nowadays practical scenarios. The work in Dambreville and Le Cadre (2004) is one of the pioneers accounting for this fact.

In a recent study, Beltagy and Abd Allah El-Hadidy (2013) addressed a particular case of the SEA problem that consists on determining the amount of search effort to be allocated at each point of a predefined search path. They considered the special case where the agent follows a parabolic spiral curve around its initial position. The authors developed a theorem that allows them to determine the expected time to find the target, which is moving randomly over a plane.

All reviewed articles tackling the continuous SEA problem assumed that search effort in infinitely divisible. It means that search effort can be allocated as finely as desired at any point of R. This assumption represent an obstacle for the application of those models in real search missions. The limitation of this assumption comes from the fact that the majority of search agents explores the region moving through a smooth path with limited control of the effort that they spend at each point it. Dogs, humans, aircrafts and almost any type of search agent has very limited control of the amount of energy, fuel or battery that they spend along their search path. This fact has motivated the preference of the discrete version of the problem for realistic applications.

In Abi-Zeid and Frost (2005), the authors present SARPlan; a realistic decision support system intended to help search planners to allocate available effort in SAR missions. SARPlan is able to receive raster files with information about the search region such as the vegetation and topography for the construction of a search effort allocation plan. The model implemented in SARPlan considers specific features of the search agents such as their maximum speed, altitude and specifications of the sensing devices to be employed in the mission. The size of the search region, the POC map and the total effort available are basic parameters for SARPlan to operate. Internally, SARPlan runs an optimization model whose solution is an effort allocation plan over a discrete set of bounded regions. The system also allows the user to specify unfeasible search zones such as high mountains or lakes if it is known that the victim might not be there.

3.1.4.2 Optimal Search Path with Effort Allocation (OSPEA)

This problem is conceptually very similar to the SEA problem, nonetheless, there exist an important difference between them; As its name suggests, the OSPEA problem consists not only in determining the best way of distributing available search effort over the region, but also in defining the search path that must be followed by each agent when allocating search effort. The OSPEA problem possess characteristics of both, *routing problems* and *resource allocation problems*. In consequence, it can be inferred that the OSPEA problem is harder than the SEA problem.

As in the case of the SEA problem, the OSPEA also presents a continuous and a discrete version, whose are analogue to those described for the corresponding versions of the SEA problem. By the side of the continuous version, Lukka (1977) first introduced necessary and sufficient conditions for the optimum solution in the case of a single moving target. Agent kinematics are explicitly considered by the model. This version of the problem can be associated to the *Honey-pot problem* and some items in the literature explicitly refers to the problem with that name. The Honey pot problem can be defined as follows:

A bear is searching for a honey-pot hidden in a bounded region. The bear does not have certainty about the honey-pot location, but it has some clues. The bear wants to find the honey-pot before tiring, so he needs to find a search path and a way of allocating its effort (time, energy) in order to maximize its probability of finding the honey-pot soon.

In DasGupta, Hespanha, and Sontag (2004), the Honey-pot problem is tackled under a two phase approach. In phase one, they transform the continuous problem into a discrete one, and then they solve the discrete problem. In the second phase, they implement a refinement method for deriving a solution of the continuous problem, from the solution obtained to the discrete problem. The authors found the discrete version of the Honey-pot problem to be computationally intractable; therefore, they exploited various mathematical properties of the problem to reduce the solution space before executing the solution method in phase one. Their method was limited to produce sub-optimal solutions and bounds for the problem. By their side, Assaf and Sharlin-Bilitzky (1994) focused on a particular case on the problem, where the missing object is known to be hidden in one of m boxes, and it occasionally moves between the boxes according to a given Markovian process. The authors proposed a pure mathematical solution method for this problem.

Alternatively, Ablavsky and Snorrason (2000) also proposed a two phase approach, but their phases are very different to those implemented in In DasGupta et al. (2004). The first phase consists on dividing the search region on a finite number of primitive shapes. This partitioning allows them to determine possible locations of the target for a given future time step. Phase two consists on building a feasible path over the sub-areas defined in phase one, in a way that the total flight time is minimized and the likelihood of finding the target is maximized. The authors assumed a single agent searching for a moving target but announced the adaptability of their methods for the case of multiple homogeneous agents.

In contrast to the decision support system (DSS) presented in Abi-Zeid and Frost (2005) for the SEA problem, in Flushing, Gambardella, and Caro (2012), the authors introduce a support system for the continuous version of the OSPEA problem. This system is able to deal with heterogeneity within the search team. Specifically, such heterogeneity is expressed as agent-dependent maximum speed and sensing range. The benefits of this consideration have been shortly mentioned before and we will provide a wide discussion on this topic later in this document. Similarly to SARPlan, the system in Flushing et al. (2012) is able to process information about topographic characteristics and vegetation distribution over the search region. The authors validated their method with a realistic instance involving three types of agents. The required processing times were not reported in the article.

3.1.4.3 Optimal Search Path (OSP)

In this version of the problem, the search region is often discretized on a finite collection of cells as explained in Section 3.1.3. All exploration resources (e.g. energy, fuel, battery, time, effort) involved on a single exploration over a given cell are predefined. The problem consists in determining the path that each agent should follow in order to obtain a maximum performance in terms of a predefined objective, subject to limited amount of effort. Reader should note that this time, the amount of effort that can be allocated at each exploration is fix, and then the effort allocation problem is absorbed by the path planning problem. By the other side, objective functions in the OSP problem are more general than those in the SEA and OSPEA problems. This fact does not imply that the same variety of objectives cannot be applied to all four types of problems. It reflects that the OSP is a newer search problem than the SEA and the OSPEA problems and current computational power has allowed researchers to make a wider diversification of the OSP problem.

Common objectives for the OSP problem are the maximization of cumulative POS, the minimization of expected search time and the minimization of the cumulative probability of non-success. In Song and Teneketzis (2004), the authors present an optimal methodology for solving an OSP problem with multiple agents under two specific and determinant conditions: (i) the time is discretized in a finite collection of time units; and (ii) the time required for agents to move among cells is zero. The discretization of time limits the ability of the model for adequately describing motion features such as the acceleration and deceleration of agents. Nonetheless according to the second condition stated in the article, those features must not be considered in the problem, since the switch among cells is instantaneous. This second condition in turn, is accepted by the authors to be unrealistic. The OSP problems where travel times are considered, result significantly more difficult to be solved than those without this consideration. However, the authors reported that travel times where not considered in the literature for the OSP problem when this research was conducted.

The authors in Jotshi and Batta (2008) developed graph-theory based heuristics for the problem of searching for an stationary target on a network. In that research, the agent capabilities of moving among nodes is constrained by the topology of the network. It means that an agent is able to move among two nodes only if there exists a link connecting them. This is a typical grounded urban search scenario where agents motion is limited to the configuration of the urban network. In Jotshi and Batta (2008), the agent is supposed to possess a reliability so that the entity is immediately identified with probability equal to 1 if it is hidden at any point of the network where the agent pass through. Both heuristics developed were found to be competitive in comparison with a test-oriented heuristic which finds the k-different open Eulerian tours by doing a partial enumeration existent open Eulerian tours.

The same problem was tackled later by Yu and Batta (2010), but instead of developing an heuristic, they developed a mathematical programming formulation. They first derived a Mixed Integer Quadratic Program (MIQP), tractable for available solvers when the graph size is small. Nonetheless, for medium size graphs, the time required for the solution became intractable. Consequently, the authors implemented a substitution scheme that migrated the model to a Mixed Integer Linear Program (MILP). Finally, they developed an algorithm to solve the MILP for medium-size instances in acceptable running times. Both articles, Jotshi and Batta (2008) and Yu and Batta (2010) assumed a single searcher with perfect detection profile searching for a uniformly-random located stationary entity.

Lo, Berger, and Noel (2012), Berger, Lo, and Noel (2013) and Berger and Lo (2015) are a collection of articles where the OSP was solved for multiple homogeneous agents through mathematical programming. In Lo et al. (2012) authors introduced their modeling scheme, based on a network representation of feasible displacements over the search region. The main contribution of that study is a methodology for computing bounds of OSP instances in acceptable running time. For this purpose, they implemented Lagrangian Integrality Constraint Relaxation to their MILP and solved the resulting model using CPLEX. In Lo et al. (2012), no updating of the POC map was implemented after each exploration. The objective function implemented by Lo et al. (2012) in their optimization program is presented in Equation 3.7 for clarity.

$$\underset{V_{cl}}{\text{Max}} \sum_{c \in N} p_{c0} \left(1 - \sum_{l=0}^{V_c} (1 - p_{cc})^l V_{cl} \right) \tag{3.7}$$

where the assignment of the decision variable $V_{cl} = 1$ represents a solution including l visits to cell c. N is the set of cells $\{1, \ldots, |N|\}$, P_{c0} refers to the current POC of cell c, V_c is an upper bound for the allowed number of visits at cell c and p_{cc} is the probability on a specific visit to correctly detect the target in cell c, given that the target is present there.

Equation 3.7, computes the cumulative POS after multiple explorations to the same cell with no updating on the POC of such cell after each exploration. This equation might be valid in certain contexts where the POC remains constant after explorations. For instance, when a sensing device is scanning items coming from a production line, looking for nonconformities. In this case, the percentage of defectives (which can be associated with the POC) will probably remain constant for several observations. However, in the search context (at least for immobile entities), it is not adequate to assume that POC of a given location remains the same after being explored. Each time that a sensing device indicates a negative result in a given location, our expectancy of finding the target at such location will be reduced. Accordingly, a model for search planning should be able to update the POC after each exploration and decive how long to remain exploring at each location, considering the resulting POC after each exploration. This aspect was attempted to be considered in Berger et al. (2013) where the model proposed in Lo et al. (2012) was extended and the POC of each cell is updated after each time step according to Equation 3.8.

$$p_{c(t+1)} = \sum_{0 \leq l \leq V_c} p_{c(t)}(1 - p_{cc})^l V_{clt} \tag{3.8}$$

Where p_{cc} and V_c have the same meaning than in Lo et al. (2012), $p_{c(t)}$ is the updated POC of cell c at the beginning of time interval t and $V_{clt} = 1$ reflects a cumulative number of explorations l in cell c, by the end of time interval t.

In Berger et al. (2013), the authors claim to update the POC after each exploration. Nonetheless, Equation 3.8 seems to correspond to the probability of not detecting the target after a given number of explorations at the same cell. In contrast, what is required to be computed is the conditional probability of the target occupying a given cell, given that there have been performed certain number of unsuccessful explorations there. In Section 3.1.3 we presented a Bayesian filter suitable for the update of the POC at a given cell after a single exploration. The particular case when agents are assumed false alarm free was also derived and presented in Equation 3.4. The generalization of that Bayesian filter for the update of the POC after a given number of explorations is presented in Sections 4.2.2 and 4.2.3 as part of the contributions of this thesis.

The model presented in the three articles, Lo et al. (2012), Berger et al. (2013) and Berger and Lo (2015) is one of the most complete and fast methods found in the literature. Nonetheless, we make emphasis on the inaccuracy present in their POC update mechanism because we have noted that this is a problem affecting a big portion of the articles in ST literature. The major part of SEA models do not even update the POC after exploring on a given location. The Bayesian filter introduced here is suggested by us as a valid updating mechanism for future researches.

Another important limitation of several OSP models arises when the time step and the spatial step coincide. Given that agents are enforced to occupy a given cell at each time step, they are only allowed to move no adjacent cells between two time periods. This imposition could be suitable for ground of maritime agents, whose exploration is continuous over a smooth path and are not allowed to "jump" between nonadjacent cells. In contrast, this imposition will reduce the effectiveness of an agents capable to "jump" between nonadjacent cells. Aircrafts are examples of these kind of agents, which can go from a location to another without exploring at every point in their way. The particular type of problems where agents displacements are limited to adjacent cells, are commonly referred as *constrained search problems*. Despite the evident limitation imposed by this rule, it has been consistently adopted in multiple studies, including those in Yan and Blankenship (1988), Eagle and Yee (1990), Hohzaki and Iida (1997), Dodin, Minvielle, and Le Cadre (2007), Hong, Cho, and Park (2009), Hong, Cho, Park, and Lee (2009), Morin, Abi-Zeid, Lang, Lamontagne, and Maupin (2009), Lanillos, Besada-Portas, Pajares, and Ruz (2012), Lo et al. (2012) and Berger and Lo (2015). However, some authors such as

Chung et al. (2009) also criticize the adoption of this rule based on the limitations here discussed.

The assumption of *local effectiveness of search effort* discussed in Section 3.1.4.2 was relaxed by Israel, Khmelnitsky, and Kagan (2012) for the OSP problem. This assumption is not very limiting for the OSP problems, since it is often assumed that the sensing range of an agent circumscribes at least a single cell with could represent hundreds of meters in a real mission. However, Israel et al. (2012) went further in the relaxation of this assumption and introduced the search on a topographic terrain. The authors assumed natural limitations to the sight of view of grounded agents such as mountains. Under this assumption, they defined two main types of areas; shadowed areas, which are those that are not visible to the agent due to the presence of some obstacle; and the observed areas, which are those areas that are clearly visible to the agent. Figure 3.3 illustrates the idea.

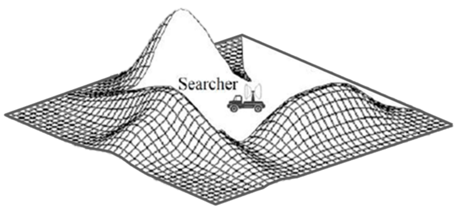


Figure 3.2: Field of view in a topographic terrain
Retrieved from: Israel et al. (2012)

Similarly, Sarmiento, Murrieta, and Hutchinson (2003) provided an herustic method to solve the OSP problem over a region which may present holes. Those holes can represent obstacles that impose motion and visibility constraints. Their method is based on a greedy algorithm that selects the most convenient location to be visited in the next step, taking the POC and the time required to reach such location as criteria. This heuristic can be solved in polynomial time, providing a relatively good but sub-optimal solution to the problem. Various other researches have implemented branch and bound algorithms for different versions of the OSP problem. For instance Eagle and Yee (1990) applied a branch and bound algorithm for the path constrained search problem with a single searcher and Dodin et al. (2007) also implemented a branch and bound algorithm, but this time for a general OSP problem. By its side, Washburn (1997) tested a branch and bound algorithm for the path constrained search problem but he focused on comparing the performance of the algorithm when defining different bounds for the computational time. It should be noted that any solution of a path planning problem which allows repeated visits to the same point is implicitly an OSPEA problem since the effort assigned to each point can be determined by standardization of the number of visits assigned to each point over the whole number of visits assigned to all points of the region. The reciprocate transformation is also feasible.

3.1.4.4 Probabilistic Search Control (PSC)

The PSC problem has the same two purposes than the OSPEA problem; (i) to determine the best places to allocate search effort and (ii) to determine the best path to be followed by each agent in the distribution of the search effort. The main difference between both types of problem is that OSPEA is solved under a planning perspective, while PSC is solved under a control perspective. The difference among planning and control approaches is that a single run of a planning model determines the solution for multiple steps of each agent, while control algorithms guide search units step by step through the mission. In order to build the search paths, a decision rule is first established. Such a decision rule is typically a greedy optimization program that determines the next position of each agent after each exploration. To start, each agent moves to a given cell according to the specified rule and performs one exploration there. Then, the POC map is updated based on findings of the agents and a new decision is taken by means of the decision rule. The process is repeated until the target is found or a given stopping condition is reached. This type of problem is oriented to automation contexts where there are robotic agents performing the mission.

In Bourgault, Furukawa, and Durrant-Whyte (2004b), authors addressed a PSC problem where multiple agents search for a single, moving, non-evader target. In that study, the authors assessed the impact that different types of obstacles present in the search region, could make on the progress of the mission and then on the evolution of the POC map. The constraints were demonstrated to have a significant effect on the computation of the updated POC map, leading to over and underestimation of the importance of the cells and the affecting the search sequences.

In Collins, Riehl, and Vegdahl (2007), the authors considered a PSC problem where multiple UAVs and their sensing devices are controlled. This is the only study tackling a PSC problem that did not applied a Bayesian filter for the updating of the POC map. In contrast, they implement Equation 3.8, which is the one adopted by Berger et al. (2013) for their OSP problem. A wide discussion on the inaccuracy of such an expression was conducted in the previous section and there is nothing more to say, than suggesting the use of Bayesian filters for this purpose, which is the adequate statistical tool for the conditional nature of the updated POC.

A dynamic programming approach was implemented in Lau, Huang, and Dissanayake (2005) for solving a PSC problem in a known built environment. Their approach allows the agents to decide their displacements, based on projected information of the POS at each location. The algorithm presents an acceptable solution time for instances up to 14 regions.

The updating of the POC map depends on the assumptions about the reliability profile of the agents. Generalizing the idea introduced in Section 3.1.3, the reliability of an agent depends on its accuracy indicating the presence or absence of the target given that the target is actually present or not. In this sense, an agent can be prone to deliver false positives, false negatives, neither or both types of verdicts. As it was clearly stated, the Bayesian filter derived in Section 3.1.3 corresponds to the particular case where agents are false alarm free, meaning that they will never indicate that the target is present on an empty cell. The majority of SEA, OSPEA and OSP related models work with this assumption. In contrast, many PSC studies assumes agents prone to both types of errors. We did not found any study in this survey where the authors assumed neither a perfect sensor nor a sensor prone just to false positives. In Table 3.3 we classified some PSC related studies found in this survey according to their assumption about sensor reliability.

Table 3.3: Sensor reliability on PSC problems

False positive	False negative	Papers
	x	Simonin, Cadre, and Dambreville (2008); Sun and Duan (2012); Gan, Fitch, and Sukkarieh (2012); Furukawa, Mak, Durrant-Whyte, and Madhavan (2012); Nguyen, Lawrance, Fitch, and Sukkarieh (2013); Lanillos, Gan, Besada-Portas, Pajares, and Sukkarieh (2014); Kassem and El-Hadidy (2014)
x	x	Chung and Burdick (2007, 2008); Chung (2009); Carpin, Burch, and Chung (2011); Kwok and Holsapple (2011); Chung and Burdick (2012)

When there are two or more agents performing the mission, the PSC problem can be tackled with a centralized or decentralized approach. In a centralized scheme, each agent shares the information achieved in its last exploration with a central system. Then, this system updates a general POC map and makes the decision about the following movement of each agent. This system is responsible of ensuring collision avoidance and also a good performance of the team as a hole system. Some examples of PSC problems solved under a centralized scheme can be found in Furukawa et al. (2012) and Wong, Bourgault, and Furukawa (2005). On the other hand, decentralized schemes do not consider any central system managing the POC updating or the decision making process. In contrast to the centralized scheme, agents share their findings with the other agents and then each agent decides its following destination independently. A common assumption in decentralized schemes, is that agents are equipped with a collision avoidance system. Some examples of PSC under decentralized schemes can be found in Bourgault, Furukawa, and Durrant-Whyte (2003), Bourgault, Furukawa, and Durrant-Whyte (2004a) and Kwok and Holsapple (2011). We provide a brief comparison between both schemes in Table 3.4, where best qualified method on each comparison variable is signed by *.

A typical objective in PSC problems is the maximization of the POS at each step. As stated in previous section, the maximization of the cumulative POs is the natural objective whenever it is available a POC map and the reliability profiles of the agents. Given that the PSC implies multiple independent decisions along the search path, the maximization of POS at each step becomes the natural objective. Researches where this objective was implemented can be found in Bonnie, Candido, Bretl, and Hutchinson (2012), Chung and Burdick (2007) and Saito, Hatanaka, and Fujita (2010).

Another interesting objective function could be the maximization of the amount of information about the target, collected by the search team at each step. In a real search mission, the amount of information collected depends on the resolution implemented to explore a given area region and also on the value of the POC at this area. Taking this into consideration, the amount of information gathered by a search team can be estimated by

Table 3.4: Comparison between centralized and decentralized schemes

Indicator	C	D	Comments
Robustness		*	It is not necessary for each agent to share information with a central system. Communication delays due to instability of the communication network will be mitigated since information will be shared just between close-by agents
Flexibility		*	On a decentralized scheme, agents are semi-independent and autonomous for deciding where to go at each step. In a centralized scheme, this independence is not possible, since all agents need the information of the other ones to know their following location. In real search missions agents must be able to perform tasks independently to the current state of other agents
Solution quality	*		The decisions that agent takes, affects the future decisions of other agents and their expected performance. Optimal solutions for the whole search team at each time step, could not be warranted under decentralized schemes

C: Centralized scheme

D: Decentralized scheme

the sum of the POS at each exploration, weighted by a factor representing the resolution implemented at each area. To account for multiple resolutions, studies in Carpin et al. (2011), Chung and Carpin (2011) and Carpin, Burch, Basilico, Chung, and Kölsch (2013) adopted a data structure known as *quadtree*. By means of this data structure, authors in those studies are able to deal with the trade-off between sensing range/field of view and resolution. When an air vehicle explores an area at a higher elevation, field of view increases but its resolution diminishes. The opposite occurs when it explores a region at a lower elevation. Figure 3.3 illustrates the idea with a quadtree.

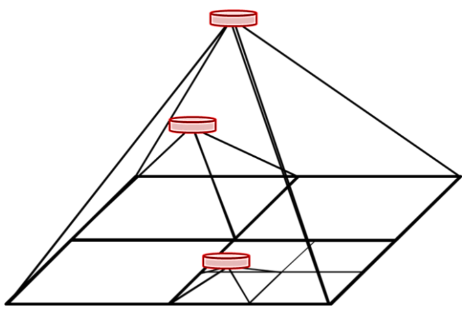


Figure 3.3: Quadtree implemented on PSC with variable resolution

It will be ideal to explore the whole search region with the higher possible resolution; nonetheless, it requires to explore at a very low altitude and takes much more time than exploring the region at a lower resolution. Models implementing quadtrees are able to decide at which level each area should be explored, dealing with the trade off between speed and resolution.

Hubenko, Fonoberov, Mathew, and Mezić (2011) also implemented a variable resolution framework for a PSC problem. In contrast to the works in Carpin et al. (2011), Chung and Carpin (2011) and Carpin et al. (2013), this author attributes the existence of variable resolution to instability on the radius of the sensor due to uncertainties about the search region and not to voluntary adaptations performed by the agents when exploring. In their model, the radius of the sensor is assumed to be a random variable that varies over a known interval. Their algorithm was compared with the billiard search and results advice that their method vastly outperforms the test-method.

In order to reduce the myopic component present in the PSC, Chung and Burdick (2007) adopted a receding horizon scheme to a PSC problem. They compared the performance of their algorithm with the results obtained through random walk and search methods inspired in saccadic movements of human eyes and drosophila's search for food. Results evidenced that longer look-ahead windows on receding horizon control have the potential of improving the performance of the mission. They also found that receding horizon control outperforms the other two search strategies tested in their study since it requires fewer steps to find the target.

All previously mentioned studies assumed simple binary detection models for the detection capabilities of the sensors. In Lanillos, Besada-Portas, Lopez-Orozco, and de la Cruz (2014), the authors tackled a PSC problem dealing with complex non-linear/non-differential models. To solve this problem, they implemented a Cross Entropy Optimization (CEO) algorithm which was proved to find attractive solutions accumulating POS faster than approaches in other works.

3.1.5 Characteristics of reviewed articles

Having defined our classification of ST representative problems, we proceed with the breakdown of literature, based on seven features belonging to the problem setting. Figure 3.4 summarizes the eleven categories of problem setting features considered by us. Categories where selected with the objective of providing a broad panorama of the problem setting considered in each study. With this in mind, we declared categories related with three basic components of any search problem: agents, targets and spatio-temporal framework. Each study in our sample was classified according to the type of problem addressed and then by each one of those categories. Results are illustrated in Tables 3.5 to 3.8.

Note that each category allows to classify a given study in one of two possible configurations. Nonetheless, there are some studies that may not be classified in any of two configurations on a given category. This happens specifically when a study considers a single search agent. Then, categories 2 to 7 become immediately not applicable to that study, since they are oriented to studies considering multiple agents. In that cases, we filled all cells from categories 2 to 7 with the symbol "-".

Agent features

1. Number of agents
    1.1. One
    1.2. Multiple
2. Decision scheme
    2.1. Centralized
    2.2. Decentralized
3. Effort
    3.1. Homogeneous
    3.2. Heterogeneous
4. Agent abilities
    4.1. Homogeneous
    4.2. Heterogeneous
5. Agent reliabilities
    5.1. Homogeneous
    5.2. Heterogeneous
6. Sensing range
    6.1. Homogeneous
    6.2. Heterogeneous
7. Travel times
    7.1. Homogeneous
    7.2. Heterogeneous

Target features

8. Number of targets
    8.1. One
    8.2. Multiple
9. Target state
    9.1. Stationary
    9.2. Moving

Scenario features

10. Time modeling
    10.1. Discrete
    10.2. Continuous
11. Space modeling
    11.1. Discrete
    11.2. Continuous

Figure 3.4: Categories for ST Taxonomy

Table 3.5: Setting of SEA problems - 17 studies

Study	1		2		3		4		5		6		7		8		9		10		11	
	1.1	**1.2**	**2.1**	**2.2**	**3.1**	**3.2**	**4.1**	**4.2**	**5.1**	**5.2**	**6.1**	**6.2**	**7.1**	**7.2**	**8.1**	**8.2**	**9.1**	**9.2**	**10.1**	**10.2**	**11.1**	**11.2**
B. Koopman (1956a)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X			X		X
B. Koopman (1956b)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X			X		X
B. O. Koopman (1957)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X			X		X
Charnes and Cooper (1958)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X			X	X	
Brown (1980)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X		X	
B. O. Koopman (1979)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X			X		X
Stone (1979)	X		-	-	-	-	-	-	-	-	-	-	-	-		X		X	X		X	
Hohzaki and Iida (2001)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X		X	X	
Iida et al. (2002)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X			X	X	
Dambreville and Le Cadre (2002)		X	X			X	X			X			X			X		X	X		X	
Dambreville and Le Cadre (2004)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X		X	
Abi-Zeid and Frost (2005)		X	-	-		X	X		X		X		X		X		X		X		X	
Hohzaki (2006)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X		X	
Simonin et al. (2009)		X	X		X		X			X	X		X		X			X	X		X	
Beltagy and Abd Allah El-Hadidy (2013)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X			X		X
Le Thi, Nguyen, and Dinh (2014)		X	X			X	X			X	X		X		X		X			X	X	
Wang and Zhou (2015)		X	X		X		X		X		X		X		X			X		X	X	
Participation	71%	29%	24%	0%	12%	18%	29%	0%	12%	18%	24%	0%	29%	0%	88%	12%	53%	47%	41%	59%	71%	29%

Table 3.6: Setting of OSPEA problems - 5 studies

Study	1		2		3		4		5		6		7		8		9		10		11	
	1.1	**1.2**	**2.1**	**2.2**	**3.1**	**3.2**	**4.1**	**4.2**	**5.1**	**5.2**	**6.1**	**6.2**	**7.1**	**7.2**	**8.1**	**8.2**	**9.1**	**9.2**	**10.1**	**10.2**	**11.1**	**11.2**
Lukka (1977)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X		X		X
Assaf and Sharlin-Bilitzky (1994)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X		X		X
Ablavsky and Snorrason (2000)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X		X	
DasGupta et al. (2004)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X			X	X	
Flushing et al. (2012)		X	X		X		X		X			X		X	X		X		X		X	
Participation	80%	20%	20%	0%	20%	0%	20%	0%	20%	0%	0%	20%	0%	20%	100%	0%	40%	60%	40%	60%	60%	40%

Table 3.7: Setting of OSP problems - 34 studies

Study	1		2		3		4		5		6		7		8		9		10		11	
	1.1	**1.2**	**2.1**	**2.2**	**3.1**	**3.2**	**4.1**	**4.2**	**5.1**	**5.2**	**6.1**	**6.2**	**7.1**	**7.2**	**8.1**	**8.2**	**9.1**	**9.2**	**10.1**	**10.2**	**11.1**	**11.2**
Yan and Blankenship (1988)	X		-	-	-	-	-	-	-	-	-	-	-	-		X		X	X		X	
Eagle and Yee (1990)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X		X	
Santos (1993)		X	X		X		X			X	X		X		X			X	X		X	
Dell, Eagle, Alves Martins, and Garnier Santos (1996)		X	X		X		X			X	X		X		X			X	X		X	
Washburn (1997)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X		X	
Hohzaki and Iida (1997)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X		X	
Hohzaki and Iida (2000)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X		X	
Sarmiento et al. (2003)	X		-	-	-	-	-	-	-	-	-	-	-	-	-	-	X			X		X
Song and Teneketzis (2004)		X	X		X		X		X		X		X		X		X		X		X	
Grundel (2005)		X	X			X	X		X		X		X		X			X	X		X	
Lau, Huang, and Dissanayake (2006)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X		X	
Dodin et al. (2007)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X		X	
Jotshi and Batta (2008)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X			X	X	
Morin et al. (2009)	X		-	-	-	-	-	-	-	-	-	-	-	-		X		X	X		X	
Hong, Cho, and Park (2009)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X		X	
Hong, Cho, Park, and Lee (2009)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X		X	
Chung et al. (2009)		X	X		X			X		X	X			X		X		X	X		X	
Royset and Sato (2010)		X	X		X		X			X	X			X		X		X	X		X	
Yu and Batta (2010)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X			X	X	
Waharte and Trigoni (2010)		X	X		X		X			X		X		X	X		X		X		X	
Morin, Lamontagne, Abi-Zeid, and Maupin (2010)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X		X	
Lo et al. (2012)		X	X		X		X		X		X		X		X		X		X		X	
Lanillos et al. (2012)	X		-	-	-	-	-	-	-	-	-	-	-	-		X		X	X		X	
Israel et al. (2012)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X		X		X
Forsmo, Grøtli, Fossen, and Johansen (2013)		X	X		X		X		X		X		X		X		X		X			X
Berger et al. (2013)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X		X		X	
Flushing et al. (2013)		X	X		X			X		X	X		X		X		X			X	X	
Berger, Lo, and Noel (2014)		X	X		X		X		X		X		X		X		X		X		X	
Khan, Yanmaz, and Rinner (2014)		X	X		X		X			X	X		X		X		X		X		X	
Lee and Morrison (2015)		X	X			X	X		X		X			X		X		X	X		X	
Kriheli, Levner, Bendersky, and Yakubov (2015)		X	X		X			X	X		X		X		X		X			X	X	
Berger and Lo (2015)		X	X		X		X		X		X		X		X		X		X		X	
Yetkin, Lutz, and Stilwell (2016)		X	X		X			X	X		X		X			X	X		X		X	
Berger, Lo, and Barkaoui (2016)		X	X		X		X		X		X			X	X		X		X		X	
Participation	50%	50%	50%	0%	44%	6%	38%	12%	29%	21%	47%	3%	35%	15%	74%	24%	44%	56%	82%	18%	91%	9%

Table 3.8: Setting of PSC problems - 27 studies

Study	1		2		3		4		5		6		7		8		9		10		11	
	1.1	**1.2**	**2.1**	**2.2**	**3.1**	**3.2**	**4.1**	**4.2**	**5.1**	**5.2**	**6.1**	**6.2**	**7.1**	**7.2**	**8.1**	**8.2**	**9.1**	**9.2**	**10.1**	**10.2**	**11.1**	**11.2**
Bourgault et al. (2003)		X		X	X		X			X	X		X		X		X		X		X	
Bourgault et al. (2004b)		X		X	X		X		X		X		X		X			X	X		X	
Bourgault et al. (2004a)		X		X	X		X			X	X		X		X			X	X		X	
Lau et al. (2005)	X		-	-	-	-	-	-	-	-	-	-	-	-		X	X		X		X	
Wong et al. (2005)		X	X		X		X		X		X		X			X		X	X			X
Chung and Burdick (2007)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X		X		X	
Collins et al. (2007)		X	X		X		X		X		X		X			X		X	X		X	
Chung and Burdick (2008)		X		X	X		X			X	X		X		X			X	X		X	
Owen and McCormick (2008)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X		X	
Simonin et al. (2008)	X		-	-	-	-	-	-	-	-	-	-	-	-		X	X		X		X	
Chung (2009)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X		X		X	
Saito et al. (2010)		X			X		X		X		X		X		X			X	X		X	
Carpin et al. (2011)	X		-	-	-	-	-	-	-	-	-	-	-	-		X	X		X		X	
Saito, Hatanaka, and Fujita (2011)		X	X		X		X		X		X		X			X		X	X		X	
Kwok and Holsapple (2011)		X		X	X		X		X		X		X		X			X	X		X	
Chung and Carpin (2011)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X		X		X	
Hubenko et al. (2011)		X	X		X		X		X		X		X		X		X		X		X	
Chung and Burdick (2012)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X		X		X	
Sun and Duan (2012)	X		-	-	-	-	-	-	-	-	-	-	-	-	X			X	X			X
Gan et al. (2012)		X		X	X		X		X		X		X		X			X	X		X	
Furukawa et al. (2012)		X	X		X		X		X		X			X	X			X	X		X	
Bonnie et al. (2012)		X		X	X		X		X		X		X		X		X		X			X
Nguyen et al. (2013)	X		-	-	-	-	-	-	-	-	-	-	-	-	X		X		X			X
Carpin et al. (2013)	X		-	-	-	-	-	-	-	-	-	-	-	-		X			X		X	
Kassem and El-Hadidy (2014)		X	X		X		X		X		X		X		X			X	X		X	
Lanillos, Gan, et al. (2014)		X		X	X		X		X		X		X			X	X		X		X	
Lanillos, Besada-Portas, et al. (2014)		X	X		X		X		X		X		X		X			X	X		X	
Participation	41%	59%	26%	30%	59%	0%	59%	0%	48%	11%	59%	0%	56%	4%	70%	30%	44%	52%	100%	0%	85%	15%

Using our taxonomy to identify useful sources

In Section 3.1.2, we state that our taxonomy allows decision makers and researchers to find studies corresponding to the same type of problem that they are facing and also taking a similar problem setting into consideration. Here we will evidence this potential with a simple hypothetical case.

Please consider a researcher planning to address the problem of searching for multiple targets on the sea. Furthermore, he wants to account for search missions involving different types of agents such as boats, aircrafts and satellites. Additionally, he wants to address the problem from a strategic point of view, where search paths are not required but search effort must be allocated at each sub-region of the possibility area. Then, the search within each sub-region is assumed to be performed in a parallel sweep scheme.

Having understood our classification of the types of search problems, the researcher will associate his particular problem with a SEA problem, given that he does not need to define search paths, but he need to allocate the effort. Taking this into account, a first step will be to focus on Table 3.5, which contains studies tackling the SEA problem. By inspecting the Table, the researcher will find interesting the works at (Dambreville & Le Cadre, 2002), (Abi-Zeid & Frost, 2005), (Simonin et al., 2009), (Le Thi et al., 2014), (Wang & Zhou, 2015). All of these studies coordinate multiple search agents accounting for different sources of heterogeneity. Particularly, he may want to detail the work in (Dambreville & Le Cadre, 2002), which accounts for multiple targets. This way, our taxonomy allowed our hypothetical researcher to easily find five useful studies for his particular purposes.

Statistical findings

Multiple inferences can be done from the classification in Tables 3.5 to 3.8. To start, it is evident a balance between studies considering a single searcher and multiple searchers in both, the OSP and PSC problems. Contrarily, the SEA and the OSPEA problem show a very less number of studies considering multiple agents than studies considering single agents. The SEA, presents just a 29% of the studies tackling problems with multiple agents (See Table 3.6). This low participation is partially explained by the fact that the SEA was the first search problem addressed in ST literature, and was first modeled under the simplest problem setting. By its side, the OSPEA reports a single study, corresponding to the 20% OSPEA related studies, considering multiple agents.

Other interesting finding is that the decentralized decision scheme was only implemented in PSC problems of our sample. It may be explained by the fact that the other three types of problems are solved by means of a single run of a planning model that coordinates the whole search team. However, a decentralized planning scheme where planning models are solved independently for sub-sets of agents should not be discarded. This kind of modeling may be suitable for cases where the tasks performed by a group of agents, depend on the tasks performed by other group of agents. The problem addressed in (Chung et al., 2009) is an example of such a case. In that study, a group of aerial agents are responsible for identifying targets from air and ground agents are responsible for intercepting and verifying devised targets.

Among the different sources of heterogeneity on the search team, agent reliabilities and and travel times (categories 5 and 7, respectively) are the most commonly tackled.

Both sources of heterogeneity have been considered by multiple studies in three of the four types of problems. By its side, agent abilities have been considered as a source of heterogeneity only in the OSP problem. Studies addressing this feature correspond to the 12% of OSP problems and the 5% of the whole sample. A more detailed exploration of the tables allows to identify combinations of sources of heterogeneity that have and have not been considered yet. Here we revised the interactions between two sources of heterogeneity. The combinations (effort - abilities), (effort - sensing range) and (abilities - sensing range) were not considered simultaneously as sources of heterogeneity in any of the studies of our sample. By its side, the OSP has been modeled, accounting for the majority of combinations between sources of heterogeneity. The following sources have been considered exclusively in OSP problems: (effort - travel times), (abilities - reliabilities), (abilities - travel times), (reliabilities - sensing range) and (reliabilities - travel times). SEA problems have only considered (effort - abilities) as a combined source of heterogenity. By its side, the OSPEA have only been modeled under the combined source of heterogeneity (sensing range - travel times). This analysis considering only combinations of two sources of heterogeneity points out multiple opportunities of improvement for future researches.

Categories 8 and 9 which are focused on target features, also present interesting behaviors. The scenario with multiple targets have been scarcely addressed in all types of problems. In fact, that scenario was not addressed by any study tackling an OSPEA problem in our sample. Problems involving a single target are often simpler to be solved than those involving multiple targets, specially if the targets are moving since that condition would imply the definition of target-dependent motion models and special treatments for POC update. However, a total of 10 studies in our study, representing the 13% of the sample, addressed problems involving multiple targets, and 7 of them also considered moving targets.

The last two categories to be assessed are 10 and 11, related to scenario features. From the participation at each category, we can appreciate that both, the discrete and continuous scheme for time and space have been consistently addressed in the literature, with some particular exceptions. All PSC problems in our sample discretized the time, and the 85% of them discretized the space. By the other side, only the 9% of the articles related to the OSP problem, considered continuous space. That statistic coincides with our description of the OSP problem in Section 3.1.4.3, where we advice that this problem is often modeled over a discrete region composed of cells. The use of discrete or continuous time and space can represent an advantage or a weakness of a model, depending on the specific context where the model is going to be applied. Inventory management problems are often solved over a time horizon which is naturally discretized by the seasons. In that case, the time discretization results a suitable approach. By the other side, machine scheduling problems usually involve heterogeneous jobs associated with different processing times. The use of a discretized time horizon in this context may imply the dilatation or contraction of real processing times in order to be fitted to the discretized time steps. This second case is a scenario where the time discretization may impose great limitations to the model in describing the reality. Real search missions commonly involve displacements over the region consuming different amounts of time. As the search region becomes larger, the imposition of time discretization will cause major distortion in the model, since multiples of the time size will be necessarily used to allow the agent for completing short and also very long displacements. To avoid such a distortion, many studies in ST limit agents displacements to the neighbor cells from their current position. Unfortunately, as we explained in Section 3.1.4.3, this imposition is only valid for a special type of agents whose exploration is continuous over a smooth path. The time discretization significantly limits the ability of a model to account for agents that are able to “jump” between distant cells without focusing on other cells in their way.

The 45% of our sample tackled their search problems under discretized time schemes. SEA and OSPEA problems are the two ones where the continuous time has attracted more attention from researchers. Space discretization does not present clear limitations to search problems. In fact, it fits very well with the procedures that many search agencies implement in practice and allows search planners to assign specific tasks to agents during the mission.

A comparison on the number of publication related to each type of problem shows that the OSP is the most frequently type of problem addressed in our sample, with 34 studies. It is followed by the PSC problem with 27 studies. An analysis on the evolution of the scientific production by type of problem shows that this has not always been the situation. Figure 3.5 provides the basis for such an analysis.

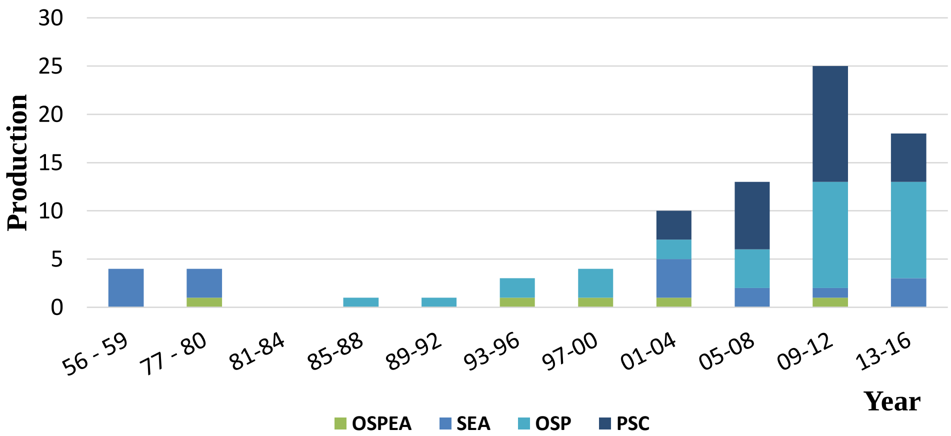


Figure 3.5: Scientific production per period - problem

It is evident from Figure 3.5 that ST literature has not only been growing at a higher rate each time, but its constitution has suffered a transformation through years. In the early years of ST, scientific literature was mainly composed by studies addressing SEA problem. As stated in the introduction, the evolution of computers had an important influence on the evolution of ST literature. The availability of cheaper and faster computers have progressively increase the range of characteristics that algorithms and models can simultaneously take into account. As a result, the OSP and the PSC problems, which are highly supported on computational processing, gained important participation on ST field. The OSPEA seems to be a type of problem that has been consistently addressed at a very low rate. However, as we mentioned in Section 3.1.4.3, any OSP problem allowing multiple visits to the same region can be associated with an OSPEA problem. By this point of view, the participation of the OSPEA problem may also have increased with the years.

Figure 3.5 also shows a rising trend in scientific production. Such tendency is expected to persist as there still been multiple areas of improvement in the literature, some of them evidenced in Tables 3.5 to 3.8.

3.1.6 Research gaps

Based on our survey, we state the following research gaps present on ST literature:

1. **The presence of a non-instantaneous disaster on the search region has not been considered yet**

   Most part of the sample in this review, only considered the POC map and the coefficients of the transition matrix as criterion for conducting the search mission. In scenarios where the search region is threatened by a non-instantaneous disaster, it should be given some priority to those zones which the threatening phenomenon have a grater potential to affect in near future periods. There is not any study in our survey that has considered this fact in their model.

2. **Mono-objective problems**

   The majority of studies in our survey modeled their problem as a single-objective problem. As stated before, the majority of decision making real problems pursue more than a single objective, and each missed objective in the formulation diminishes the validity of the model for real scenarios. In the case of search problems, combinations of the following objectives, and so on, might be included in a decision making model:

   - Minimization of the expected span of the mission
   - Minimization of the number of cells being exhausted by the disaster before being explored
   - Maximization of the area covered by the agents within a given time lapse
   - Maximization of the cumulative POC
   - Maximization of the cumulative POS

3. **Missing or out of context updating of POC map**

   Several path planning models included in our survey keep the POC map constant over each time window for which the model is solved. This fact considerably reduces the efficacy of the search mission since the potential effect of explorations performed by agents over a given location are not considered until the model is solved again. If such probability is updated after each agent explores a cell, then all agents including itself will be able to take their following decision based on updated data about the expectancy of finding the target in different locations. Some of the articles that aim to update the POC map within each time window, applied out of context formulations that diminish the accuracy of their methods. Bayes formulations seems to be the adequate tool for executing such update.

4. **Time discretization**

   Several authors in our sample discretized the time in their models in order to simplify the handling of some elements in their formulations such as collision avoidance and updating of POC map. Nonetheless, the discretization of the time considerably limits the capacity of the model to account for some features in the motion of search agents such as heterogeneity among their maximum velocities and accelerations. Additionally, the time discretization forces the agents to perform unnecessary explorations on their way to an attractive exploration zone.

5. **Limitation on agents movements only to adjacent cells**

   This imposition causes serious limitations to missions where agents are able to perform fast displacements among distant points without focusing in each section of their paths. The imposition of this rule will prohibit such kind of displacements.

In order to fulfill those gaps defined above, we proposed the following research at the proposal stage of this thesis:

> The construction of a search methodology for scenarios where the search region is threatened by a non-instantaneous disaster. Such methodology will include a mathematical model for path planning capable of combining the objectives of prioritizing explorations in the most critical cells with the objective prioritizing explorations in cells with the higher expectancy of holding the victim. This model will be suitable for multiple heterogeneous agents that may be joined to the mission by different emergency management agencies and research groups for real search missions. In this case, the heterogeneity of agents will be associated with agent-dependent probabilities of detection, travel times and exploration times. Additionally, the POC map will be updated after the exploration performed by each agent on each cell. This fact will improve the expected results of the decisions taken by agents within a given time window. A network topology will be imposed over the search region in order to allow displacements among any two locations, enhancing the description of agents' motion in comparison with models that discretize the time and those that prohibits displacements to not adjacent cells.

At this time, our research has been executed and the stated methodology was generated according to the description provided in the paragraph above.

Chapter 4

Problem Modeling

In the previous chapter, we presented a literature review in Search Theory which allowed us to identify existent research gaps. In the present chapter, we illustrate the methods and techniques that we implemented during the execution of this thesis in order to fulfill such gaps.

We start by providing an overview of the search methodology environment. It includes a general description of the system where our products will perform in cooperation with other components. Then we introduce and explain our mathematical model in both, its nonlinear and linear version. We also present the solution methods developed and implemented for the solution of our research problem. Finally, we describe the simulation scheme adopted to emulate disaster's behavior.

4.1 Search methodology environment

4.1.1 Components of the search methodology

In this thesis we have focused on the development of a OR-based methodology for search planning under dynamic disaster scenarios. Such a methodology is mainly composed by a mathematical programming model and a solution method able to produce feasible search sequences over most attractive zones of the search region. Although our focus is on the development of those two products, for a better comprehension here we provide an overview of those and other important components that would take part on real SAR missions. Figure 4.1 illustrates those components.

The Danger map will be explained in Section 4.1.2, for now it can be seen a numerical grid that indicates the current and predicted location and severity of the disaster.

The monitoring system is any kind of sensing system capable of tracking the disaster. Its objective is to collect relevant information about the current state of the disaster that could be used to forecast its future state. This system could be composed by a fleet of UAVs provided with different kind of sensing devices depending on the disaster type. For instance, In case of a wildfire, the sensing device could be thermal camera capable of detecting different temperatures in the search region.

The processing system is a computerized module that is engaged of forecasting the future state of the disaster and running our search planning model, taking resulting forecast as an input. It is also responsible of the storage and updating of the POC map.

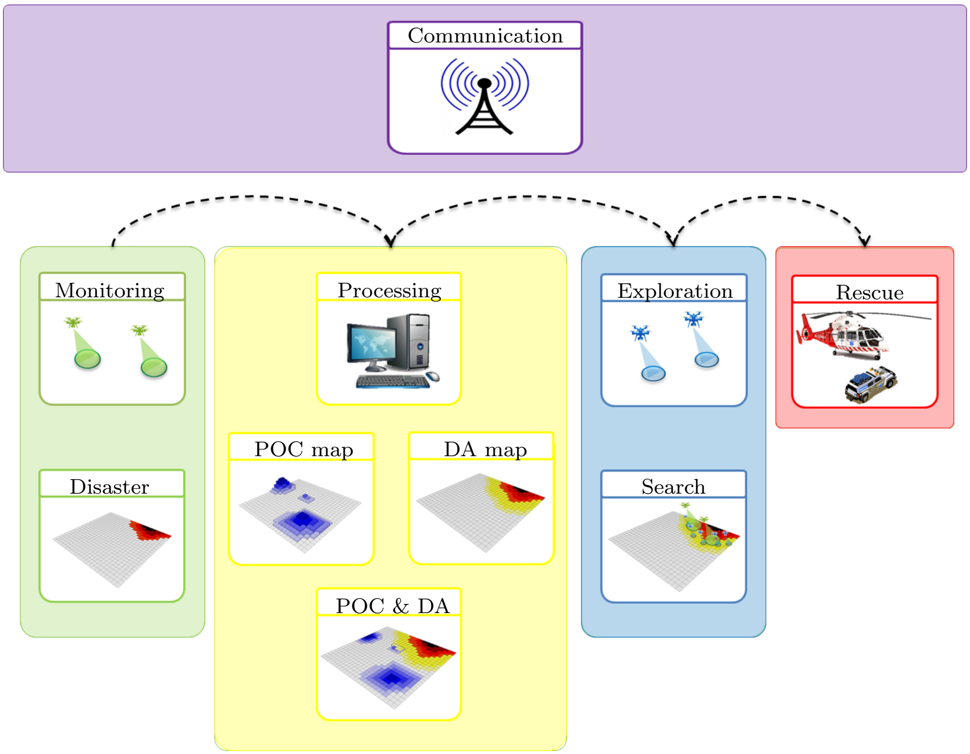


Figure 4.1: Systems involved in our SAR system

The exploration system is composed by a team of searchers with heterogeneous features. In this thesis, such heterogeneity is represented by agent-dependent travel times, exploration times and sensing capabilities (reliabilities), specifically agent-dependent probabilities of detection.

As its name suggests, the rescue system is composed by the team of rescuers and their vehicles. Depending on the danger of the scenario, the rescue team could be supporting the exploration system in the search task or waiting in a safe place for the advice of the exploration system to aid a detected victim.

Finally, the communication system is typically a wireless network which supports the information sharing between the other systems involved in the mission. This system is critical, because in real search missions, it is prompt to suffer interruptions and then part of the information does not reach the destination. There is an ongoing research field on the design of robust wireless networks for mobile systems (Frew & Brown, 2009; Ho, Grøtli, Sujit, Johansen, & Borges Sousa, 2013; Bekmezci, Sahingoz, & Temel, 2013). This topic is not the main subject of this thesis and consequently, we will not address it here. For this thesis, we will assume that the communication system is perfect and imperturbable.

In this research, we are only concerned about the functioning of the processing and the exploration system. The construction of the monitoring and rescue systems, the wireless network and the navigation controllers for cases of autonomous search vehicles are beyond the scope of this thesis. We expect those elements that will not be addressed here, could be developed in future research.

Simulation of a disaster demands some expertise in the factors that influence its behavior. The acquisition of this expertise may demand investment of several hours studying and learning how to operate the simulator. Due to the limited time budget available for this research and without loss of generality, we decided to select only one type of disaster to be considered for validations of our products. Specifically, we decided to work with wildfires, given the availability of simplified fire propagation models suitable for our purposes.

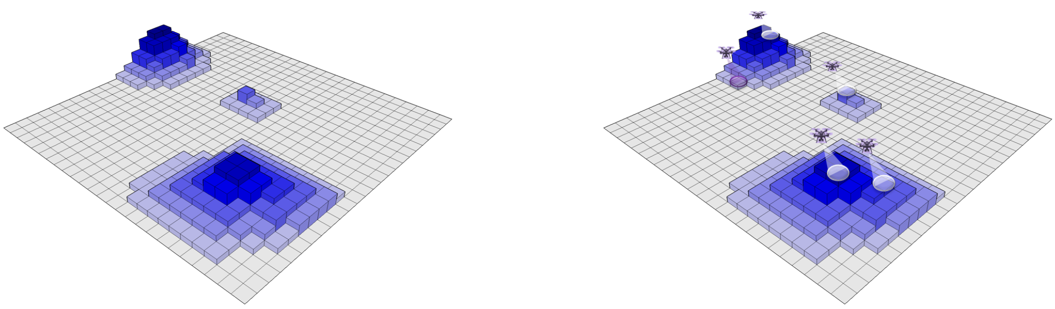
4.1.2 Search based on Danger and POC maps

Our search methodology is based on the aggregation of the five systems described in Section 4.1.1. In that methodology, those systems work together as a macro system which executes a search mission pursuing the following two objectives:

- Maximization of the cumulative probability of success collected by the agents.
- Maximization of the cumulative danger collected by the agents.

By collected we refer to the sum of the updated POC and the sum of the danger of the cells that the agents visit through their search paths. To explain how this combination will interact, we will first explain each one of them separately.

In a search based on a POC map, the responsible of the mission constructs an initial POC map, based on the information available about victims location at the beginning of the mission. As explained in Section 3.1.3 - Task 2, typical aspects considered for the construction of the POC map are the topographic characteristics of the terrain, the last reported sighting of the victim, and the coordinates of the last communication. Basically, a POC map is a probability distribution over a grid, that represents the expectancy of finding the victim on each cell. An example of a POC map is presented in Figure 4.2a. In this figure, those cells filled with a darker blue tone, possess a higher POC value. Conversely, those cells filled with a lighter blue tone, possess a lower POC value. Considering this convention, a natural intuition about how to perform a search based on a POC map may be to prioritize explorations over darker cells and then advance exploring thorough lighter cells. Figure 4.2b illustrates the concept of a search mission based on a POC map.

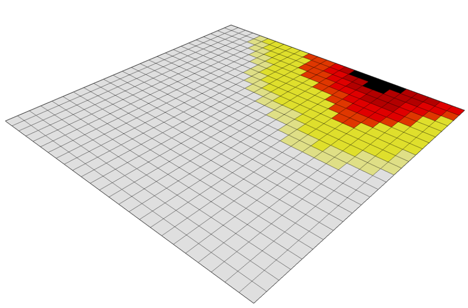


(a) POC map representation over a grid **(b)** UAVs exploring based on POC map

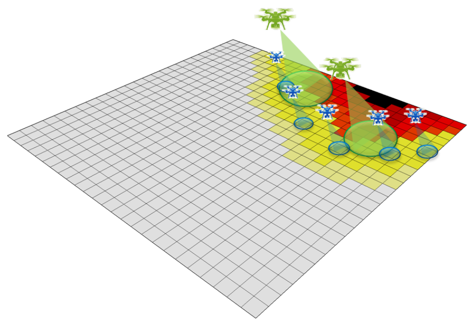
Figure 4.2: POC on search Theory

An important portion of the sample in our literature review, adopted the POC map as the unique criterion to structure the search plan. Apart from the POC map, we only identified one element considered by reviewed papers as a criterion to decide where to search. Such an element is the transition matrix, which allows to account for the potential displacement of the target through the search region.

In this thesis, we propose to consider a new element to be taken into account as a criterion to lead the search mission. Such element, mentioned in Section 4.1.1 and illustrated in Figure 4.1 is referred to in this thesis as *Danger map* (DA map). Here we define the DA map as a distribution over the search region of the danger caused by the disaster. The danger at a given location in turn can be associated with the closeness between the current position of the disaster and such location, the probability of disaster arriving there within a given horizon or alternatively, with the severity of the damage that the disaster could cause if it reaches that point of the search region. For clarity, we present example of a DA map for a search mission during a wildfire in Figure 4.3a.



(a) Wildfire representation over a grid



(b) UAVs exploring based on danger (wildfire)

Figure 4.3: Search in presence of a wildfire

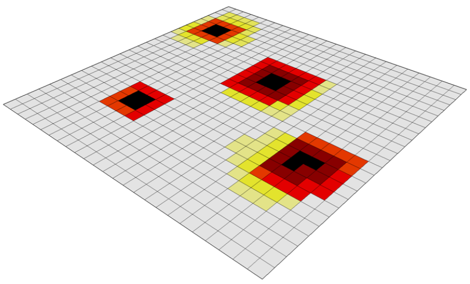
The black colored cells in Figure 4.3a represent burned areas. Those are cells that the wildfire have destroyed yet. By their side, those cells filled with a dark red color, represent locations that are currently fired and where the wildfire presents a high destructive power. Cells with a lighter red fill represent currently firing areas where the wildfire present a low destructive power. In this thesis we assume that the chances of finding the victim alive in any of the three types of cells mentioned before, is zero. Consequently, it does not have any sense to search for a victim in such cells. The search efforts should be focused on yellow filled cells. Those cells filled with a dark yellow color represent locations with a high DA coefficient and the remaining light yellow colored cells represent locations with a medium DA coefficient. All yellow filled cells are locations that the wildfire have not reached yet but there is a high risk of being devastated in a forthcoming time window. In our methodology, DA coefficients might be obtained through the following two steps:

1. Recollection of relevant information of current state of the disaster that could be used to forecast its future state.

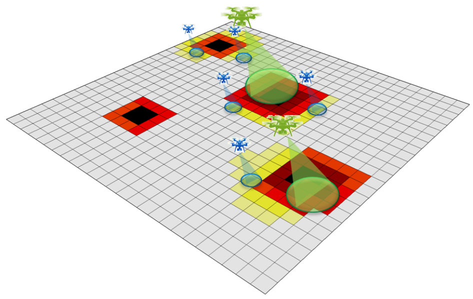
2. Forecasting of future state of the disaster.

Such a procedure can be applied to different types of non-instantaneous disasters. In the case of a search mission based on a DA map, the natural intuition is to prioritize explorations over darker yellow cells, and then advance exploring thorough lighter yellow cells. We provide an example of such kind of mission in Figure 4.3b. In that figure, the green UAVs compose the monitoring system, responsible of tracking the wildfire and the blue UAVs conform the exploration system responsible of searching for victims.

The distribution of shapes over a DA map depends on the specific type of disaster considered. The shapes in Figure 4.3a where localized over a defined section of the search region. This distribution represents the typical behavior of a wildfire, with an initial central point and the fire expanding around it. In contrast, a DA map for a post-earthquake scenario might contain multiple scattered center points associated with damaged structures that present risk of collapse. We present a DA map for such scenario in Figure 4.4a. The corresponding search mission based on the DA map is presented in Figure 4.4b. In both figures, the green UAVs represent the monitoring system and the blue ones represent the exploration system.



(a) Earthquake representation over a grid

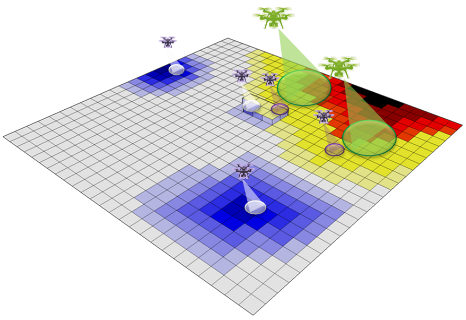


(b) UAVs exploration based on danger (earthquake)

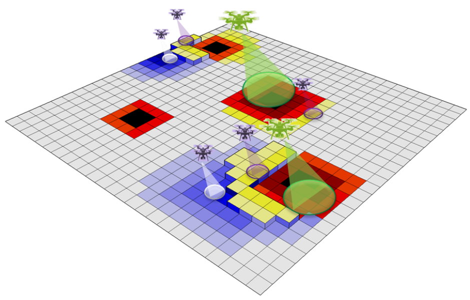
Figure 4.4: Search after an earthquake

The combined adoption of the POC map and the DA map provides en alternative for structuring search plans in presence of non-instantaneous disasters. The POC map indicates the locations where it is more probable to find the victim. By its side, the DA map highlights the most risky locations of the search region for a near future horizon.

The concept of a search mission based on the combined adoption of the POC and the DA maps as planning criteria is illustrated in Figures 4.5a and 4.5b for the both, the wildfire and post-earthquake scenarios considered before.



(a) UAVs exploring over a POC grid during a wildfire



(b) UAVs exploring over a POC grid after an earthquake

Figure 4.5: Search based on POC combined with DA

As shown in Figures 4.5a and 4.5b, a search mission based on a combination of POC and AD maps, will be characterized by a distribution of the search agents over most likely and most risky zones. Cells with a high coefficient in both features will be the most attractive locations to be explored.

4.1.3 Receding horizon for dynamic planning

As we have been explaining in previous sections, the disasters considered in this thesis are dynamic threats that evolve during the search mission. As a direct consequence, a suitable search methodology for such a scenario, is required to be adaptive. In Figure 4.6, we illustrate the receding horizon scheme that we propose in this thesis, aiming to set up our methodology with the ability to adapt to disaster's behavior. The receding horizon scheme shown in Figure 4.6, is composed of four main processes, named *monitoring*, *forecasting*, *path planning* and *exploration*. Those processes are executed by the monitoring, processing and exploration systems, described in Section 4.1.1.

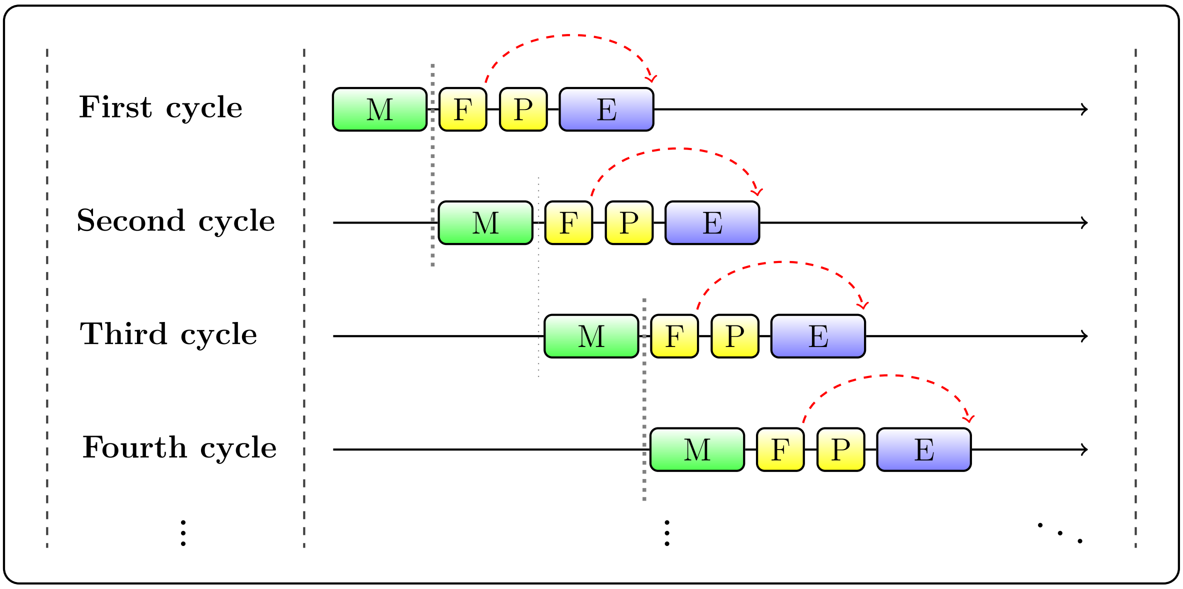


Figure 4.6: Receding horizon scheme

The monitoring process involves the continuous tracking of the disaster and the collection of relevant information about its current state that could be used to forecast its future behavior. This task is responsibility of the monitoring system. The determination of the mechanism for collecting the data is a crucial task in this process. It represents by itself an engineering problem that is currently being subject of research. Some examples of monitoring systems for disasters can be found in (Hirokawa, Kubo, Suzuki, Meguro, & Suzuki, 2007; Adams & Friedland, 2011; Baiocchi, Dominici, Milone, & Mormile, 2013). In this thesis, we will not consider the sampling process required to achieve updated information about disaster's behavior. In contrast, our experiments assume that the monitoring system has access to real time information about the state of the disaster all over search region.

The forecasting process consists of the estimation of relevant features about the future state of the disaster. Disasters are complex phenomenons whose forecast often requires the computation of multiple formulations and the solution of several mathematical models. For that reason, a good alternative to carry on such estimation may be via computational simulation. Such an approach is implemented in this thesis. In order to emulate real life missions, we implemented two types of disaster simulations. The first type of simulation is a single run of the fire spread model representing the real behavior of the disaster. By its side, the second type of simulation will be composed by multiple runs of the fire spread model, which accounts for the forecast of disaster's behavior.

The path planning process consists on the solution of our search planning model via one of our solution methods. Finally, the exploration process deals with the the execution of the plan provided by the processing system and the updating of the POC map.

Those four processes are executed multiple times in an iterative fashion during the search mission. The mission starts with a preliminary track of the disaster. When such exploration is ended, the information collected by the monitoring system is shared with the processing system via the communication system. Then, a computerized module executes the forecasting process and uses the outputs of the simulation as inputs of the search planning model. The solution of the model is then shared with the exploration system via the communication system and the search team executes the plan. In order to maintain a continuous functioning of the exploration system during the whole mission, the last forecast should remain valid at least until the search plan for the next cycle is delivered by the processing system. The red dotted arrow in Figure 4.6 illustrates that requirement. In a real search mission, the forecast horizon may be extended in order to provide the protocol with time buffers in case that one of the three first processes suffers a failure.

The dotted line in Figure 4.6 indicates that each cycle must start after the end of the monitoring process of the previous cycle. In consequence, it is expected that both, the monitoring system and the exploration system remain active during the whole mission. The addition or removal of an agent during the mission will be possible at any moment within a given cycle before the solution of the search planning model for the next time window.

The rescue system, will be waiting at every moment of the mission for the advice of the exploration system, via the communication system, to aid and rescue marked victims.

4.2 Formulations

4.2.1 Problem definition

Let us consider a single missing entity reported lost on a large open region hit by a non-instantaneous natural disaster. The area of interest is discretized in a grid composed by n cells, reflecting natural partitioning of the region or virtual sectioning defined by search agencies in order to aid analytical methods and assign specific tasks to agents through the mission (Chung et al., 2009). A team of heterogeneous agents $K = \{k_1, \ldots, k_m\}$ is deployed over the search region with the objective of identifying the location of the missing entity. In order to represent possible displacements among cells, we adopt a complete undirected graph $G\{I, A\}$, connecting cells $I = \{i_1, \ldots, i_n\}$ through a set of edges $A = \{(i, j) / i, j \in I\}$. The edge i, j is supposed to be the shortest path connecting cells i and j, while avoiding any obstacle in the way. Sensing reliabilities α_k, exploration times O_k and travel times T_k are considered as sources of heterogeneity, meaning that they are all agent-dependent properties.

At the beginning of the mission, there is available a POC map[1] resulting from the exploitation of available data on the last known position and expected trip of the entity. Such a POC map provides each cell with a coefficient γ_i, representing the probability of finding the person there. The search horizon is decomposed in a receding horizon fashion, with the purpose of dynamically adapt the search strategy to the behavior of the disaster. At the beginning of each time window of the receding horizon scheme, a forecast for disaster's state is produced. As a result of the forecast, each cell acquires a danger coefficient w_i that will remain constant for the next time window, but may change at future moments of the mission.

At each time window, all agents depart from their last assigned position in the previous time window; in the case of the first time window, the manager is allowed to specify the initial position of the agents. We have the following assumptions:

- The triangle inequality is satisfied so that $t_{iv} + t_{vj} \geq t_{ij}$,
- There is no possibility of collision while agents travel among cells,
- Agents are false-alarm free, i.e. they will never announce a finding in a cell given that the entity is not there,
- All agents has sufficient resources available (e.g. fuel, energy) for conducting the largest possible search sequence on a time window of length W,
- The communication system that trades information among agents and the planning module is perfect and imperturbable,
- There is available information about the state of the disaster at each location of the search region throughout the whole mission.

The problem consists on finding the search sequence of each agent, aiming to maximize the chances of finding the entity safe and opportunely. Such goal is seek by means of the following two objective functions:

- Maximization of the cumulative danger collected by the search team
- Maximization of the cumulative probability of success collected by the search team

The first objective prioritizes explorations in most risky zones, which are those with a higher risk coefficient w_i. The second objective prioritizes explorations in most likely zones by sending most reliable agents to explore cells with the higher probability of containment γ_i.

A feasible solution for the problem stated above must satisfy the following constraints: (i) simultaneous explorations at the same cell must not be performed; (ii) The length of the sequence of each agent should be shorter than the length of each time window W.

Following the decomposition of Search Theory problems, introduced by us in Section 3.1, we call this problem the *Optimal Search Path with Effort Allocation Problem under Dynamic Disaster* (OSPEAD).

[1]See Section 3.1.3 - Task 2 for a full definition of POC map.

4.2.2 Updating the POC map

In this section, we introduce the recursive Bayesian filter that we implement in our mathematical model for updating the POC map. Bayesian filters have been widely implemented in the literature for such purpose, nonetheless, as mentioned in Section 3.1.6 several models in the literature present out of context updating mechanism or does not present any. The majority of models that update the POC map after each exploration are those related to PSC problems[2]. In contrast, the majority of studies that tackle path planning problems, such as the OSP[2] and the OSPEA[2] problems, only update the POC map at the end of each time window. This fact represent a weakness of path planning models because it implies loss of accuracy in the information that supports decision taking by such models. To illustrate this idea, imagine that there are only two cells in the search region, $c1$ and $c2$, where $c1$ has a POC of 0.8 and $c2$ has the remaining 0.2 of POC. Now, imagine that a non-perfect search agent visits $c1$ one time and it does not find the target. It is natural to think that the POC of $c1$ could not remain being 0.8 after this visit due to the fact that now the expectancy of finding the target in $c1$ is lower. If the agent does not update the POC map during a given time window, he will keep comparing 0.2 with 0.8 after each visit in such time window and then it will remain exploring $c1$ during the whole time window.

To mitigate this issue, we developed a Bayesian filter able to update the POC map depending on the number of visits carried out in a given cell. Such a filter allows an accurate computation of the cumulative POS and aids our path planning model to decide how long to remain exploring at each location. Our Bayesian filter is presented below.

Consider the following notation:

- α_k: probability that agent k announces detection given that the target is present in explored cell.
- $p_j^{(f)}$: probability of containment of cell j after f visits.

Then, the conditional probability of containment of cell j after a single visit of agent $k$can be defined as follows:

$$p_j^{(f+1)} = \frac{p_j^{(f)}(1-\alpha_k)}{p_j^{(f)}(1-\alpha_k) + (1-p_j^{(f)})(1-\delta_k)} \tag{4.1}$$

In this thesis, we assume that UAVs are false-alarm free, which means that $\delta = 0$. Replacing this value in 4.1, we obtain the following expression:

$$p_j^{(f+1)} = \frac{p_j^{(f)}(1-\alpha_k)}{1-p_j^{(f)}\alpha_k} \tag{4.2}$$

We can apply the same formulation for any pair of subsequent visits on cell j. For convenience, we present the expressions for $f = 0$, $f = 1$, $f = 2$ and $f = 3$.

$$p_j^{(1)} = \frac{p_j^{(0)}(1-\alpha_k)}{1-p_j^{(0)}\alpha_k} \tag{4.3}$$

[2]See Section 3.1.4 for a full description of the main four types of search problems

$$p_j^{(2)} = \frac{p_j^{(1)}(1-\alpha_k)}{1-p_j^{(1)}\alpha_k} \tag{4.4}$$

$$p_j^{(3)} = \frac{p_j^{(2)}(1-\alpha_k)}{1-p_j^{(2)}\alpha_k} \tag{4.5}$$

$$p_j^{(4)} = \frac{p_j^{(3)}(1-\alpha_k)}{1-p_j^{(3)}\alpha_k} \tag{4.6}$$

We are interested in developing an expression for p_j^f, dependent on the value of p_j^0, that could be implemented for any value of f. It means, that we aim to build a generic Bayesian formulation that be able to provide the updated POC after any number of visits to a given cell. To obtain such expression, we followed these steps:

1. Replace 4.3 in 4.4

$$p_j^{(2)} = \frac{\left[\frac{p_j^{(0)}(1-\alpha_k)}{1-p_j^{(0)}\alpha_k}\right](1-\alpha_k)}{1-\left[\frac{p_j^{(0)}(1-\alpha_k)}{1-p_j^{(0)}\alpha_k}\right]\alpha_k} \tag{4.7}$$

2. Factorize to obtain an expression for p_j^2, dependent on the value of p_j^0

$$p_j^{(2)} = \frac{p_j^{(0)}(1-\alpha_k)^2}{1-p_j^{(0)}\alpha_k\left[1+(1-\alpha_k)\right]} \tag{4.8}$$

3. Replace 4.8 in 4.5

$$p_j^{(3)} = \frac{\left[\frac{p_j^{(0)}(1-\alpha_k)^2}{1-p_j^{(0)}\alpha_k\left[1+(1-\alpha_k)\right]}\right](1-\alpha_k)}{1-\left[\frac{p_j^{(0)}(1-\alpha_k)^2}{1-p_j^{(0)}\alpha_k\left[1+(1-\alpha_k)\right]}\right]\alpha_k} \tag{4.9}$$

4. Factorize to obtain an expression for p_j^3, dependent on the value of p_j^0

$$p_j^{(3)} = \frac{p_j^{(0)}(1-\alpha_k)^3}{1-p_j^{(0)}\alpha_k\left[1+(1-\alpha_k)+(1-\alpha_k)^2\right]} \tag{4.10}$$

5. Replace 4.10 in 4.6

$$p_j^{(4)} = \frac{\left[\dfrac{p_j^{(0)}(1-\alpha_k)^3}{1-p_j^{(0)}\alpha_k\left[1+(1-\alpha_k)+(1-\alpha_k)^2\right]}\right](1-\alpha_k)}{1-\left[\dfrac{p_j^{(0)}(1-\alpha_k)^3}{1-p_j^{(0)}\alpha_k\left[1+(1-\alpha_k)+(1-\alpha_k)^2\right]}\right]\alpha_k} \tag{4.11}$$

6. Factorize to obtain an expression for p_j^4, dependent on the value of p_j^0

$$p_j^{(3)} = \frac{p_j^{(0)}(1-\alpha_k)^3}{1-p_j^{(0)}\alpha_k\left[1+(1-\alpha_k)+(1-\alpha_k)^2+(1-\alpha_k)^3\right]} \tag{4.12}$$

7. Finally, from equations 4.3, 4.8, 4.10 and 4.12 we can deduce the following general expression for $p_j^{(f)}$, dependent on the value of $p_j^{(0)}$

$$p_j^{(f)} = \frac{p_j^{(0)}(1-\alpha_k)^f}{1-p_j^{(0)}\alpha_k\displaystyle\sum_{\ell=0}^{f-1}(1-\alpha_k)^\ell} \tag{4.13}$$

The Bayesian filter proposed in Equation 4.13 is a valid expression for the computation of the updated POC as a conditional probability. However, this filter has the nonlinear nature coming from its Bayesian basis. It is well known that nonlinear problems are intrinsically more difficult to solve than their linear counterparts and the optimum is only warranted for some specific structures of the model. Therefore, in this thesis we conduct a linearization scheme which starts with the removal of the summation present in the denominator of the filter, whose upper limit depends on the number of explorations, which in turn is naturally a variable of search planning models. The expression for the convergence of a truncated geometric series could be useful for this purpose as follows:

$$p_j^{(f)} = \frac{p_j^{(0)}(1-\alpha_k)^f}{1-p_j^{(0)}\alpha_k\left(\dfrac{1-(1-\alpha_k)^f}{1-(1-\alpha_k)}\right)} \tag{4.14}$$

Equation 4.14 can be then simplified to obtain Equation 4.15

$$p_j^{(f)} = \frac{p_j^{(0)}(1-\alpha_k)^f}{1-p_j^{(0)}(1-(1-\alpha_k)^f)} \tag{4.15}$$

Equation 4.15 is fundamental for the computation of one our the objective functions of our path planning model. Further procedures to reach a linear version of our filter are presented in Section 4.2.5.

4.2.2.1 Contribution of our Bayesian recursive filter

In Section 3.1.4.3, we advised that several ST studies, many of them tackling the OSP problem, present missing or out of context mechanisms for POC updating. As an example, we cited the mathematical expression adopted in Berger et al. (2013), which is in fact one of the most common approaches for such purpose. Such expression is rewritten below for the convenience of the reader.

$$p_{c(t+1)} = \sum_{0 \leq l \leq V_c} p_{c(t)} (1 - p_{cc})^l V_{clt} \tag{3.8}$$

Where pcc is the probability on a specific visit to correctly detect the target in cell c; $p_{c(t)}$ is the updated POC of cell c at the beginning of time interval t; and $V_{clt} = 1$ reflects a cumulative number of explorations l in cell c, by the end of time interval t.

As it can be appreciated, Equation 3.8 only fits to the numerator of the recursive Bayesian filter proposed by us in Equation 4.15. The difference between both approaches is evidenced in Figures 4.7 and 4.8, where both expressions were implemented to compute the POS and the cumulative POS, respectively. In both figures, the l represents the number of explorations, the orange curve is the one obtained by means of the Bayesian filter derived in Equation 4.15 and the yellow curve is the one obtained by means of the approximation presented in Equation 3.8.

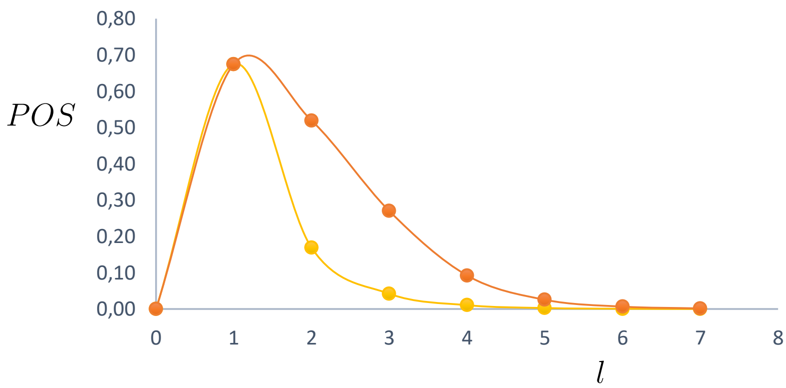


Figure 4.7: Comparison of POS curves

It can be deduced that the approximation underestimates the remaining importance of a cell after being explored and then it leads agents to leave cells prematurely. The Bayesian filter introduced here is suggested by us as a valid updating mechanism for future researches in order to avoid such issue.

4.2.3 Computation of cumulative POS

Let $p_j^{(f)}$ be the probability of containment of cell j after f visits and α_k be the probability that agent k announces detection given that the target is present in explored cell, as in Section 4.2.2. We need to define an expression for the cumulative POS that could be implemented in our path planning model, accounting for the updating of the POC map depending on the number of explorations performed at each location. Let $POS^{(f)}$ be

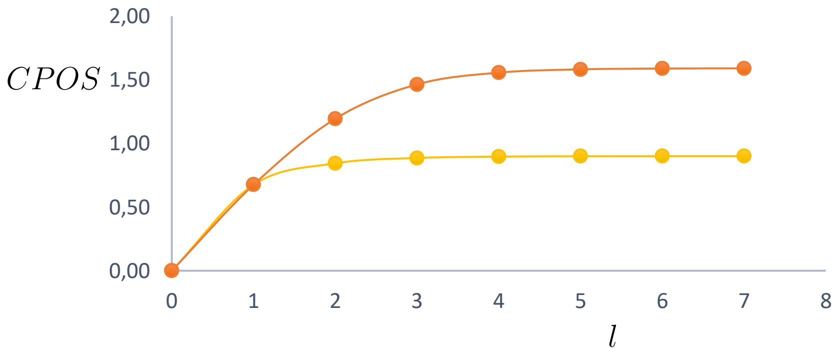


Figure 4.8: Comparison of cumulative POS curves

the cumulative POS after f visits to a given cell. Assuming that the POC and the agent reliability are independent events, the cumulative POS of agent k after a single exploration at cell j, $POS^{(1)}$, can be computed as the product between the POC of cell j and the probability of detection of agent k. The following equation illustrates that computation:

$$POS^{(1)} = p_j^{(0)} \alpha_k \tag{4.16}$$

If agent k performs a new exploration at cell j, the probability of finding the target will be obtained by multiplying the updated POC by the probability of detection of the agent. This computation is illustrated in Equation 4.17.

$$POS^{(2)} = p_j^{(1)} \alpha_k \tag{4.17}$$

In this expression, the updated value of the POC must be obtained by implementing the Bayesian filter introduced in Section 4.2.2. This computation will be analogue for any future exploration. At each exploration, the only term that changes is the POC of the cell that is being explored. One might be tented to implement Equation 4.18 instead of Equation 4.17 for the computation of the POS at the second exploration. Nonetheless, strictly speaking, Equation 4.18 computes the probability of not finding the target in the first exploration but finding it at the second exploration if the POC remains the same after the first exploration. As discussed in a previous section, the POC must not remain the same after an exploration and then, Equation 4.18 become discarded.

$$POS^{(2)} = p_j^{(0)} \alpha_k + p_j^{(0)} (1 - \alpha_k) \alpha_k \tag{4.18}$$

According to Equations 4.16 and 4.17, the the cumulative POS of agent k after two explorations at cell j, can be computed as follows:

$$POS^{(2)} = p_j^{(0)} \alpha_k + p_j^{(1)} \alpha_k \tag{4.19}$$

This expression can be generalized for f visits at cell j as it is shown in Equation 4.20.

$$POS^{(f)} = \sum_{\ell=0}^{f-1} p_j^{(\ell)} \alpha_k \tag{4.20}$$

Similar results were obtained by Frost (1998). Now that we defined general expressions for the updating of the POC map and for the cumulative POS, we can proceed with the introduction of our optimization model.

4.2.4 Mixed Integer Nonlinear Program (MINLP)

In this section we propose a MINLP mathematical formulation for the Optimal Search Path with Effort Allocation Problem under Dynamic Disaster (OSPEAD), formally defined in Section 4.2.1. The notation for the model is presented in Table 4.1. A thorough explanation is subsequently made.

Table 4.1: Notation of the Mixed Integer Nonlinear Program

Sets		
n		Total number of cells, with set $I = \{1, \ldots, n\}$
a		Total number of agents, with set $K = \{1, \ldots, a\}$
Mr_k		Maximum number of visits allowed to agent k, with set $R = \{1, \ldots, Mr_k\}$
Mf_k		Maximum number of consecutive exploration allowed to agent k on the same visit, with set $F = \{1, \ldots, Mf_k\}$
Region and horizon parameters		
γ_i	$\in \mathbb{R}^{[0,1]}$	Probability of containment of cell i at the beginning of the time window
w_i	$\in \{0, 1\}$	Risk coefficient of cell i
W	$\in \mathbb{R}^{\geq 0}$	Length of the time window
Agent parameters		
α_k	$\in \{0, 1\}$	Sensing reliability of agent k
O_k	$\in \mathbb{R}^{\geq 0}$	Exploration time of agent k
t_{ij}^k	$\in \mathbb{R}^{\geq 0}$	Time required by agent k to cover the link ij
b_i^k	$\in \mathbb{R}^{\geq 0}$	Earliest moment at which agent k is able to visit cell i for first time, given its initial position
Big-Ms		
M_T	$\in \mathbb{R}^{\geq 0}$	Sufficiently large value of time
M_P	$\in \{0, 1\}$	Sufficiently large value of probability
Decision variables		
Y_{ir}^k	$\in \{0, 1\}$	1 if agent k is assigned to visit cell i for r-th time; 0 otherwise

Continued on next page

X_{ir}^k	$\in \mathbb{Z}^{\geq 0}$	Number of scans performed by agent k on its r-th visit to cell i
V_{fjr}^k	$\in \{0,1\}$	1 if agent k performs f scans at its r-th visit to cell j; 0 otherwise
T_{ir}^k	$\in \mathbb{R}^{\geq 0}$	Arrival time of agent k, being its $r-th$ visit to cell i
Φ_{ir}^k	$\in \{0,1\}$	Probability of containment found by agent k on its r-th visit to cell i
P_{ir}^k	$\in \{0,1\}$	Probability of success collected by agent k on its r-th visit to cell i
$S_{ijrr'}^k$	$\in \{0,1\}$	1 if agent $k \in K$ visits cell $i \in I$ for $r-th$ time before it visits cell $j \in I$ for $r'-th$ time; 0 otherwise
$B_{irr'}^{kd}$	$\in \{0,1\}$	1 if agent $k \in K$ visits cell $i \in I$ for $r-th$ time just before agent $d \in K$ visits such cell for $r'-th$ time; 0 otherwise
$H_{irr'}^{kd}$	$\in \{0,1\}$	1 if agent $k \in K$ visits cell $i \in I$ for $r-th$ time at any moment before agent $d \in K$ visits such cell for $r'-th$ time; 0 otherwise

The mathematical model can be written as follows:

$$\underset{X_{ir}^k}{\text{Max}} \quad CumDanger = \sum_k \sum_i \sum_r w_i X_{ir}^k \tag{O1}$$

$$\underset{P_{ir}^k}{\text{Max}} \quad CumPOS = \sum_k \sum_i \sum_r \alpha_k P_{ir}^k \tag{O2}$$

s. t.

Routing

$$X_{ir}^k = \sum_k f V_{fir}^k, \qquad \forall i, \forall r, \forall k, \tag{C1}$$

$$Y_{ir}^k = \sum_k V_{fir}^k, \qquad \forall i, \forall r, \forall k, \tag{C2}$$

$$\sum_f E_{fir}^k \leq 1, \qquad \forall i, \forall r, \forall k, \tag{C3}$$

$$Y_{ir}^k \leq Y_{i(r-1)}^k, \qquad \forall i, \forall r, \forall k : r{>}1, \tag{C4}$$

$$T_{ir}^k \geq T_{i(r-1)}^k + O_k - M_T(2 - Y_{ir}^k - Y_{i(r-1)}^k), \qquad \forall i, \forall r, \forall k : r{>}1, \tag{C5}$$

$$T_{ir}^k \geq b_i^k Y_{ir}^k, \qquad \forall i, \forall r, \forall k, \tag{C6}$$

$$T_{ir}^k \leq W Y_{ir}^k, \qquad \forall i, \forall r, \forall k, \tag{C7}$$

$$T^k_{jr'} \geq T^k_{ir} + \left(t^k_{ii} + O_k\right) X^k_{ir} + t^k_{ij} - M_T\left(1 - S^k_{ijrr'}\right) - M_T\left(2 - Y^k_{ir} - Y^k_{jr'}\right), \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \quad \text{(C8)}$$

$$S^k_{iirr} = 0, \qquad \forall i, \forall r, \forall k, \quad \text{(C9a)}$$

$$S^k_{ijrr'} + S^k_{jir'r} \geq Y^k_{ir} + Y^k_{jr'} - 1, \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \quad \text{(C9b)}$$

$$S^k_{ijrr'} + S^k_{jir'r} \leq Y^k_{ir}, \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \quad \text{(C9c)}$$

$$S^k_{ijrr'} + S^k_{jir'r} \leq Y^k_{jr'}, \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \quad \text{(C9d)}$$

$$W \geq T^k_{ir} + \left(t^k_{ii} + O_k\right) X^k_{ir}, \qquad \forall i, \forall r, \forall k, \quad \text{(C10)}$$

Scheduling

$$T^d_{ir'} - T^k_{ir} \geq O_k X^k_{ir} - M_T\left(1 - H^{kd}_{irr'}\right) - M_T\left(2 - Y^k_{ir} - Y^d_{ir'}\right), \qquad \forall i, \forall r, \forall r', \forall k, \forall d, \quad \text{(C11)}$$

$$T^m_{ir''} \geq T^d_{ir'} + O_d - M_T\left(2 - H^{km}_{irr''} - B^{kd}_{irr'}\right), \qquad \forall i, \forall r, \forall r', \forall r'', \forall k, \quad \text{(C12)}$$

$$B^{kd}_{irr'} \leq H^{kd}_{irr'}, \qquad \forall i, \forall r, \forall r', \forall k, \forall d, \quad \text{(C13)}$$

$$\sum_{\ell=1}^{m} \sum_{v=1}^{Mr} B^{k\ell}_{irv} \geq H^{kd}_{irr'}, \qquad \forall i, \forall r, \forall r', \forall k, \forall d, \quad \text{(C14)}$$

$$\sum_{\ell=1}^{m} \sum_{v=1}^{Mr} B^{\ell d}_{ivr'} \geq H^{kd}_{irr'}, \qquad \forall i, \forall r, \forall r', \forall k, \forall d, \quad \text{(C15)}$$

$$\sum_{k} \sum_{r} B^{kd}_{irr'} \leq 1, \qquad \forall i, \forall r', \forall d, \quad \text{(C16)}$$

$$\sum_{d} \sum_{r'} B^{kd}_{irr'} \leq 1, \qquad \forall i, \forall r, \forall k, \quad \text{(C17)}$$

$$H^{kk}_{irr} = 0, \qquad \forall i, \forall r, \forall k, \quad \text{(C18a)}$$

$$H^{kd}_{irr'} + H^{dk}_{ir'r} \geq Y^k_{ir} + Y^d_{jr'} - 1, \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \forall d, \quad \text{(C18b)}$$

$$H^{kd}_{irr'} + H^{dk}_{ir'r} \leq Y^k_{ir}, \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \forall d, \quad \text{(C18c)}$$

$$H^{kd}_{irr'} + H^{dk}_{ir'r} \leq Y^d_{jr'}, \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \forall d, \quad \text{(C18d)}$$

POC updating and POS collecting

$$P^k_{ir} \leq \sum_{\lambda=0}^{f-1} \left(\frac{\Phi^k_{ir}(1-\alpha_k)^\lambda}{1 - \Phi^k_{ir}(1-(1-\alpha_k)^\lambda)} \right) + M_P(1 - E^k_{fir}), \qquad \forall i, \forall r, \forall k, \forall f, \quad \text{(C19)}$$

$$P^k_{ir} \leq M_P Y^k_{ir}, \qquad \forall i, \forall r, \forall k, \quad \text{(C20)}$$

$$\Phi^k_{ir'} \leq \frac{\Phi^d_{ir}(1-\alpha_d)^f}{1 - \Phi^d_{ir}(1-(1-\alpha_d)^f)} + (2 - B^{dk}_{ir'r} - V^d_{fir'}), \qquad \forall i, \forall r, \forall r', \forall k, \forall d, \forall f, \quad \text{(C21)}$$

$$\Phi^k_{ir} \leq \gamma_i Y^k_{ir}, \qquad \forall i, \forall r, \forall k, \quad \text{(C22)}$$

Domain

$$T^k_{ir},\ \Phi^k_{ir},\ P^k_{ir} \in \mathbb{R}^{\geq 0}, \qquad \forall i, \forall r, \forall k, \quad \text{(C23)}$$

$$V^k_{fir} \in \{0, 1\}, \qquad \forall f, \forall i, \forall r, \forall k, \quad \text{(C24)}$$

$$S^k_{ijrr'} \in \{0, 1\}, \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \quad \text{(C25)}$$

$B_{irr'}^{kd},\ H_{irr'}^{kd} \in \{0,1\}, \qquad \forall i, \forall r, \forall r', \forall k, \forall d, \quad \text{(C26)}$

$X_{ir}^{k} \in \mathbb{Z}^{\geq 0}, \qquad \forall i, \forall r, \forall k, \quad \text{(C27)}$

4.2.4.1 Explanation of the formulation

Now that we have formally defined our model, we provide an explanation on its functioning.

Objective Functions

The Objective function O1 maximizes the cumulative danger, collected by all agents during upcoming time window. This function will lead agents to prioritize visits to most dangerous cells[3].

Objective function O2 maximizes the cumulative probability of success (POS) collected by all agents along their search sequences. This function is an extension of Equation 4.20 derived in Section 4.2.3 for multiple agents, multiple cells and and multiple visits. Both Expressions are rewritten below for convenience of the reader.

$$\underset{P_{ir}^{k}}{\text{Max}} \ \ CumPOS = \sum_{k}\sum_{i}\sum_{r} \boxed{\alpha_k}\ \boxed{P_{ir}^{k}} \tag{O2}$$

$$POS^{(f)} = \boxed{\alpha_k}\ \boxed{\sum_{\ell=0}^{f-1} p_j^{(\ell)}} \tag{4.20}$$

As it can be inferred, variable P_{ir}^{k} in the Objective function O2 represents the probability of containment (POC) collected by agent k on its r-th visit to cell i, which is then multiplied by the reliability of the agent α_k to obtain the cumulative POS. The value of P_{ir}^{k} is determined by means of the recursive Bayesian filter developed in Section 4.2.3, which was implemented in Constraint C19. There, the variable E_{fjr}^{k} is responsible of activating the constraint for the specific number of explorations performed by agent the agent.

Constraints

Constraints C1, C2 and C3 determine the number of explorations performed by agent k on its r-th visit to cell i. Constraint C1 ensures that the binary variable E_{fir}^{k} takes a value of 1 if and only if the subscript f is equal to the number of visits assigned to agent k on its r-th arrival to cell i. Constraint C2 links the number of explorations with the visited locations. By its side, Constraint C3 enforces the number of explorations assigned to agent k on its r-th visit to cell i to be unique.

Constraint C4 states that a visit r to cell i in the sequence of agent k can only be assigned if the visit r-1 was also assigned. Coherently, Constraint C5 states that visit r

[3] See Section 4.1.2 for a full definition of the facts that makes a zone to become dangerous in this thesis.

of agent k to cell i must be performed after visit r-1 of such agent to that cell. Originally, this constraint is oriented to enforce that $T^k_{ir} > T^k_{i(r-1)}$ if both, Y^k_{ir} and $Y^k_{i(r-1)}$ are equal to one. Nonetheless, strict inequalities are not treated in linear programming, since the solution is not guaranteed to exist on corner points. Then, we added the value o_k to the right side in order to change which allowed us to change the strict inequality to a greater or equal one.

Constraint C6 determines the earliest time at which agent k is able to visit cell i for first time. This constraint assumes that the agent is ready to launch from its initial position at time 0 of the planning horizon. Otherwise, the set-up time should be added to b^k_i. Constraint C8 determines the exploration sequence for agent k, reflected on the arrival times T^k_{ir}. Constraints C9b, C9c and C9d establish the relation between sequencing variables S and assignment variables Y. According to such constraints, $S^k_{ijrr'}$ and $S^k_{jir'r}$ can only take a positive value if both, the r-th visit on cell i and the r'-th visit on cell j are assigned to the sequence of agent k. If at least one of those assignments is not done, both sequencing variables are fixed to be zero. Finally, Constraint C10 determines the maximum length of any sequence.

Constraint C11 avoids simultaneous explorations of different agents to the same cell. The transitivity among predecessors at a given cell is ensured by Constraint C12. It states that if agent k in visits cell i for r-th time just before agent d visits that cell for r'-th time and agent k in visits cell i for r-th time at any time before agent m visits that cell for r''-th time, then the arrival time of agent m on its r''-th visit to cell i must be greater than the arrival time of agent d on its r'-th visit to such cell. An illustrative example is presented in figure 4.9. There, the three agents k, d and m are assigned to visit the yellow cell in that order. The solid line represent the immediate precedence of agent k over agent d, while the dotted line represent the non-immediate precedence of agent k over agent m, implying that d must enter on the cell before m.

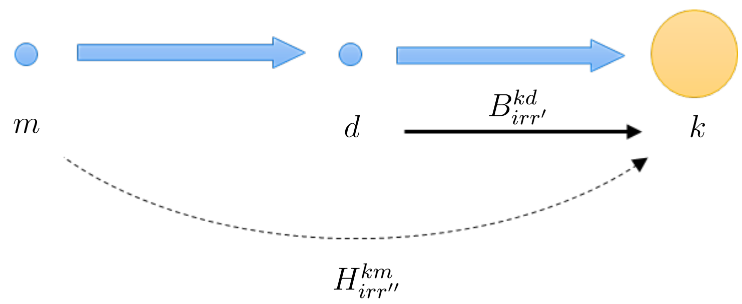


Figure 4.9: Transitivity constraint

Constraint C13 states that every immediate predecessor should also be a general immediate predecessor. Constraints C14 and C15 imply that every general predecessor should also be an immediate predecessor and every general successor should also be an immediate successor, respectively. On the other hand, Constraints C16 and C17 allow an agent to be immediate predecessor and successor of at much a single agent.

Similarly to Constraints C9b- C9d, Constraints C18b - C18d establish the relation between sequencing variables H and assignment variables Y. They state that $H^{kd}_{irr'}$ and $H^{dk}_{ir'r}$ can only take a positive value if both, the r-th visit on cell i and the r'-th visit on cell j are assigned to the sequence of agent k.

Switching Constraints C7, C20 and C22 allows to assign values to arrival time T_{ir}^k, collected probability of containment P_{ir}^k and updated probability of containment Φ_{ir}^k, respectively, only if visit r to cell i is part of the sequence of agent k. If at least one of those assignments is not done, both sequencing variables are fixed to be zero. Constraints C9a and C18a prohibit reflexive predecessor relation within and among sequences, respectively.

Finally, but essential, Constraint C21 determines the updated POC that every agent finds at each cell on its sequence, given previous explorations assigned to other agents. Such update is done by means of the Bayesian filter introduced in section 4.2.2. Constraints C23 to C27 state the domain of the variables.

4.2.4.2 Valid bounds for Mr_k and Mf_k

In previous sections we explained the limitations that time discretization impose to the model, reducing its capacity of describing the dynamics in agents' motion. Our modeling approach uses visits (r) and explorations (f) instead of time instants, allowing agents to move free among any pair of cells at any moment within the time window. Given that both, r and f are indices in our model, it is necessary to define the maximum number of visits that an agent will be allowed to perform within the time window and the maximum number of explorations that an agent will be allowed to perform per visit. Those values will determine the size of the sets R and F according to the notation presented in Table 4.1.

Given that agents may be heterogeneous in their maximum kinematic speed, acceleration and scanning time, it is possible to define agent-dependent bounds Mr and Mf. The steps that we implemented to define valid bounds for both parameters are illustrated below. In the case of Mr_k, we assume that $t_{ij}^k = t_{ji}^k \;\; \forall \; i, j, k$.

Bounds for Mr_k:

1. Select an agent k,
2. Take a cell i as first destination of agent k,
3. Make t_1 the time required for agent k to move from its initial position p_0 to its first destination i, plus its exploration time o_k,
4. Make t_2 the time required for agent k to move from its first destination i to the closest neighbor i', plus its exploration time o_k,
5. Make $M_i = \left\lfloor \dfrac{W - t1}{2 * t2} \right\rfloor + 1$,
6. Make $Mr_k = \text{Max}_i \{M_i\}$.

Bounds for Mf_k:

1. Select an agent k
2. Make t_1 the time required for agent k to prepare for a new scan at cell (if necessary), plus its exploration time o_k

3. Make $Mf_k = \left\lfloor \frac{W}{t1} \right\rfloor$

4.2.4.3 Valid big-M values

Seeking to obtain a linear version of our problem, we implemented the big-M approach at multiple constraints. Theoretically, any sufficiently large value of M will be suitable for our formulation. Nonetheless, it is well-known that very large values of M could cause numerical instability, leading to inaccurate results. In this section we briefly define valid values for Ms in our model.

Valid M_T value. This parameter is present in the following set of constraints:

$$T_{ir}^k \geq T_{i(r-1)}^k + O_k - M_T(2 - Y_{ir}^k - Y_{i(r-1)}^k), \tag{C5}$$

$$T_{jr'}^k \geq T_{ir}^k + \left(t_{ii}^k + O_k\right) X_{ir}^k + t_{ij}^k - M_T\left(1 - S_{ijrr'}^k\right) - M_T\left(2 - Y_{ir}^k - Y_{jr'}^k\right), \tag{C8}$$

$$T_{ir'}^d - T_{ir}^k \geq O_k X_{ir}^k - M_T\left(1 - H_{irr'}^{kd}\right) - M_T\left(2 - Y_{ir}^k - Y_{ir'}^d\right), \tag{C11}$$

$$T_{ir''}^m \geq T_{ir'}^d + O_d - M_T\left(2 - H_{irr''}^{km} - B_{irr'}^{kd}\right), \tag{C12}$$

In all cases M_T is used to inactivate the constraint when it is necessary. The large number that must be canceled by M_T could be $T_{ir}^k + O_k$. We know that T_{ir}^k is bounded by W from Constraint C7. Taking this into account, we defined M_T as follows:

$M_T = W + MaxO$,
Where $MaxO = \mathrm{Max}_k\,\{O_k\}$.

Valid M_P value. By its side, M_P is present in the following two constraints:

$$P_{ir}^k \leq \sum_{\lambda=0}^{f-1} \left(\frac{\Phi_{ir}^k (1-\alpha_k)^\lambda}{1 - \Phi_{ir}^k (1 - (1-\alpha_k)^\lambda)} \right) + M_P(1 - E_{fir}^k), \tag{C19}$$

$$P_{ir}^k \leq M_P Y_{ir}^k, \tag{C20}$$

In this case we must search for a value sufficiently large to allow real feasible POC accumulation. Given that Mf_k represents the largest number that f could take, a valid bound for M_P could be found based on the following procedure:

1. Select a pair agent - cell (k, i),
2. Make Tmp_{ik} the value of cumulative POC according to Constraint C19, assuming that agent k is the first agent that arrives to cell i, finding its POC as γ_i, and it performs Mf_k scans there,
3. Make $M_P = \mathrm{Max}_{ik}\,\{Tmp_{ik}\}$.

4.2.5 Piecewise linear approximation

The optimization program provided above was built based on the formulations proposed by Dondo and Cerdá (2007), Wex, Schryen, Feuerriegel, and Neumann (2014) and Berger et al. (2013) for the MDVRPTW[4] with heterogeneous vehicles, the RUASP[5] and the OSP[6], respectively. The resulting model is majorly linear, which is an advantage in terms of tractability. Unfortunately, Constraints C19 and C21 remain nonlinear since both of them implement the recursive Bayesian filter derived in Section 4.2.2, which is intrinsically nonlinear. However, modern solvers have relatively efficient methods for Mixed Integer Linear Programs (MILPS) in terms of accuracy and solution time (Borghetti, D'Ambrosio, Lodi, & Martello, 2008). Therefore, we decided to implement a linearization scheme in order to obtain a MILP version of our OSPEAD model. Let us first define a property of our model that will support our linearization approach.

> **Separability:** Separable functions are those that can be decomposed into sums of single decision variables, and separable programs are nonlinear programs over separable objective functions and constraints (Rardin, 1998). Those types of problems have much similarity to linear programs, where functions are always scalar multiples of the decision variables. Therefore, the separability of a nonlinear program is a desired property that offers the possibility of building an approximate linear version of the original NLP by means of piecewise linear approximation. Both nonlinear constraints in our model are separable, since each term of the Bayesian filter only depends on Φ_{ir}^k, and remaining terms in both equations are linear.

The expression to be approximated corresponds to our recursive Bayesian filter as a function of the updated POC, taken α_k and f as constants. Equation 4.21 illustrates such function; for simplicity, in this section we will call it G for both constraints.

$$G\left(\Phi_{ir}^k\right) = \frac{\Phi_{ir'}^d(1-\alpha_d)^f}{1-\Phi_{ir'}^d(1-(1-\alpha_d)^f)} \tag{4.21}$$

Piecewise approximation consists in dividing the domain of Φ_{ir}^k into a collection of intervals c and interpolate linearly to approximate the value of G. Following such procedure, Equation 4.21 can be approximated according to Expression 4.22.

$$G\left(\Phi_{ir}^k\right) \approx \widehat{G}\left(\Phi_{irc}^k\right) = \sum_c m_{\lambda c}^k \Phi_{irc}^k \tag{4.22}$$

where $m_{\lambda c}^k$ is the slope of the piecewise approximation in the segment c. The value of the original variables can be computed as the sum of corresponding new variables over all the segments. The approximation scheme is almost complete by adding the following set of constraints to the model:

$$0 \leq \Phi irc^k \leq z_c^k - z_{(c-1)}^k \tag{4.23}$$

[4] Multi Depot Vehicle Routing Problem with Time Windows.
[5] Rescue Unit Assignment and Scheduling Problem.
[6] Optimal Search Path Problem.

where z_c are the interval breakpoints for variable Φirc^k ($z_0^k = 0$).

Resulting linear program gives a correct representation of the original problem, whenever the solution satisfy Condition 4.24. It states that the variable corresponding to a given segment should take its maximum value in order to allow the variable in the subsequent segment to be positive.

$$\Phi^k_{ir(c+1)} > 0 \qquad \text{only if} \qquad \Phi^k_{irc} = (z_c - z_{(c-1)}), \qquad \forall i, r, k, c \tag{4.24}$$
$$\text{whith } z_0 = 0$$

Note that slopes m can be defined independently of the cell i and the visit r indices. Figure 4.10 illustrates the approximation scheme for two dummy curves with parameters $(\alpha_k, f) = (0.9, 1)$ and $(\alpha_k, f) = (0.5, 1)$, being the lower one, that corresponding to the greater α_k. It is evident that the precision of the approximation in that example is not quite good, specially for higher values of α_k. It can be easily noted that the accuracy of piecewise approximation can be enhanced by increasing the number of breakpoints or by defining better values for the same number of breakpoints. The first approach seems attractive since a sufficiently large number of breakpoints will provide any desired level of accuracy. Nonetheless, as explained above, piecewise approximation requires the introduction of a new set of variables, each one representing the value of the original decision variable on a given interval of the domain. If we take q as the number of segments in our approximation, it will be necessary to add $i * r * k * q$ new variables to the model. Based on it we should proceed carefully with the first way of improving piecewise approximation accuracy.

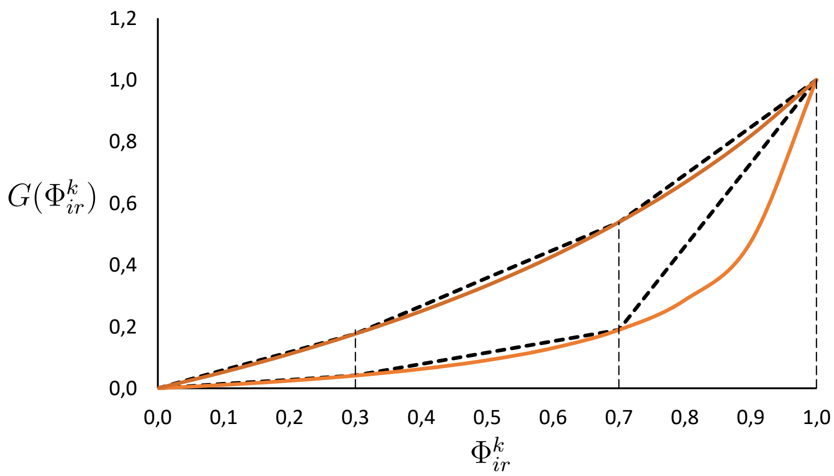


Figure 4.10: Accuracy of piecewise linear approximation

The second approach consists on finding good values for the same number of breakpoints. However, for some types of problems it could be difficult to identify suitable breakpoints, specially when the domains of decision variables are not naturally upper bounded. In those cases, the analyst may not be able to figure out the shape of the curve until the model has been solved. In our case, the decision variable Φ^k_{ir} is a probability and then it ranges in the interval $[0, 1]$. To decide how many and where to add breakpoints, we plotted $G(\Phi_{ir}^k)$ for several combinations of α_k and f, searching for common inflection points. Resulting plot is shown in Figure 4.11.

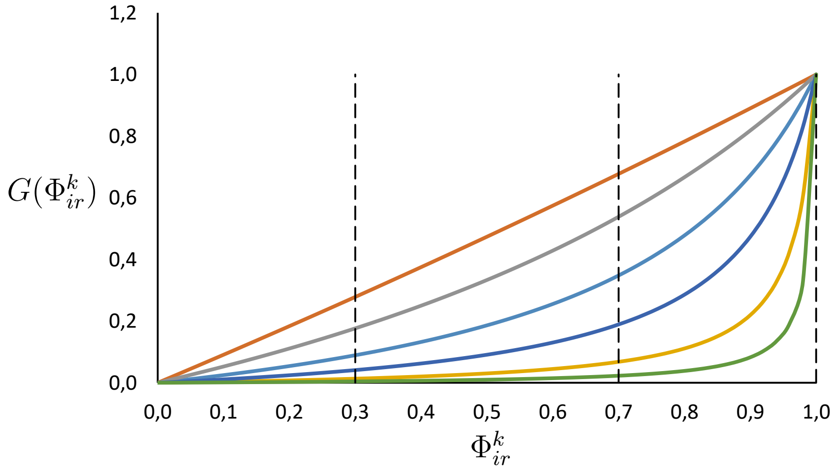


Figure 4.11: Plot sectioned by slope behavior

Based on the slopes of the curves in Figure 4.11, we can identify three main relevant sections. The region on the right side of the plot $(0.7 - 1.0)$, where the inflection point is often located; the middle region, which has no sharp curves but still being clearly nonlinear $(0.3 - 0.7)$; and finally the left side region, where slopes tend to be constant $(0.0 - 0.3)$. Based on this behavior, we decided to work with the set of breakpoints $Z = \{0.3, 0.5, 0.7, 0.8, 0.9, 0.96, 1\}$. Resulting approximation for the example provided above is illustrated in Figure 4.12 and its goodness of fit is tested in Table 4.2.

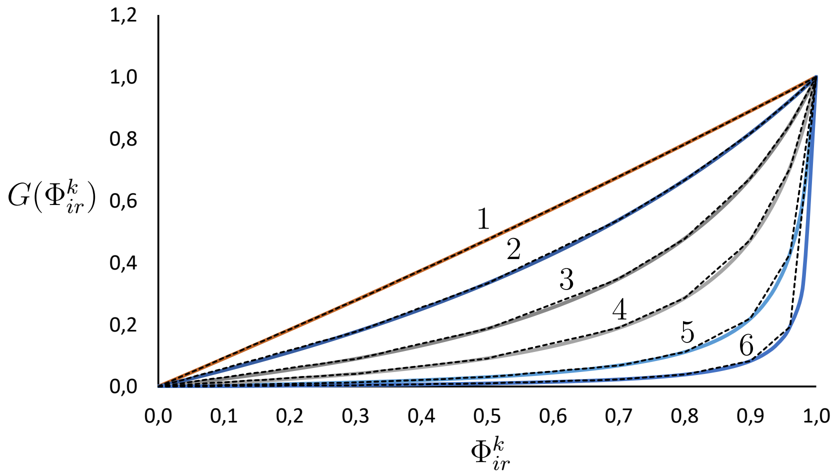


Figure 4.12: Curve fitting reached

It is evident from Figure 4.12 and Table 4.2 that our approximation has a very good fit to the original function. Even the worst Mean Standard Error, the one corresponding to curve 6, is relatively small in comparison with the values that needs to be calculated by the expression, which are probabilities. It can be also appreciated from Figure 4.12 that the greatest deviations in that curve occur at values of Φ_{ir}^k higher than 0.95. However, real life missions are commonly subject to high uncertainty and such high values for the POC of a given cell are unlikely. In contrast, the approximation is much better for mid and low values of Φ_{ir}^k, which are more common.

Table 4.2: Goodness of fit of the approximation

Curve	α_k	f	Error (MSE)
1	0.1	1	9,45E-07
2	0.1	14	3,28E-05
3	0.5	1	1,72E-05
4	0.5	5	2,80E-04
5	0.9	1	5,37E-05
6	0.9	2	1,42E-03

There is still a last thing to consider before finishing the approximation process. That last thing is the non-convexity of our problem.

> **Non-convexity:** Convex programs are those where: (a) convex objective functions are minimized or (b) concave objective functions are maximized, subject to concave $\geq$ constraints, convex $\leq$ ones and linear equalities (Rardin, 1998). By the other side, piecewise approximations to separable convex programs have an optimal solution satisfying sequencing condition 4.24. Contrarily, non-convex models impose additional difficulties to the application of piecewise linear approximation. In order to proceed with the approach, it is necessary to introduce a set of binary variables and switching constraints that will push the decision variable to its upper bound if the subsequent one is positive.
>
> It is straightforward to determine that our problem is non-convex. The presence of integer decision variables is a sufficient condition to classify a model as non-convex (FrontlineSolvers, n.d.). Consequently, we proceed to define a set of binary decision variables R_{irc}^k and the following set of constraints:

$$(z_c - z_{(c-1)})R_{(c+1)}^k \leq \Phi_{irc}^k \leq (z_c - z_{(c-1)})R_{irc}^k \qquad \forall i, r, k, c \tag{4.25}$$

Now that our linear approximation is completely defined, we proceed to present the linear version of our mathematical model.

4.2.6 Mixed Integer Linear Program (MILP)

Considering the piecewise approximation derived in previous section, the notation of the linear version of our model is presented in Table 4.3.

Table 4.3: Notation of the Mixed Integer Linear Program

Sets		
n		Total number of cells, with set $I = \{1, \dots, n\}$
a		Total number of agents, with set $K = \{1, \dots, a\}$
Mr_k		Maximum number of visits allowed to agent k, with set $R = \{1, \dots, Mr_k\}$
Mf_k		Maximum number of consecutive exploration allowed to agent k on the same visit, with set $F = \{1, \dots, Mf_k\}$
Q		Number of segments in the piecewise linear approximation, with set $C = \{1, \dots, Q\}$

Region and horizon parameters		
γ_i	$\in \mathbb{R}^{[0,1]}$	Probability of containment of cell i at the beginning of the time window
w_i	$\in \{0, 1\}$	Risk coefficient of cell i
W	$\in \mathbb{R}^{\geq 0}$	Length of the time window

Agent parameters		
α_k	$\in \{0, 1\}$	Sensing reliability of agent k
O_k	$\in \mathbb{R}^{\geq 0}$	Exploration time of agent k
t_{ij}^k	$\in \mathbb{R}^{\geq 0}$	Time required by agent k to cover the link ij
b_i^k	$\in \mathbb{R}^{\geq 0}$	Earliest moment at which agent k is able to visit cell i for first time, given its initial position

Piecewise Parameters		
z_c	$\in \mathbb{R}^{\geq 0}$	Breakpoint at segment c
$m_{\lambda c}^k$	$\in \mathbb{R}^{\geq 0}$	Slope of segment c for agent k and f scans

Big-Ms		
M_T	$\in \mathbb{R}^{\geq 0}$	Sufficiently large value of time
M_P	$\in \{0, 1\}$	Sufficiently large value of probability

Decision variables		
Y_{ir}^k	$\in \{0, 1\}$	1 if agent k is assigned to visit cell i for r-th time; 0 otherwise
X_{ir}^k	$\in \mathbb{Z}^{\geq 0}$	Number of scans performed by agent k on its r-th visit to cell i

Continued on next page

V^k_{fjr}	$\in \{0,1\}$	1 if agent k performs f scans at its r-th visit to cell j; 0 otherwise
T^k_{ir}	$\in \mathbb{R}^{\geq 0}$	Arrival time of agent k, being its $r-th$ visit to cell i
Φ^k_{ir}	$\in \{0,1\}$	Probability of containment found by agent k on its r-th visit to cell i
Φ^k_{irc}	$\in \{0,1\}$	Artificial value for Φ^k_{ir} on segment c of the linear approximation
P^k_{ir}	$\in \{0,1\}$	Probability of success collected by agent k on its r-th visit to cell i
$S^k_{ijrr'}$	$\in \{0,1\}$	1 if agent $k \in K$ visits cell $i \in I$ for $r-th$ time before it visits cell $j \in I$ for $r'-th$ time; 0 otherwise
$B^{kd}_{irr'}$	$\in \{0,1\}$	1 if agent $k \in K$ visits cell $i \in I$ for $r-th$ time just before agent $d \in K$ visits such cell for $r'-th$ time; 0 otherwise
$H^{kd}_{irr'}$	$\in \{0,1\}$	1 if agent $k \in K$ visits cell $i \in I$ for $r-th$ time at any moment before agent $d \in K$ visits such cell for $r'-th$ time; 0 otherwise

The linear approximation to our planning model is presented below:

$$\underset{X^k_{ir}}{\text{Max}} \;\; CumDanger = \sum_k \sum_i \sum_r w_i X^k_{ir} \tag{O1}$$

$$\underset{P^k_{ir}}{\text{Max}} \;\; CumPOS = \sum_k \sum_i \sum_r \alpha_k P^k_{ir} \tag{O2}$$

s. t.

Routing

$$X^k_{ir} = \sum_k f V^k_{fir}, \qquad \forall i, \forall r, \forall k, \tag{C1}$$

$$Y^k_{ir} = \sum_k V^k_{fir}, \qquad \forall i, \forall r, \forall k, \tag{C2}$$

$$\sum_f E^k_{fir} \leq 1, \qquad \forall i, \forall r, \forall k, \tag{C3}$$

$$Y^k_{ir} \leq Y^k_{i(r-1)}, \qquad \forall i, \forall r, \forall k : r{>}1, \tag{C4}$$

$$T^k_{ir} \geq T^k_{i(r-1)} + O_k - M_T(2 - Y^k_{ir} - Y^k_{i(r-1)}), \qquad \forall i, \forall r, \forall k : r{>}1, \tag{C5}$$

$$T^k_{ir} \geq b^k_i Y^k_{ir}, \qquad \forall i, \forall r, \forall k, \tag{C6}$$

$$T^k_{ir} \leq W Y^k_{ir}, \qquad \forall i, \forall r, \forall k, \tag{C7}$$

$$T_{jr'}^{k} \geq T_{ir}^{k} + \left(t_{ii}^{k} + O_k\right) X_{ir}^{k} + t_{ij}^{k} - M_T\left(1 - S_{ijrr'}^{k}\right) - M_T\left(2 - Y_{ir}^{k} - Y_{jr'}^{k}\right), \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \quad \text{(C8)}$$

$$S_{iirr}^{k} = 0, \qquad \forall i, \forall r, \forall k, \quad \text{(C9a)}$$

$$S_{ijrr'}^{k} + S_{jir'r}^{k} \geq Y_{ir}^{k} + Y_{jr'}^{k} - 1, \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \quad \text{(C9b)}$$

$$S_{ijrr'}^{k} + S_{jir'r}^{k} \leq Y_{ir}^{k}, \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \quad \text{(C9c)}$$

$$S_{ijrr'}^{k} + S_{jir'r}^{k} \leq Y_{jr'}^{k}, \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \quad \text{(C9d)}$$

$$W \geq T_{ir}^{k} + \left(t_{ii}^{k} + O_k\right) X_{ir}^{k}, \qquad \forall i, \forall r, \forall k, \quad \text{(C10)}$$

Scheduling

$$T_{ir'}^{d} - T_{ir}^{k} \geq O_k X_{ir}^{k} - M_T\left(1 - H_{irr'}^{kd}\right) - M_T\left(2 - Y_{ir}^{k} - Y_{ir'}^{d}\right), \qquad \forall i, \forall r, \forall r', \forall k, \forall d, \quad \text{(C11)}$$

$$T_{ir''}^{m} \geq T_{ir'}^{d} + O_d - M_T\left(2 - H_{irr''}^{km} - B_{irr'}^{kd}\right), \qquad \forall i, \forall r, \forall r', \forall r'', \forall k, \quad \text{(C12)}$$

$$B_{irr'}^{kd} \leq H_{irr'}^{kd}, \qquad \forall i, \forall r, \forall r', \forall k, \forall d, \quad \text{(C13)}$$

$$\sum_{\ell=1}^{m} \sum_{v=1}^{Mr} B_{irv}^{k\ell} \geq H_{irr'}^{kd}, \qquad \forall i, \forall r, \forall r', \forall k, \forall d, \quad \text{(C14)}$$

$$\sum_{\ell=1}^{m} \sum_{v=1}^{Mr} B_{ivr'}^{\ell d} \geq H_{irr'}^{kd}, \qquad \forall i, \forall r, \forall r', \forall k, \forall d, \quad \text{(C15)}$$

$$\sum_{k} \sum_{r} B_{irr'}^{kd} \leq 1, \qquad \forall i, \forall r', \forall d, \quad \text{(C16)}$$

$$\sum_{d} \sum_{r'} B_{irr'}^{kd} \leq 1, \qquad \forall i, \forall r, \forall k, \quad \text{(C17)}$$

$$H_{irr}^{kk} = 0, \qquad \forall i, \forall r, \forall k, \quad \text{(C18a)}$$

$$H_{irr'}^{kd} + H_{ir'r}^{dk} \geq Y_{ir}^{k} + Y_{jr'}^{d} - 1, \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \forall d, \quad \text{(C18b)}$$

$$H_{irr'}^{kd} + H_{ir'r}^{dk} \leq Y_{ir}^{k}, \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \forall d, \quad \text{(C18c)}$$

$$H_{irr'}^{kd} + H_{ir'r}^{dk} \leq Y_{jr'}^{d}, \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \forall d, \quad \text{(C18d)}$$

POC updating and POS collecting

$$P_{ir}^{k} \leq \sum_{\lambda=0}^{f-1} \sum_{c} \left(m_{\lambda c}^{k} \Phi_{irc}^{k}\right) + M_P(1 - E_{fir}^{k}), \qquad \forall i, \forall r, \forall k, \forall f, \quad \text{(C19)}$$

$$P_{ir}^{k} \leq M_P Y_{ir}^{k}, \qquad \forall i, \forall r, \forall k, \quad \text{(C20)}$$

$$\Phi_{ir'}^{k} \leq + \sum_{c} \left(m_{fc}^{d} \Phi_{irc}^{d}\right)(2 - B_{ir'r}^{dk} - V_{fir'}^{d}), \qquad \forall i, \forall r, \forall r', \forall k, \forall d, \forall f, \quad \text{(C21)}$$

$$\Phi_{ir}^{k} \leq \gamma_i Y_{ir}^{k}, \qquad \forall i, \forall r, \forall k, \quad \text{(C22)}$$

$$\Phi_{irc}^{k} \geq (z_{irc}^{k} - z_{ir(c-1)}^{k}) R_{ir(c+1)}^{k}, \qquad \forall i, \forall r, \forall k, \forall c, \quad \text{(C23a)}$$

$$\Phi_{irc}^{k} \leq (z_{irc}^{k} - z_{ir(c-1)}^{k}) R_{irc}^{k}, \qquad \forall i, \forall r, \forall k, \forall c, \quad \text{(C23b)}$$

$$\Phi_{ir}^{k} = \sum_{c} \Phi_{irc}^{k}, \qquad \forall i, \forall r, \forall k, \quad \text{(C24)}$$

Domain

$$T_{ir}^{k},\ \Phi_{ir}^{k},\ P_{ir}^{k} \in \mathbb{R}^{\geq 0}, \qquad \forall i, \forall r, \forall k, \quad \text{(C25)}$$

$$\Phi_{irc}^{k} \in \mathbb{R}^{\geq 0}, \qquad \forall i, \forall r, \forall k, \forall c, \quad \text{(C26)}$$

$$V_{fir}^{k} \in \{0, 1\}, \qquad \forall f, \forall i, \forall r, \forall k, \quad \text{(C27)}$$

$$S_{ijrr'}^{k} \in \{0, 1\}, \qquad \forall i, \forall j, \forall r, \forall r', \forall k, \quad \text{(C28)}$$

$$R_{irc}^{k} \in \{0, 1\}, \qquad \forall i, \forall r, \forall k, \forall c, \quad \text{(C29)}$$

$$B_{irr'}^{kd},\ H_{irr'}^{kd} \in \{0, 1\}, \qquad \forall i, \forall r, \forall r', \forall k, \forall d, \quad \text{(C30)}$$

$$X_{ir}^{k} \in \mathbb{Z}^{\geq 0}, \qquad \forall i, \forall r, \forall k, \quad \text{(C31)}$$

This linear version of our model is the one that we implemented in our computation experiments that will be shown later. Note that Constraint C24 and variable Φ_{ir}^{k} could be omitted and the sum of artificial variables Φ_{irc}^{k} over the segments could be used instead of it. Nonetheless, for the sake of clarity, we included both types of variables, Φ_{ir}^{k} and Φ_{irc}^{k}, in our formulation.

Chapter 5

Solution Methods

In this chapter we present the solution techniques that we developed to solve the mathematical bi-objective problem defined in Section 4.2.1. Until today, there are several well known optimization problems that have been studied in their single-objective version. However, in the real world it is common to find that there are several criterion to be taken into account in the decision for the majority of problems. In many cases, such multiple objective conflict one with each other. In the case of a search in presence of a non-instantaneous disaster, it could be intuitive to prioritize zones with the higher POC and also zones with a high probability of being exhausted by the phenomenon in a near future moment. The two objectives included in our model, are designed to pursue such prioritization.

The optimality concept is not directly applicable to multi-objective problems, specially for cases where there is at least one objective that is expressed in different units than the others. In multi-objective optimization, it is required to introduce the concept of *Pareto-optimality*. A Pareto-optimal vector is a set of solutions for a given multi-objective problem, each one which have a higher value than the others for at least one of the objective functions. Figure 5.1 illustrates the idea for a bi-objective optimization, where both objectives are maximized. In this Figure, the light dots over the orange line represent solutions in the Pareto-optimal vector, and the dark dots represent dominated solutions. As it can be appreciated, there is no solution that overwhelms a Pareto-optimal solution in both objectives. When the optimization problem has multiple objectives, the solution consists in building the Pareto front for the problem and then selecting one Pareto-optimal solution to be implemented.

Our first attempt is based on the construction of a quasi-exact Pareto front through the iterative solution of our MILP on an ϵ-constraint fashion. Our second approach attempts to build an approximated Pareto front by implementing an Ant Colony Optimization algorithm. Our third alternative is a lexicographic based method where the two extremes of the Pareto front are identified by solving our MILP.

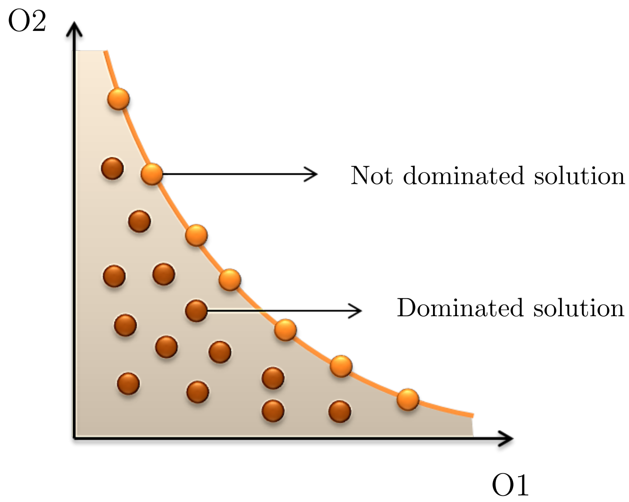


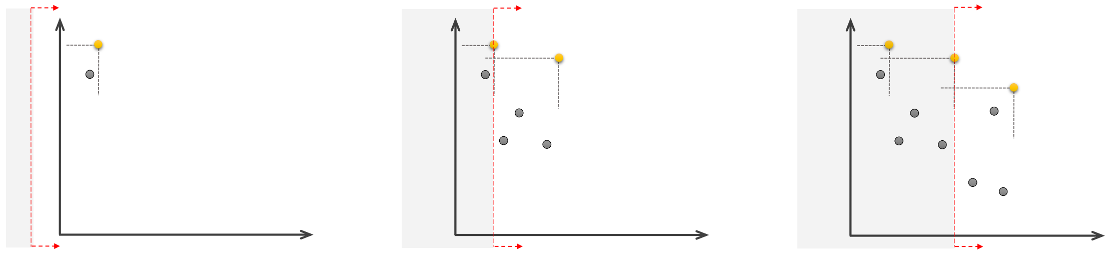
Figure 5.1: Pareto front for bi-objective optimization

5.1 Epsilon Constraint Method

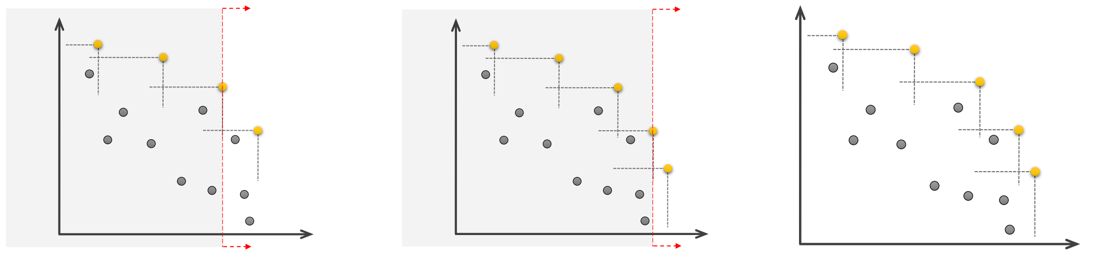
This first solution method corresponds to an implementation of the adaptive epsilon-constraint scheme introduced in Laumanns, Thiele, and Zitzler (2006). In general, the epsilon-constraint method works by selecting one objective function as the only objective and set the remaining objectives as constraints. A systematic variation of constraint bounds allows the method to explore the solution space, identifying elements of the Pareto front. The applicability of the method depends on the availability of a procedure to solve the mono-objective problem, here referred as $opt(M(f, \epsilon))$ following the notation of Laumanns et al. (2006).

The original epsilon-constraint method presents the necessity of determining the constraint values a priori, which in practice could result very hard to do. The adapted version proposed by Laumanns et al. (2006) deals with that drawback of the original version by generating appropriate constraint values during the run based on updated information about solution space. The construction procedure for a bi-objective Pareto front is illustrated in Figure 5.2, step by step. This example corresponds to a problem where both objectives are maximized, which is the type of problem addressed in this thesis. To start, the coordinate of epsilon constraint is fixed on $-\infty$ so that the first region explored by the algorithm is the whole potential objective space, $\mathbb{R}^2$. In step 1 $opt(M(f_1, -\infty))$ is solved and the first solution of the Pareto front, x is found. Then, the epsilon constraint is moved up to $f_2(x)$ and $opt(M(f_1, f_2(x)))$ is solved. The process is repeated at each step until a stopping condition is met. In our experiments, we have run the algorithm until the whole Pareto front was found. In order to know when it happens, we added a step 0 to the procedure, where we solved $opt(M(f_2, -\infty))$ and fixed $f_2(x)$ as the maximum possible value of f_2 that can be found.

The adaptive ϵ- constraint procedure implemented here presents significant advantages in comparison with other methods that attempt to build the whole Pareto front. The weighted sum method, for example, only guarantees to find supported Pareto-optimal solutions, which are those lying in convex regions of the objective space Laumanns et al. (2006). This drawback is evidenced in Figure 5.3. In this adaption of our example, red so-



(a) ϵ-constraint - Step 1 **(b)** ϵ-constraint - Step 2 **(c)** ϵ-constraint - Step 3



(d) ϵ-constraint - Step 4 **(e)** ϵ-constraint - Step 5 **(f)** Max-Max Pareto front

Figure 5.2: ϵ-constraint procedure

lution is a non-supported Pareto-optimal solution that weighted sum method, represented by the red dotted lines, is unable to find. The epsilon-constraint method implemented here guarantees to find this kind of solutions, provided that the mono-objective solution method is able to reach that point.

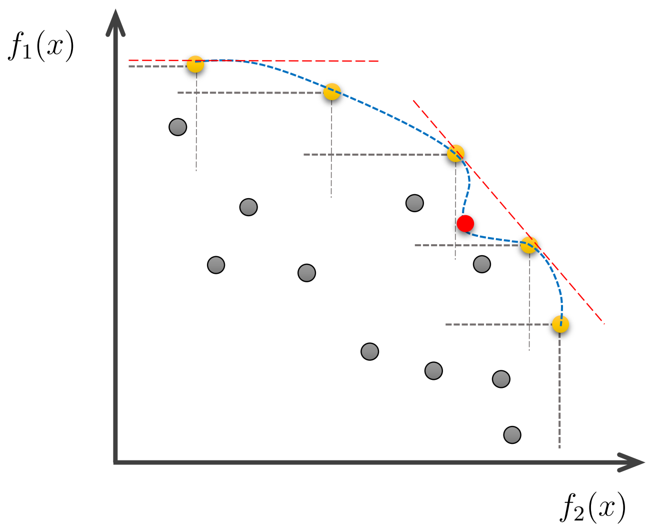


Figure 5.3: Weighted sum and non-supported Pareto-optimal solutions

The pseudo-code for our bi-objective problem is presented in Algorithm 1. Steps 2 and three correspond to the delimitation step where we define the higher value of f_2 to be searched, and thus we find a suitable stopping condition for the algorithm. Each iteration of the loop in steps 3-7, provides a new efficient solution of the bi-objective problem.

In this thesis, the mono-objective version of our linear model is solved by means of the $branch - and - cut$ algorithm implemented in CPLEX 12.6.2 for solving MILPs.

Algorithm 1 ϵ -constraint method for bi-objective Pareto Front construction

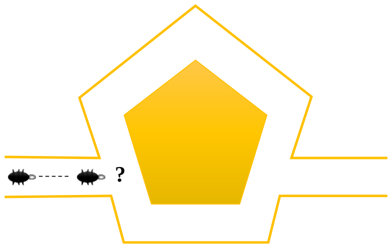
$\mathbf{P} := \emptyset; \quad \epsilon := -\infty$
$\mathbf{x}' := opt(M(f_2))$
$\mathbf{b} := f_2(\mathbf{x}')$
while $<\epsilon < b>$ **do**
  $\mathbf{x} := opt(M(f_1, \epsilon))$
  $\mathbf{P} := \{\mathbf{P}, \mathbf{x}\}$
  $\epsilon := f_2(\mathbf{x})$
end while
Output: Set of Pareto-optimal decision vectors $\mathbf{P}$

5.2 Pareto Multi-Agent Ant Colony Optimization

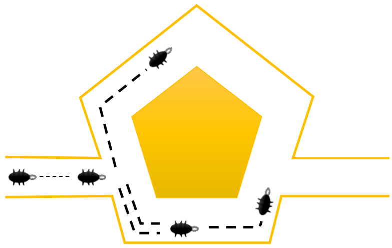
The OSPEAD can be seen as a generalization of the Team Orienteering Problem (TOP) where multiple visits to nodes are allowed, the profit is decreased each time a member of the team visits a node and the final destination of each member of the team is free. The TOP in turn, has as special case the Orienteering Problem (OP), known also as Selective Traveling Salesman Problem (STSP). The OP has been shown to be NP-hard by Golden, Levy, and Vohra (1987). Therefore, it can be concluded that the OSPEAD is NP-hard and thus, no efficient solution techniques for the problem at hand exists, especially when considering multiple objectives. Based on this assertion we worked on the construction of an alternative solution method, able to deal with the trade-off between solution quality and processing time. Seeking for such an equilibrium, we decided to work on a meta-heuristic algorithm based on Ant Colony Optimization (ACO). ACO was introduced on the 1990s by Dr. M. Dorigo and colleagues as a nature-inspired heuristic for the solution of hard combinatorial optimization problems (Dorigo & Blum, 2005). Since then, ACO has been effectively applied to several problems in routing and scheduling domains. For very good reviews on ACO, reader is referred to Dorigo and Stützle (2010) and Mohan and Baskaran (2012).

ACO philosophy is based on the natural learning process of real ants, who are capable of finding the shortest path from a food source to their nest by exploiting pheromone information (Dorigo & Gambardella, 1997). The process is illustrated in Figure 5.4. In the first scene, ants arrive at decision point with no information about paths length and start taking paths randomly. Given that ants move at an approximate constant rate, the shortest path is traversed by more ants in the same period. In consequence, pheromone level in the shortest path increases faster than it does in alternative path. The pheromone trail suffers evaporation, but ants keep using the shortest path at a higher rate and thus the pheromone level remains higher there than in alternative paths. As time advances, almost every ant chooses the shortest path.

ACO problem possess an exploration component which accounts for diversification on the solutions and avoids local optima. By the other side, ACO presents an exploitation component that focuses on the local improvement the most attractive solutions found at each iteration. Furthermore, ACO is known to be an alternative for obtaining good solutions in relatively short time. In the OSPEAD, most attractive cells are grouped on a short number of areas, typically two; the area surrounding the threat and the area



(a) Ants arrive at decision point



(b) Some ants choose the upper path and some the lower path

(c)
Pheromone level increases faster in shortest path

Figure 5.4: How real ants find shortest path
Based on: Dorigo and Gambardella (1997)

containing the POC peak. In some cases where there are expectations of finding the person at different areas, the POC could become multimodal. In that case the number of attractive areas could be greater than two, but the priority still grouped on a small number of areas. We find the exploitation component of ACO, a very interesting characteristic that could lead agents to converge exploring over the most critical zones. As a complement, the exploration component will avoid taking greedy decisions such as sending most reliable agents to cells with the highest POC or danger, without any care on the travel time involved. Our approach is an extension of the Pareto Ant Colony Optimization (PACO) method proposed in Schilde, Doerner, Hartl, and Kiechle (2009) for the bi-objective orienteering problem (OP). Our ACO method adds multiple agents, multiple visits and decreasing profits to the framework presented in Schilde et al. (2009). The most similar problem to the OSPEAD, found in the literature by us, which is solved by ACO is the VRP with multiple time windows and multiple visits addressed by Favaretto, Moretti, and Pellegrini (2007). Despite the similarities, Favaretto et al. (2007) assume fixed profits related to nodes and predefined time windows. The OSPEAD involves decreasing profits depending on the visits at each cell and dynamic time windows, which makes their approach unsuitable for our purposes. In this sense, the ACO method introduced here can be seen as an alternative to solve the VRP with multiple time windows, multiple visits and decreasing profits. The validation of this alternative is proposed as a future research.

In order to introduce our ACO algorithm, we will first describe the approach proposed in Schilde et al. (2009) and then we will explain the adaptions that we implemented in order to build an ACO system suitable for the OSPEAD.

5.2.1 Pareto Ant Colony Optimization (PACO)

The algorithm proposed in Schilde et al. (2009) can be decomposed in the following five main steps:

1. Initialization: a population is created and each ant is positioned at the starting vertex v_0. Additionally, each ant is provided with a set of objective weights p which intends to direct ants to find solutions uniformly distributed along the Pareto front. Additionally, each arc (v_i, v_j) is assigned an initial pheromone value $\tau_{ij}^f = \tau_0$ for each objective function f.

2. Construction: each ant builds a feasible tour from starting vertex to the ending vertex following the ant colony system scheme proposed by Dorigo and Gambardella (1997). The probability of visiting one of the feasible vertices is computed based on two man sources of information, namely the heuristic information η_{ij}^f and the pheromone information τ_{ij}^f.

3. Local pheromone update: each time a vertex is added to a partial sequence, a local pheromone update takes place. In this step, pheromone level at visited link (v_i, v_j) is decreased as a simulation of pheromone trail evaporation. This mechanism brings diversification of the set of solutions.

4. Iterative improvement: after each ant has built a complete feasible tour, a local search heuristic is applied to each solution. Best solutions resulting from this step are stored in an external memory and other ones are deleted.

5. Global pheromone update: finally, a global pheromone update procedure is performed where the pheromone level at links visited in the best and second best solutions for each objective, is increased by a predefined factor. This mechanism gives the algorithm the capability to exploit attractive solutions and build better paths from them.

* Repeat the process until stopping conditions are meet

5.2.2 Extending Pareto Ant Colony Optimization

In the previous section we described the PACO algorithm. Now we describe the adaptations that we implemented in PACO in order to make it suitable for solving the OSPEAD. There are three main differences among the PACO algorithm and the Pareto Multi-Agent Ant Colony Optimization (PMAACO) method introduced here:

1. PACO algorithm is oriented to problems involving a single agent like the TSP and the OP. Our PMAACO algorithm is able to deal with problems involving multiple agents like the VRP, the TOP and of course, the OSPEAD.

2. PACO considers fixed profits associated with each node. In OSPEAD, profits are allowed to variate, depending on the visits scheduled at each cell. PMAACO is adapted for this kind of behavior in the profits.

3. PACO allows a single visit per node. PMAACO not only considers multiple visits, but avoids simultaneous explorations at the same cell by imposing dynamic time windows.

The PMAACO algorithm implements virtually the same steps than the PACO algorithm, except for the iterative improvement, which we plan to cover on a future extension of this research. The pseudo-code for the PMAACO is presented in Algorithm 2.

Algorithm 2 PMAACO for the OSPEAD

while Stopping conditions remain unsatisfied **do**
  Create a new colony Col of size Psize
  ColonyState := partial
  for `<f=1:Psize>` **do**
    Assign a weight vector p to the ant
    Locate each agent at its initial position
    Tag each agent as Partial
    Tag Ant as partial
  end for
  while `<ColonyState == partial>` **do**
    for `<each ant in Col>` **do**
      if (ant f remains partial) **then**
        Randomly select a partial agent k from ant f
        Select next cell j to visit for agent k
        Update time window of chosen cell
        Perform local pheromone update on used links
        Check which agents remain partial
      end if
      Check if ant remains partial
    end for
    Check if colony remains partial
  end while
  Check efficiency
  Perform global pheromone update
end while
Output: Set of Pareto-optimal decision vectors **P**

As the PACO algorithm, PMAACO starts by creating a new colony. This time, ants will not be located at the initial vertex. In its place, agents will be located at their initial cell. The initialization step puts in evidence the first adaption of our algorithm, which is the depth of the data structure hierarchy three. PMAACO algorithm present a data structure three similar to that presented in Figure 5.5, where agents are located at the extreme of each branch. In this scheme, ants still representing solutions, but the routing decisions are individually taken by each agent. PACO algorithms do not consider multiple agents and then their data structure hierarchical three is limited to the ants' level. As in the case of PACO, we allow ants to share pheromone information. This feature makes agents from a given solution (ant) to enforce diversification on agents other solutions.

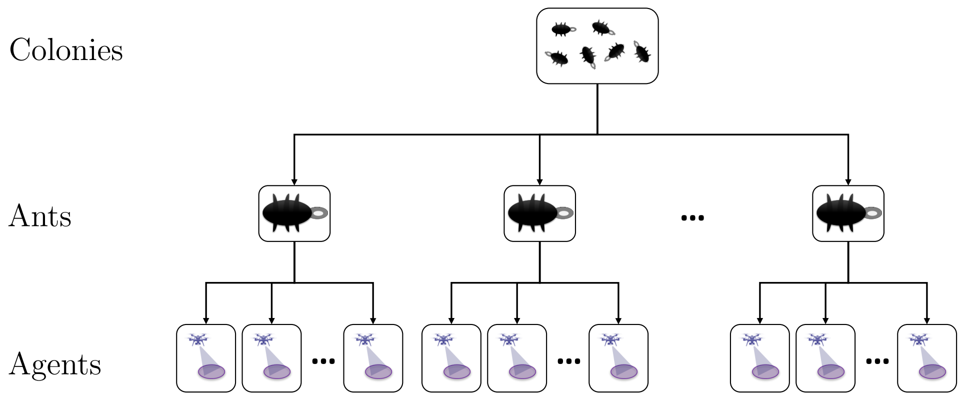


Figure 5.5: PMAACO hierarchy

During the first loop of Algorithm 2, each ant is provided with a weight vector p which will determine the degree of priority that such ant will give to each objective. Additionally, each agent is tagged as partial and remains at this state until there is any cell that could be visited by him without exceeding the time horizon W or performing simultaneous explorations at a given cell with other agent. Finally, each ant is tagged as partial and remains as this until sequences of all its agents become complete.

The construction step takes place in the second loop of Algorithm 2. In our first attempt to define that step, we declared the following procedure:

1. Select a partial ant;

2. Select each partial agent on this ant and choose the next cell on its sequence.

Fortunately, we promptly noted that such a procedure was reducing the solution space and some alternatives were never produced. In consequence, we modified the construction step to this new procedure:

1. Select a partial ant;

2. Randomly select a partial agent on this ant and choose the next cell on its sequence.

This new construction scheme keeps unaltered the solution space, allowing PMAACO to find any feasible solution. As stated before, the pheromone information is shared by all agents in a colony. In consequence, the local pheromone updated modifies the τ matrix for all agents. After adding a cell to the sequence of an agent, it is necessary to check which agents remain partial. Given that we are working with multiple agents and simultaneous explorations at the same cell are not allowed, the addition of a cell in the sequence of an agent may cause multiple partial agents to become complete agents. In contrast, in the PACO algorithm it is only necessary to check if the ant who selected the new destination remains partial. In the OSPEAD, an agent remains partial if there exists at least one cell that can be visited and scanned by the agent within the time window of length W, and without violating the constraint on no simultaneous explorations. After this step, it is possible to determine if the ant remains partial. An ant remains partial until all agents belonging to it become complete. Having updated all ants, it is necessary to determine if the colony remains partial. A colony remains partial until all ants belonging to it become complete.

When a colony is completed, the efficiency of encountered solutions is checked in order to delete dominated solutions. Then a global pheromone update is performed, increasing the attractiveness of those cells visited by all agents in the best two solutions for each objective function. The process is completed until a given stopping condition is meet. In our experiments, we predefined a number of colony generations as the stopping condition.

5.2.2.1 Heuristic information

The heuristic information η is an indicator of the benefit of visiting a candidate cell j departing from the agent's current position i. In the multi-objective case, η is typically calculated for each objective as:

$$\eta_{ij}^{o} = \frac{s_j^o}{c_{ij}} \tag{5.1}$$

where s_j^o represents the benefit of adding vertex j to the sequence and c_{ij} represents the corresponding cost. Following the notation for the OSPEAD declared in Section 4.2, PMAACO computes the heuristic information as follows:

$$\eta_{ij}^{k1} = \frac{w_j}{t_{ij}^k + O_k}, \qquad \eta_{ij}^{k2} = \frac{\gamma_j}{t_{ij}^k + O_k} \tag{5.2}$$

Given that we allow agent-dependent travel times and exploration times, PMAACO was adapted to compute agent dependent values for the heuristic information. Additionally, the value of η_{ij}^{ko} is updated each time that an agent is going to chose a destination. It is necessary since γ_i is decreased each time that an agent performs explorations there. However, given that each ant is solving an independent run of the problem, the POC map of each ant is also independent and thus the values of η_{ij}^{ko} in a given ant are not modified by decisions taken by other ants.

5.2.2.2 Pheromone information

Pheromone information has the objective of diversifying solutions within each colony and lead to convergence around most attractive solutions from a colony to another. As stated before, we work with a single pheromone value for link ij that is the same for all the ants in the colony. This approach makes the decisions of each agent to influx not only on agents belonging to the same ant, but also in agents belonging to other ants.

5.2.2.3 Decision rule

Let i be the last visited cell in the sequence of agent k and also let Ω_{ki} to be the set of feasible destinations for agent k located at cell i. Next destination in the sequence of agent k is selected by applying one of the following pseudo-random decision rules:

$$decisionrule = \begin{cases} rule\ 1, & \text{if } q \leq q_0 \\ rule\ 2, & \text{if } q_0 < q \leq q_1 \\ rule\ 3, & \text{otherwise} \end{cases}$$

The variable q is a random number uniformly distributed in $[0, 1]$, whereas q_0 and q_1 are parameters in the range $0 - 1$ specified by the user.

Rule 1: it can be seen as a greedy decision rule where agent k always selects the most attractive feasible cell from its current position as its next destination. Mathematically, this rule can be expressed as:

$$v_j = \underset{v_l \in \Omega(ki)}{\operatorname{argmax}} \left\{ \sum_o (\tau_{ilo}^{\alpha} \cdot \eta_{ilok}^{\beta} \cdot p_o) \right\} \tag{5.3}$$

Rule 2: similarly to rule 1, this rule gives priority to most attractive feasible cells from current position of agent k. Contrary to rule 1, this second rule is not deterministic and even the less attractive feasible destinations could be chosen. For this decision rule, a uniformly distributed pseudo-random number is generated and the destination is selected according to the following probability pie.

$$P(v_j) = \frac{\sum_o (\tau_{ijo}^{\alpha} \cdot \eta_{ijok}^{\beta} \cdot p_o)}{\sum_{l \in \Omega(ki)} \sum_o (\tau_{ilo}^{\alpha} \cdot \eta_{ilok}^{\beta} \cdot p_o)} \tag{5.4}$$

Rule 3: this last rule is not present in the PACO algorithm. We added it because we identified that with the previous two rules it was very difficult to reach more than 3 steps in a sequence without executing a greedy or semi-greedy decision and once again, there were some feasible solutions that were virtually never explored. This new decision rules gives the algorithm the opportunity to escape from obvious greedy sequences that we found to be dominated by experimentation. The rule consists on selecting a cell by the set of feasible destinations according to a uniformly distributed probability pie. This rule can me mathematically expressed as:

$$P(v_j) = \frac{1}{|\Omega_{ki}|} \tag{5.5}$$

5.2.2.4 Pheromone update

As mentioned we have two updating mechanisms responsible of evaporation and enhancement of the pheromone level. Each time an ant adj a cell to its sequence, the local pheromone update reduces the level of pheromones in traversed link ij. This update procedure can be seen as a mechanism of diversification that enhances the capacity of exploration oof the algorithm within each colony. By decreasing the level of pheromones in the last traversed link, unexplored combinations of cells involving a relatively high amount of pheromone become very attractive for future decisions. Local pheromone update is performed based on Equation 5.6, where ρ is a parameter that simulates the evaporation rate ($0 \leq \rho \leq 1$).

$$\tau_{ij}^k = (1 - \rho)\tau_{ij}^k + \rho\tau_0 \tag{5.6}$$

By the other side, global pheromone update takes place each time that a colony becomes complete. This time, pheromone level is increased in those links belonging to the best b solutions of each objective function. Global pheromone update is performed according to Equation 5.7.

$$\tau_{ij}^k = \tau_{ij}^k + \Delta\tau_{ij}^k \tag{5.7}$$

In this thesis, the value of $\Delta\tau_{ij}^k$ was fixed in τ_0 based on Schilde et al. (2009) and Dorigo and Gambardella (1997).

5.3 Lexicographic Method

Our third method seeks to find two efficient Pareto solutions in a short time. In fact, through this method we attempt to find the two extreme solutions of the Pareto front. For this purpose, we make use of the Lexicographic method which consists on ranking the set of objectives $F = \{f_1, ..., f_q\}$ by priority and then solve q single-objective problems, each one optimizing the objective function f_i, subject to optimal values obtained by the first f_{i-1} iterations to the first $i-1$ objective functions.

Let $L(f_r, f_s)$ be a function that returns the solution for the Lexicographic method giving priority to objective function f_r over objective function f_s. Then, this third approach consists on calling $L(f_1, f_2)$ and then $L(f_2, f_1)$ ans save both solutions.

This method could seems quite simple, nonetheless, it could provide an interesting alternative for cases when the threaten phenomenon is very aggressive and the information on the POC map is highly reliable so that it is very desirable to decide for a single one of the two objectives. In that scenario, the Lexicographic method will provide the decision maker with the ability to find the two extreme solutions and then apply a trade off rule to make the last decision. A procedure supporting this kind of decision is explained in the following section.

5.4 Artificial Decision maker

In previous sections, we have suggested the use of unmanned air vehicles (UAVs) as search agents. Our methods are suitable for other types of agents as humans, helicopters or airplanes, nonetheless, the use of UAVs eliminates the exposure of search agents to risky conditions on the search area during long time periods. By suggesting the use of UAVs, we propose to use human agents only to perform rescue tasks once the victim has been located, and also as search agents in relatively save places, far from the current location of the threatening phenomenon. Additionally, we are seeking for the automation of the decision making process for the exploration. An automated search planning based on last information recorded by UAVs will lead to faster and possibly more accurate decisions. In order to implement automation in a problem where the decision making problem involves multiple objectives, it is necessary to define an artificial decision maker (ADM). An ADM will emulate the reasoning of a human decision maker, based on predefined selection rules that will lead him to a single solution. In this section, we propose the use of the *max-min* ADM for the automation of the decision process in the OSPEAD.

The *max-min* ADM is oriented to find a solution where the worst of the two objectives values is not too bad. Suppose that we have the set of solutions presented in Table 5.1, composing our Pareto front.

Table 5.1: Dummy Pareto front

Solution	1	2	3	4	5	6
Obj_1	32	28	19	15	7	2
Obj_2	3	9	11	13	15	28

The first step consists on dividing the f_1 values for all the solutions by its maximum value, which in our example is 32. The same procedure should be performed for the f_2 values, dividing each one of them by 28. Resulting standardized values are shown in Table 5.2.

Table 5.2: Dummy Pareto front standardized

Solution	1	2	3	4	5	6
Obj_1	1.000	0.875	0.594	0.469	0.219	0.063
Obj_2	0.107	0.321	0.393	0.464	0.536	1.000

The next step is to identify the minimum of the two standardized objectives for each one of the solutions. The resulting vector is shown in Figure 5.3.

Table 5.3: Mins of Dummy Pareto front

Solution	1	2	3	4	5	6
$Min\{f_1, f_2\}$	0.107	0.321	0.393	0.464	0.219	0.063

Finally, the ADM selects the solution with the higher value on Table 5.3, which corresponds to solution 4.

5.5 Dynamic disaster simulation

A fundamental characteristic of the OSPEAD is the dynamic behavior of the disaster which progressively consumes parts of the search region. In order to validate our methods, we have searched a simulation tool capable of producing a Danger map over a discretized region. As part of the search process, we contacted numerous researchers and even agencies specialized in wildfire modeling. Thunderhead was the only organization that kindly put in our service a key for their fire simulation tool Pyrosim. Unfortunately, that simulation tool is mainly oriented to simulate indoor fires and then it was not suitable for our purposes.

In face of this situation, and knowing that this was not the focus of our research, we decided to build a basic wildfire simulation algorithm based on Cellular Automata (CA). This approach is particularly suitable for our methods since it decomposes the region of interest in a collection of cells and is able to build burn probability maps with a high level of accuracy in a very short time. Our wildfire simulation scheme is presented below:

Physical environment

The physical environment on CA is composed by a finite collection of cells. Typically those cells are equally in size and the conditions within each cell are assumed to be homogeneous. He have exploited the convenience of that discretization to build a general cellular decomposition of the search region, feasible for planning the search mission and conducting fire spread forecasts.

Cells' states

Each cell can be in one of a given set of predefined states. Such a state is typically represented by a number for convenience in analytic and computational analysis. In our approach, the following three states are considered:

1. Not burning: cells containing combustible vegetation
2. Burning: cells which are currently being consumed by fire
3. Burned: cells which have already been consumed by the fire

Transition rules

A transition rule acts upon a cell, defining its change of state from a discrete time step to another one. The CA evolves simultaneously in space and time by performing iterative implementations of the transition rule over the whole region. The transition rule determines the evolution on cells state based on the state of neighbor cells. In our CA algorithm, we adopted the following transition rules for a given cell i:

- if cell i is not burning, it becomes burning in the following time step with a probability $\frac{fc * ir}{nc}$, where fc are the number of burning neighbor cells, ir is a factor defined by us as ignition rate ($0 \leq ir \leq 1$). This factor determines how fast or easy the fire spreads from a cell to another. Finally, nc is the number of neighbor cells for cell i
- if cell i is burning, it becomes burned in the following time step with probability $1 - fd$, where fd is an indicator of the quality of combustibles present in the region. Higher values of fd elongates the presence of the fire on cell i
- if cell i is burned, it remains burned for the following time step

Figure 5.6 presents a simulation performed by us as part of our initial tests with CA.

For each run of our computational experiment, we performed 300.000 simulations of fire spread for the given time window in order to reach convergence on the forecast. The time required for that pre-computation was not significant, even in the largest instances.

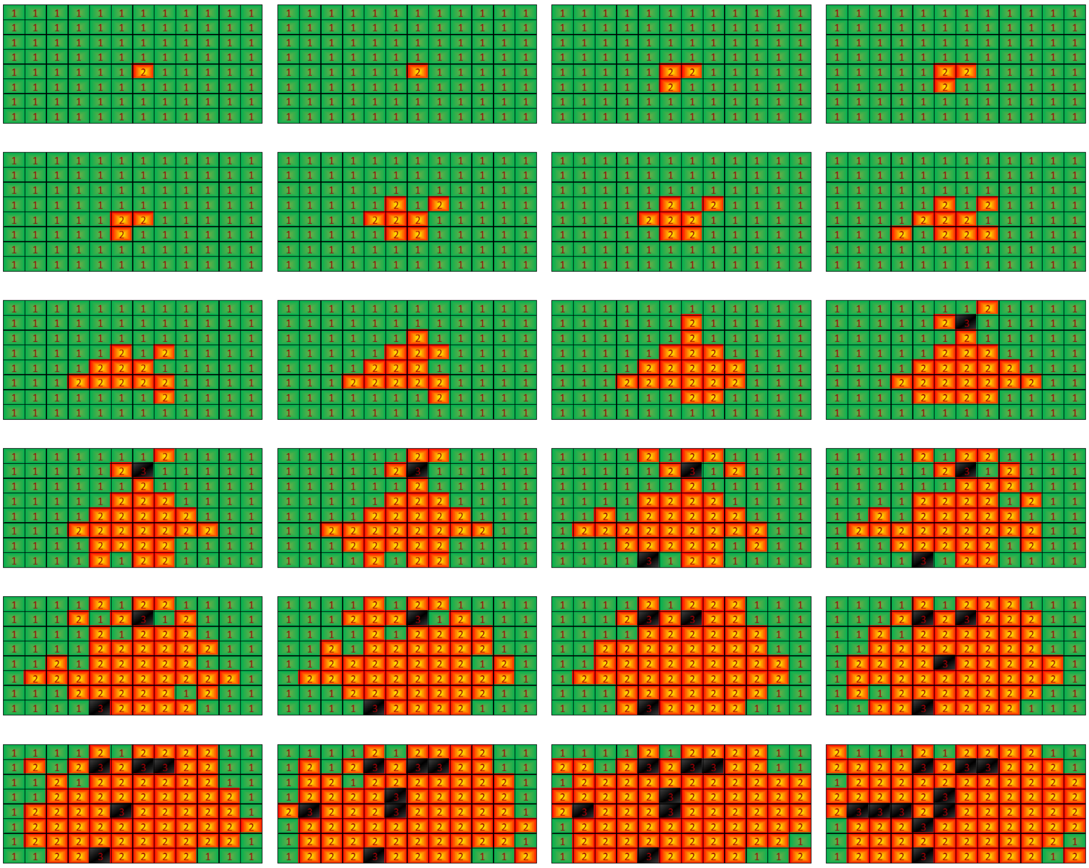


Figure 5.6: Example of cellular automata simulation of fire spread

Chapter 6

Findings

In previous two chapters we introduced the methods implemented in this thesis for the solution of the OSPEAD. We first developed a Mixed Integer Nonlinear Program, which was then linearized by means of piecewise linear approximation. Then, we presented three solution methods which attempt to solve the problem with different trade-offs between solution quality and processing time. In this chapter, we test the performance of our tree solution methods in multiple scenarios. In order to perform a fairly comparison among the three methods, the values obtained by the Epsilon-constraint method and the Lexicographic method where corrected after the run. This procedure sometimes makes dominated solutions to appear in the Pareto front coming from those two methods, even when they are solved by exact algorithms. This happens because the piecewise linear approximation implemented here does not guarantee a perfect fitting to the original curve. In consequence, those two methods are prone to detect false efficient solutions and ignore efficient ones. However, as it will be evidenced with the experiments, the solutions delivered by Epsilon-constraint the Lexicographic methods are relatively good in comparison with the PMAACO algorithm.

The last section of this chapter validates the receding horizon approach proposed in Section 4.1.3, which makes our methodology reactive to the dynamic behavior of the disaster. Additionally, that section implements the Decision Maker proposed in Section 5.4 which allows the application of our methods with autonomous robotic search agents.

We used Netbeans 8.0.2 and JDK 1.8.0 for implementation and simulation. All experiments were run on an Intel Core i5-6300U with 2.5GHz and 8GB RAM.

6.1 Data sets

Aiming to provide validity to our results, we made an effort for implementing realistic parameters in our experiments. The fire front speed (Fs) was fixed on 10 km/h = 2.78 m/s, which corresponds to a regular wildfire Scott (2012). The cell size (ℓ x ℓ) was fixed according to agent physical and technological capabilities. According to Goodrich et al. (2008) and Lin and Goodrich (2009), the maximum area that could be covered by a camera in order to enable for recognition of a human is no wider than 32m x 24m. Following them, we fixed our cell size in 24m x 24m. Given the values of (Fs) and ℓ, we were able to determine the length of a fire step, meaning the expected time to consume a cell with 24 x 24 m^2. The fire step (Ft) was empirically computed as ℓ/Fs, resulting in 8.64 s $\approx$ 9 sec. We considered time windows of one and two Ft, corresponding to 9 and 18 sec, and grids composed by 9 and 16 cells, representing regions of 5184 and 9216 m^2. The values of σ_1, σ_2 and ρ: where set in 7, 15 and 0 respectively, based on Chung and Burdick (2007). Additionally, the POC distribution was centered at the right down corner of the region in all scenarios as follows:

- Scenarios 1, 3, 5 and 7: $\mu_1 = 2$, $\mu_2 = 0$
- Scenarios 2, 4, 6 and 8: $\mu_1 = 3$, $\mu_2 = 0$

Finally, three different types of agents were considered. Their specifications are detailed in Table 6.1.

Table 6.1: Agents' specifications

Agent	**1**	**2**	**3**	**Units**	**Source**
Speed	11	13.4	10	m / sec	S1
Reliability	0.9	0.6	0.7	% accuracy	S2
Exploration time	4.4	4.5	4.3	sec	S3

S1. Casbeer et al. (2005); Carpin et al. (2013)
S2. Waharte et al. (2010); Le Thi et al. (2014)
S3. Flushing et al. (2014)

6.2 Validation

In this section, we test the coherence on the results delivered by our three solution methods for a small study case. Such study case corresponds to the smallest possible instance that can be constricted based on our data set. It involves two search agents, 18 seconds of planning and a 3x3 grid decomposition of the search region. The objective values for the solutions obtained by our three solution methods are plotted in Figure 6.1.

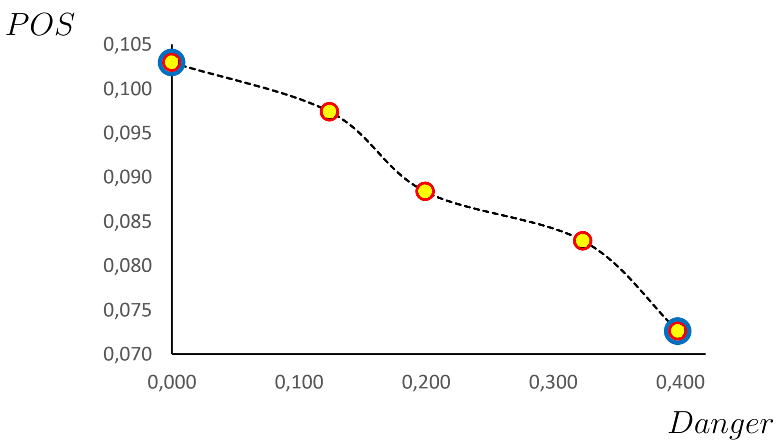


Figure 6.1: Pareto front Scenario 1

Yellow points correspond to the solutions obtained by the Epsilon-constraint method, red bordered circles correspond to the solutions delivered by our PMAACO algorithm and blue bordered circles correspond to the solutions delivered by the Lexicographic method. As it can be appreciated, the Epsilon-constraint method and the PMAACO are able to find the whole Pareto front for this small instance. By its side, the Lexicographic method achieves its goal of finding the extreme Pareto efficient solutions.

The search sequences corresponding to the solutions plotted in Figure 6.1 are presented in Tables 6.2 to 6.4. For all the sequences, we validated that no simultaneous explorations were performed simultaneously at the same cell. Additionally, we verified that all the sequences were shorter than the time window of 9 seconds.

Table 6.2: Epsilon-Constraint Scenario 1

Epsilon - constraint (0.748 sec)					
Solution	1	2	3	4	5
Agent 1	0 - 1	0 - 4	0 - 1	0 - 4	0 - 3
Agent 2	0 - 0 - 0	0 - 0 - 0	0 - 7	0 - 7	0 - 7
Danger (O1)	0.0	0.12415	0.19945	0.32359	0.39834
POS (O2)	0.10294	0.09738	0.08834	0.08278	0.07257

Table 6.3: Ant Colony Scenario 1

Ant Colony (0.567 sec)					
Solution	1	2	3	4	5
Agent 1	0 - 1	0 - 4	0 - 1	0 - 4	0 - 3
Agent 2	0 - 0 - 0	0 - 0 - 0	0 - 7	0 - 7	0 - 7
Danger (O1)	0.0	0.12415	0.19945	0.32359	0.39834
POS (O2)	0.10294	0.09738	0.08834	0.08278	0.07257

Table 6.4: Lexicographic method Scenario 1

Lexicographic (0.27 sec)		
Solution	1	5
Agent 1	0 - 1	0 - 3
Agent 2	0 - 0 - 0	0 - 7
Danger (O1)	0.0	0.39834
POS (O2)	0.10294	0.07257

Finally, we implemented the *max-min* ADM[1] introduced in Section 5.4 to select a single solution for each one of the sets of solutions coming from the three solution methods.

ADM applied to Pareto front

As described in Section 5.4, the first step for the application of the *max-min* ADM is to identify, for each objective function, the solution with the greatest objective value. In

[1] Artificial Decision Maker

this case, such value is 0.39834 for cumulative Danger and 0.10294 for cumulative POS. Then, each one of the objective values of each objective function is standardized with respect to the greatest value of that function. Resulting values for our small study case are presented in Table 6.5.

Table 6.5: Standardized Pareto front Scenario 1

Solution	1	2	3	4	5
$Danger(O1)$	0.000	0.312	0.501	0.812	1.000
$POS(O2)$	1.000	0.946	0.858	0.804	0.705

Next step is to identify the smallest standardized objective value of each objective function. The resulting vector is shown in Table 6.6.

Table 6.6: Mins of Pareto front Scenario 1

Solution	1	2	3	4	5
$Min\{f_1, f_2\}$	0.000	0.312	0.501	0.804	0.705

Finally, the solution associated to the highest value of Table 6.6 is selected as the solution to be executed by the search team. In this case, Solution 4, with a Danger value of 0.32359 and a POS value of 0.08278 is the one selected by the *max-min* ADM. That solution corresponds to the 81.2% and the 80.4% of the greatest values present in the Pareto front for Danger and Cumulative, respectively.

ADM applied to extreme points

In the case of the Lexicographic method, the selection of a single solution only requires the comparison between the two extreme Pareto points. Those two points correspond to solutions 1 and 5 in Table 6.6. From there, it can be concluded that the solution taken by the *max-min* ADM for the case of the Lexicographic method is number 5. That solution presents a Danger value of 0.39834 and a POS value of 0.07257, corresponding to the 100% and the 70.5% of the best objective values found in the Pareto front for the corresponding objective function.

6.3 Comparison of CPU performance

In this section we perform a large number of experiments that allow us to compare the performance on our three methods. At each experiment, a single run of the Epsilon-constraint method and the Lexicographic method was sufficient to find the intended solution. In contrast, the results of our Ant Colony algorithm were averaged over 30 independent trials.

All three solution methods proposed in this thesis are prone to find non-efficient solutions as part of their Pareto front. In the case of the PMAACO it happens because as a meta-heuristic, it does not warranty to find the exact Pareto front. By their side, the Epsilon-constraint method and the Lexicographic method are affected by the linear approximation scheme adopted in order to make the model linear. The linearization modifies the solution space of the original problem and then, those two methods only warranty to find the exact Pareto front of the modified problem. In this case, it is the trade of for working with a linear programming model. Fortunately, if the approximation breakpoints are adequately selected, our model tends to find the exact Pareto front. However, in order to test the performance of our algorithm, we defined the best merged Pareto front $F*$, which results From the combination of all efficient solutions found by each one of the three solution methods. The combined Pareto front substitutes the exact Pareto front and is adopted as an ideal in our comparisons. The following performance indicators were implemented for each experimental condition tested:

1. $|P|$: number of solutions in the Pareto front found by a given method
2. $|P \cap F^*|$: number of solutions in the intersection between the best combined Pareto front $F*$ and the Pareto front P found by a given algorithm.
3. $|P \cap F^*|$ / $|F^*|$: percentage of the best combined Pareto front $F*$, discovered by a given solution method.
4. D: generational distance found by means of Equation 6.1.

$$D = \frac{\left(\sum_{i \in P} {d_i}^q\right)^{1/q}}{|P|} \tag{6.1}$$

In this experiment, the value of q was fixed at 2.

The set of experimental conditions considered in our experiment are shown in Table 6.7. For each one of those conditions, we run our methods and saved their advances at different time steps. Results are summarized in Tables 6.8 to 6.15.

Table 6.7: Experimental design

Scenario	Agents (number)	Window (secs)	Region (dimensions)
1	2	9	3x3
2	2	9	4x4
3	2	18	3x3
4	2	18	4x4
5	3	9	3x3
6	3	9	4x4
7	3	18	3x3
8	3	18	4x4

Table 6.8 corresponds to the scenario that we used in the last section to validate our methods. The results for both, Scenario 1 and Scenario 2 are quite similar. In both cases, all three algorithms were able to find the whole Pareto front in a short time. The increase in the grid size from Scenario 1 to Scenario 2 caused a minimal enlargement of solution times.

Table 6.8: Scenario 1

	Ants	Lexicographic			Epsilon			
$Time(sec)$	0,567	0,201	0,270	0,216	0,332	0,632	0,668	0,748
$\|P\|$	5,0	1,0	2,0	1,0	2,0	3,0	4,0	5,0
$\|P \cap F^*\|$	5,0	1,0	2,0	1,0	2,0	3,0	4,0	5,0
$\|P \cap F^*\| \ / \ \|F^*\|$	100,0%	20,0%	40,0%	20,0%	40,0%	60,0%	80,0%	100,0%
D	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000

Table 6.9: Scenario 2

	Ants		Lexicographic		Epsilon		
$Time(sec)$	0,820	1,639	0,316	0,453	0,356	0,492	1,144
$\|P\|$	3,0	3,0	1,0	2,0	1,0	2,0	3,0
$\|P \cap F^*\|$	3,0	3,0	1,0	2,0	1,0	2,0	3,0
$\|P \cap F^*\| \ / \ \|F^*\|$	100,0%	100,0%	33,3%	66,7%	33,3%	66,7%	100,0%
D	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000

Pictures 6.2 and 6.3 show the percentage of $F*$ and the generational distance D for each solution method, at each one of the time steps included in Table 6.9, respectively. For better visualization, all time steps where scaled to the same length. In this small Scenario, the behavior of three algorithms is very similar. Nonetheless, Ant Colony algorithm begins showing its natural ability as an heuristic, which rapidly identifies good solutions. From Figure 6.3 it is evident that all solutions found by each solution method at each time step were part of $F*$.

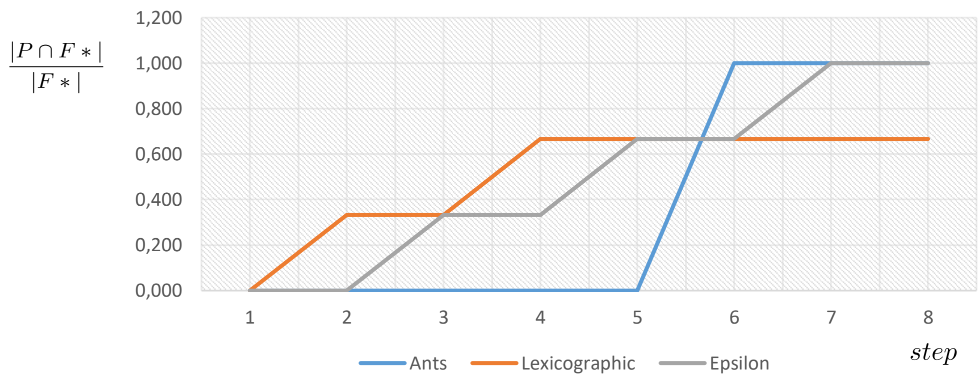


Figure 6.2: $|P \cap F^*|$ vs *step* - Scenario 2

In Scenarios 3 and 4, the same combinations on number of agents and grid size than Scenarios 1 and 2 were tested for a duplicated time window of 18 secs. Under these conditions, all three methods suffered a noticeable increase in its running time. Particularly, the Epsilon-constraint increased its time from 0.748 secs in Scenario 1 to 73.707

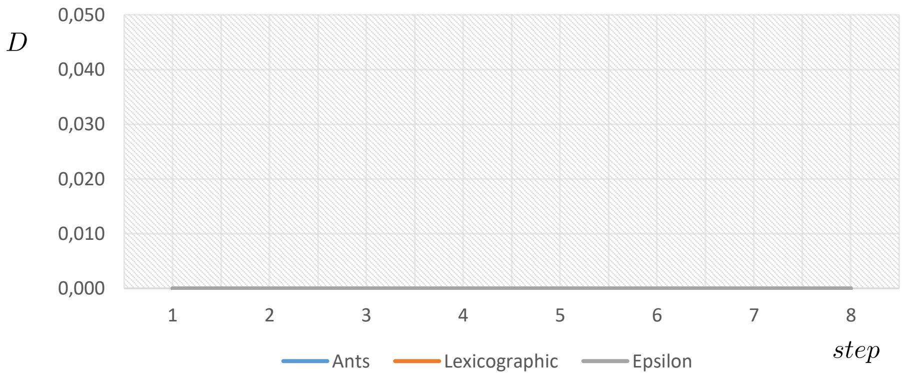


Figure 6.3: D vs $step$ - Scenario 2

secs in Scenario 2, and from 1.144 secs in Scenario 3 to 283.548 secs in Scenario 4. It can also be noted that the performance of Ant Colony was better than Epsilon-constraint performance in Scenario 3, in terms of the percentage of $F*$ found. In contrast, Epsilon-constraint was able to find the whole Pareto front over its 283.548 secs on running time for Scenario 4, while Ant Colony ended with a regular value of 76.4%. It is also interesting to see that independently of the increase in solution time, all three methods conserved a short generational distance in all four Scenarios. It means that the quality of solutions obtained by all three methods was high in the first four scenarios.

Table 6.10: Scenario 3

	Ants				Lexicographic		Epsilon					
$Time(sec)$	1,765	3,530	5,295	7,059	4,120	9,363	22,498	36,464	48,676	59,651	67,155	73,706
$\|P\|$	21,3	23,7	24,5	25,0	1,0	2,0	4,0	8,0	12,0	16,0	19,0	22,0
$\|P \cap F^*\|$	17,7	21,2	22,9	23,8	1,0	2,0	4,0	5,0	9,0	13,0	16,0	19,0
$\|P \cap F^*\| \,/\, \|F^*\|$	70,6%	84,7%	91,6%	95,0%	4,0%	8,0%	16,0%	20,0%	36,0%	52,0%	64,0%	76,0%
D	0,00259	0,00140	0,00085	0,00073	0,00000	0,00000	0,00000	0,02706	0,00000	0,00000	0,00000	0,00000

Table 6.11: Scenario 4

	Ants				Lexicographic		Epsilon					
$Time(sec)$	8,182	16,365	24,547	32,729	3,413	23,566	100,602	189,510	240,116	258,919	271,845	283,548
$\|P\|$	21,2	21,7	22,1	22,8	1,0	2,0	5,0	10,0	15,0	19,0	23,0	27,0
$\|P \cap F^*\|$	10,3	14,0	16,7	19,1	1,0	2,0	5,0	9,0	14,0	18,0	21,0	25,0
$\|P \cap F^*\| \,/\, \|F^*\|$	41,0%	56,0%	66,9%	76,4%	4,0%	8,0%	20,0%	36,0%	56,0%	72,0%	84,0%	100,0%
D	0,00372	0,00212	0,00122	0,00051	0,00000	0,00000	0,00000	0,00488	0,00000	0,00000	0,00584	0,00000

The change of behavior registered in Figure 6.2 with respect to Figure 6.4 is also evident. This time, the Ant Colony Optimization was able to find more than the 70% of $F*$ in the few first time steps and then it became idle at time step 7. By its side, Epsilon Constraint showed a relatively stable increase on the size of P with each time step from the moment when it founds the first solution.

In Scenarios 5 and 6, the time window was fixed again in 9 secs but this time the number of agents was increased to 3. This time, the result was a reduction on the solution times in a similar order to the solution times in Scenarios 1 and 2. With a time window of length 9 secs, the effect of increasing the number of agents from 2 to 3 is not significant. A probable reason for this behavior may be that having the same window size, no matter

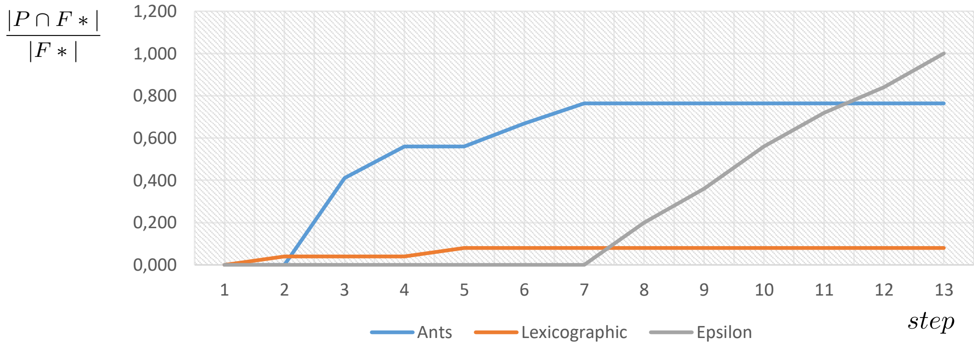


Figure 6.4: $|P \cap F^*|$ vs *step* - Scenario 4

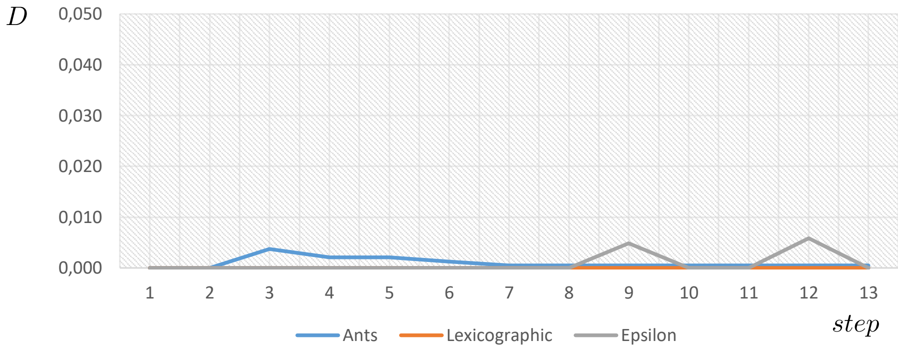


Figure 6.5: D vs *step* - Scenario 4

how far the region is extended, agents are limited to perform a given set of explorations. To make it clear, if we set a region of dimensions 100 x 100 without increasing the time window, the number of variables and constraints will be extended but the feasible region will remain the same.

Once again, in Scenarios 5 ans 6, Ant Colony and Lexicographic method achieved their goal by finding the combined Pareto front $F*$ and the extreme Pareto solutions, respectively. Epsilon-constraint also shown a very good performance with an 83.3% of $F*$ in Scenario 5 and 100% in Scenario 6.

Table 6.12: Scenario 5

	Ants	Lexicographic		Epsilon					
$Time(sec)$	0,696	0,404	0,489	0,285	0,371	0,539	0,597	0,665	0,711
$\|P\|$	6,0	1,0	2,0	1,0	2,0	3,0	4,0	5,0	6,0
$\|P \cap F^*\|$	6,0	1,0	2,0	1,0	2,0	3,0	4,0	4,0	5,0
$\|P \cap F^*\|$ / $\|F^*\|$	100,0%	16,7%	33,3%	16,7%	33,3%	50,0%	66,7%	66,7%	83,3%
D	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00136	0,00000

For Scenarios 7 and 8, the number of agents remained in 3 as in Scenarios 5 and 6, but this time we also duplicated the time window to 18 secs. Based on our discussion about the results from Scenarios 5 and 6, it was expected to find and increase on the solution time for Scenarios 7 and 8, based on the fact that the number of agents was fixed in the top, with a combination of a duplication in the time window, which dramatically increases the size of the solution space. The results was as expected, observing the largest solution

Table 6.13: Scenario 6

	Ants		Lexicographic		Epsilon		
$Time(sec)$	0,923	1,847	0,526	0,649	0,474	0,593	0,697
$\|P\|$	3,0	3,0	1,0	2,0	1,0	2,0	3,0
$\|P \cap F^*\|$	3,0	3,0	1,0	2,0	1,0	2,0	3,0
$\|P \cap F^*\| \ / \ \|F^*\|$	100,0%	100,0%	33,3%	66,7%	33,3%	66,7%	100,0%
D	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000

times of this section of our experiment. Epsilon-constraint reached approximately the 80% and 95% of $F*$ at Scenarios 7 and 8, respectively in 20 minutes and 85 minutes. By its side, Ant Colony took less than 1 minute in both Scenarios to be solved, reaching the 60% and 9% of $F*$ on Scenarios 7 and 8, respectively. The low percentage of $F*$ found in Scenario 8 is not an advice of a bad performance of the algorithm. In fact, it found 32.6 solutions on average, presenting less than 0.005 of generational distance from the best combined Pareto $F*$. It means that despite solutions found by Ant Colony were majorly out of $F*$ set, they were high quality solutions.

Table 6.14: Scenario 7

	Ants				Lexicographic		Epsilon					
$Time(sec)$	8,105	16,209	24,314	32,419	3,647	112,332	746,726	1008,191	1126,248	1184,889	1222,603	1242,691
$\|P\|$	13,9	14,2	14,7	14,8	1,0	2,0	4,0	8,0	11,0	14,0	17,0	20,0
$\|P \cap F^*\|$	3,0	5,1	7,4	9,6	0,0	0,0	0,0	3,0	6,0	9,0	12,0	13,0
$\|P \cap F^*\| \ / \ \|F^*\|$	19,0%	31,7%	46,3%	59,7%	0,0%	0,0%	0,0%	18,8%	37,5%	56,3%	75,0%	81,3%
D	0,00577	0,00477	0,00410	0,00318	0,20069	0,00015	0,06015	0,00095	0,00000	0,00000	0,00000	0,01629

Table 6.15: Scenario 8

	Ants				Lexicographic		Epsilon					
$Time(sec)$	12,494	24,988	37,483	49,977	20,697	293,732	2282,485	3579,598	4389,569	4796,585	5034,174	5127,199
$\|P\|$	24,4	28,8	31,6	32,6	1,0	2,0	7,0	14,0	21,0	28,0	35,0	42,0
$\|P \cap F^*\|$	1,3	2,0	2,8	3,6	1,0	2,0	7,0	14,0	19,0	26,0	33,0	39,0
$\|P \cap F^*\| \ / \ \|F^*\|$	3,0%	4,8%	6,8%	8,7%	2,4%	4,9%	17,1%	34,1%	46,3%	63,4%	80,5%	95,1%
D	0,00536	0,00473	0,00423	0,00400	0,00000	0,00000	0,00000	0,00000	0,00501	0,00000	0,00000	0,00139

This fact is evidenced in Figures 6.6 and Table 6.7, where the percentage of $F*$ found by Epsilon goes up as a rocket in a constant rate and the same percentage for Ant Colony remains very low, while the generational distance D for all methods remains very low.

6.3.1 Performance on the top of the experimental region

The computational time in Scenario 8 was so large in comparison to other scenarios, especially for the Epsilon-constraint method which employed more than one hour in the solution. Nonetheless, our Ant Colony algorithm found multiple solutions with a short generational distance from the Pareto front. That result motivated us to extend our experiments on the top of the experimental region in order to compare the performance of our algorithms when the scenario seems to become large. The new experiment consists on taking Scenario 8 and creating three new scenarios, each one by increasing one factor independently to an additional level. Table 6.16 shows the experimental conditions that we implemented for the extension of our experiment.

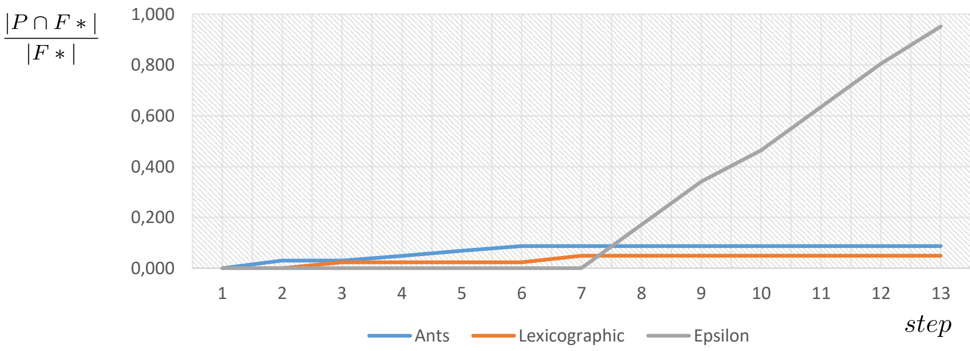


Figure 6.6: $|P \cap F^*|$ vs *step* - Scenario 8

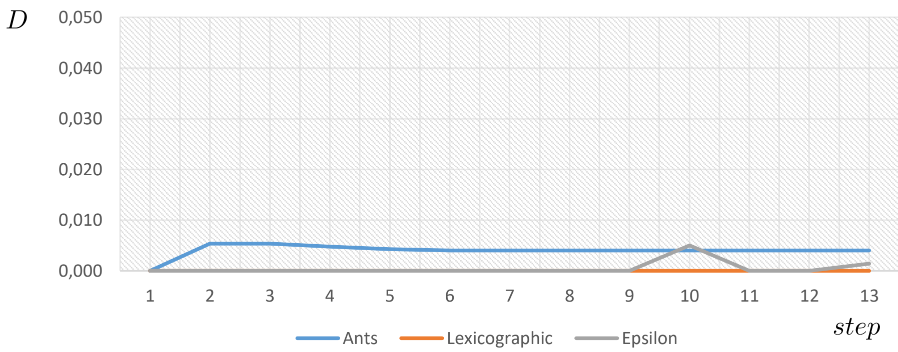


Figure 6.7: D vs *step* - Scenario 8

The computational time of the Epsilon-constraint method resulted prohibitive in all cases. Particularly in Scenarios 10 and 11, this method required 8 and 15 hours respectively to be completed. Times of the Lexicographic method remained manageable except for Scenario 11, where it was solved in approximately a half hour. By its side, our Ant Colony algorithm always performed in approximately one minute. Once again, the percentage of $F*$ found does not provide sufficient information, and the generational distance tells the truth. The approximated Pareto fronts found by Ant Colony in all three scenarios was very close to the Pareto front found by the Epsilon-constraint method. Actually, in Scenario 10, our Ant Colony method identified multiple efficient solutions that the Epsilon-constraint method was not able to identify. All three Pareto fronts are plotted in Figure 6.8.

Table 6.16: Extended experiment

Scenario	Agents (number)	Window (secs)	Region (dimensions)
9	3	18	5x5
10	3	27	4x4
11	4	18	4x4

Table 6.17: Scenario 9

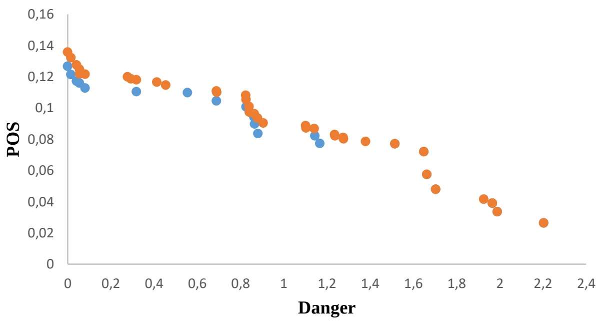
	Ants				Lexicographic		Epsilon					
$Time(sec)$	14,187	28,375	42,562	56,750	4,553	163,469	2468,685	3796,152	4317,715	4521,942	4589,640	4617,543
$\|P\|$	19,0	21,0	21,9	23,4	1,0	2,0	6,0	12,0	18,0	24,0	30,0	36,0
$\|P \cap F^*\|$	3,6	5,1	6,4	8,1	1,0	2,0	6,0	12,0	18,0	24,0	30,0	36,0
$\|P \cap F^*\| / \|F^*\|$	10,1%	14,2%	17,9%	22,4%	2,8%	5,6%	16,7%	33,3%	50,0%	66,7%	83,3%	100,0%
D	0,00525	0,00456	0,00437	0,00395	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000

Table 6.18: Scenario 10

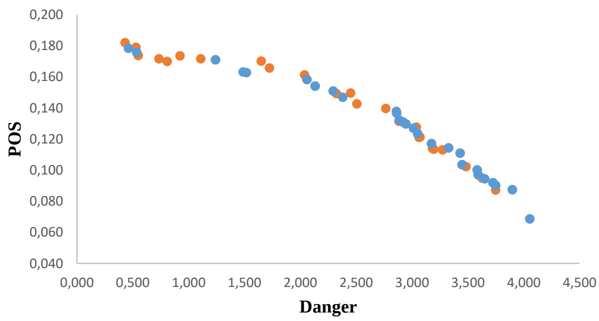
	Ants				Lexicographic		Epsilon					
$Time(sec)$	13,708	27,416	41,125	54,833	46,083	160,114	27515,217	28237,739	28651,414	29726,831	30004,499	30176,728
$\|P\|$	22,3	24,2	26,2	28,6	1,0	2,0	8,0	16,0	23,0	30,0	37,0	44,0
$\|P \cap F^*\|$	5,1	8,8	12,0	16,2	1,0	2,0	5,0	12,0	18,0	23,0	28,0	34,0
$\|P \cap F^*\| / \|F^*\|$	13,1%	22,6%	30,9%	41,6%	2,6%	5,1%	12,8%	30,8%	46,2%	59,0%	71,8%	87,2%
D	0,01325	0,01103	0,00968	0,00804	0,00000	0,00000	0,02686	0,00340	0,00055	0,00166	0,00791	0,00040

Table 6.19: Scenario 11

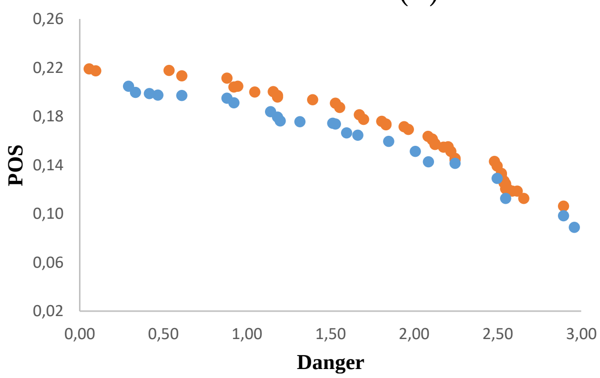
	Ants				Lexicographic		Epsilon					
$Time(sec)$	15,452	30,904	46,357	61,809	31,493	1745,724	43874,563	50111,802	51964,323	53396,104	53911,711	54192,922
$\|P\|$	19,8	21,5	23,2	25,1	1,0	2,0	8,0	16,0	24,0	32,0	40,0	47,0
$\|P \cap F^*\|$	0,2	0,3	0,4	0,5	1,0	2,0	5,0	13,0	21,0	28,0	36,0	43,0
$\|P \cap F^*\| / \|F^*\|$	0,5%	0,6%	1,0%	1,2%	2,3%	4,7%	11,6%	30,2%	48,8%	65,1%	83,7%	100,0%
D	0,01365	0,01209	0,01151	0,01059	0,00000	0,00000	0,01410	0,00000	0,00000	0,00358	0,00000	0,00000



(a) Pareto front for Scenario 9



(b) Pareto front for Scenario 10



(c) Pareto front for Scenario 11

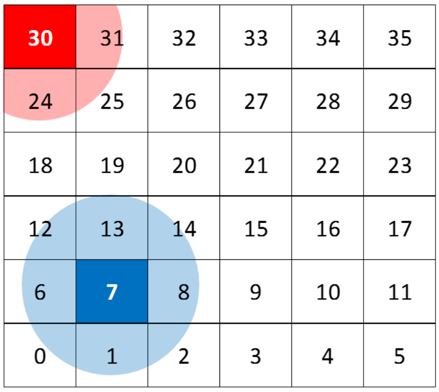
Figure 6.8: Pareto front on big size instances

6.4 Statistics of complete missions applying a receding horizon scheme

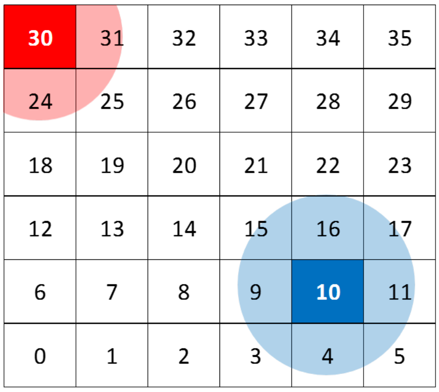
In Section 4.1.3 we proposed the solution of the OSPEAD under a receding horizon scheme. That scheme is intended to provide our methodology with adaption capabilities, suitable for the dynamic environment imposed by a non-instantaneous disaster. In the present section, we illustrate a set of experiments where we implement such scheme aiming to validate its applicability.

All experiments presented in previous sections were oriented to compare the performance of the three solution methods in terms of solution quality and processing time, considering a single time window. By the other side, the application of the receding horizon approach allows us to perform complete search missions. In out second set of experiments, every complete mission ended with one of the following two events: (1) the target was found; (2) the target was reached by the disaster.

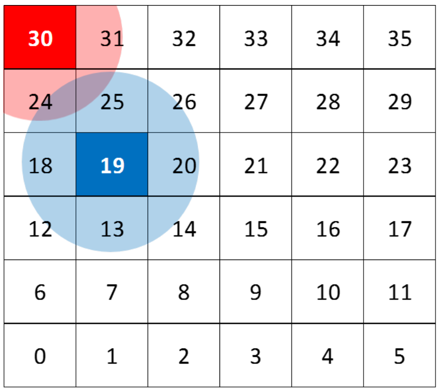
In this new experiment we centered the POC map at four different cells as is shown in Figure 6.9. In each scenario, we performed 50 simulations of complete search missions, where the victim was randomly located based on the probability distribution stated by the POC map. Additionally, independent fire simulations were performed for each one of the 50 simulations of each scenario. Wildfire initial point was always located at cell 30 in the upper left corner of the grid.



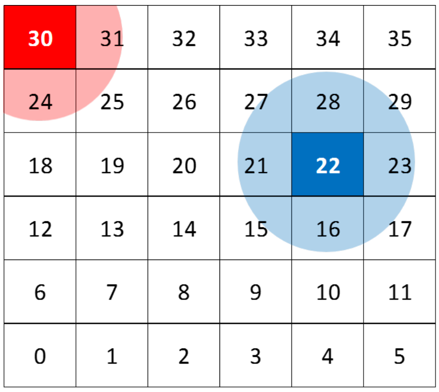
(a) Complete mission Scenario 1



(b) Complete mission Scenario 2



(c) Complete mission Scenario 3



(d) Complete mission Scenario 4

Figure 6.9: Scenarios for complete-mission related experiments

At each simulation, the experiment proceeded as follows:

1. Randomly locate the victim based on POC distribution
2. Locate all agents at their starting point in cell 0
3. Generate a danger map
4. Solve the problem for a time horizon of 27 seconds
5. Implement the *max-min* ADM for selecting a single solution
6. Simulate explorations in chronological order
   - If the victim was found, stop the mission and save the mission time
   - If the victim was not found simulate fire spread
     - If the victim was reached by the fire, stop the mission
7. If the victim was not detected by any agent or reached by the disaster, update the danger map with a new forecast and return to step 4

In this experiment, we kept equal many of the parameters adopted for experiments in previous section. The fire front speed was fixed on 10 km/h = 2.78 m/s, the cell size (ℓ x ℓ) was fixed on 24m x 24m and the fire step was set on 9 sec. We considered time windows of three fire steps, corresponding to 27 sec, and a single grid composed by 36 cells, representing regions of 20736 m^2. The values of σ_1, σ_2 and ρ: where set in 7, 15 and 0 respectively. Three different types of agents were considered as previous section. Their specifications are detailed in Table 6.1.

As can be noted, the time window adopted for this experiment correspond to the largest one tested in the previous section. Furthermore, the size of the search region is much larger than those tested before. Results from those previous experiments evidenced that the time required by Epsilon-constraint and Lexicographic methods is prohibitive for long size instances. For that reason, in the following experiment we only solved the problem by means of the PMAACO algorithm.

This experiment was intended to test the performance of our methodology given different levels of proximity between the victim and the fire. For that purpose, we computed the percentage of times that the victim was found alive and the time required to find the victim at each successful mission. Both computations were done for each one of the four scenarios.

Table 6.20: Statistics from complete missions

Scenario	**Effectiveness**	**Time (sec)**		
		Min	**Max**	**Avg**
1	62%	4.40	27.00	16.01
2	68%	6.68	26.96	17.61
3	66%	4.40	26.55	16.35
4	68%	6.95	26.96	18.64

Our methodology presented a stable performance through the experiment. The level of effectiveness remained between 60% and 70% , which represents an attractive performance in terms of saved lives. It can be noted that the times required to find the victim were shorter in Scenarios 1 and 3, where the victim was located nearer to the fire. Unfortunately, the percentage of saved lives is lower in Scenarios 1 and 3 than in Scenarios 2 and 4. This phenomenon occurs because, when the POC and the fire are closer among them, higher values of POC and Danger got overlapped. This makes efforts of the whole search team to be focused on a shorter number of cells than if the POC peak and the fire are located distant. At the same time, this reduces the number of chances that agents have to find the victim. In those scenarios, a failed detection is often fatal.

In contrast, Scenarios where POC peak and Fire are distant present higher chances of finding the victim alive, associated with a lower risk on its potential locations. However, in those circumstances, the search team is distributed over the search region from the beginning of the mission in order to cover both, danger and likely zones. This represents a lower number of agents searching on the surroundings of POC focus, which leads to higher times to decision.

The construction of the search plan for each single window in our experiment took between 22 and 30 seconds. Given the time window imposed of 27 seconds, this fact evidences the applicability of our methods in real search scenarios. The test of our methodology in physic simulated scenarios will be useful for verifying these findings.

Chapter 7

Conclusions and Future Research

This thesis studies how to conduct a search mission in scenarios where the search region is being affected by a dynamic threat. This problem has great practical relevance given that consistently, many search missions take place under these kind of circumstances. Despite the large amount of literature in Search Theory (more than a half century of production), as far as we know, this type of search problem was never mathematically solved before. To tackle this problem, we structured an adaptive planning and searching methodology, oriented to maintain updated information about the state of the threat throughout the entire mission. The main impact of this methodology is mitigating the risk of missing people to become critically injured by dynamic threats present in the search region. Such a reduction in risk, naturally implies an increment on the rate of saved lives during this kind of missions.

Within the scope of this study, we developed a mathematical formulation and implemented three solution methods for the problem, which determine the search path for a team of heterogeneous search agents possessing particular reliability profiles, travel times and exploration times. The formulation is a bi-objective programming model that prioritizes explorations in both, riskiest zones and locations where it is more likely to find the missing person. A vital component of the formulation is a Recursive Bayesian filter which updates the expectancy of finding the person at a given location, once an agent performs an unfruitful exploration there. That Bayesian filter is one of our major contributions to Search Theory, since several studies in this field were applying out of context expressions or inaccurate approximations that underestimate the expectancy of finding the missing person on a revised location.

Unfortunately, such a Bayesian filter is by nature a nonlinear expression that made our formulation a non-linear programming model. Seeking for a more convenient formulation, we fitted the Recursive Bayesian expression by Piecewise Linear approximation and thus, we obtained a linear programming formulation of the problem. Such a linear formulation is our second main contribution, since it provides a comparison mechanism to any alternative solution method that may be developed as part of future research.

Another advantage of our formulation is that it does not discretize the time as the majority of previous search planning models do. In contrast, our model controls the dynamics of the model by means of continuous variables, allowing totally free displacements of the agents through the search region at any moment required. As a consequence, our model takes a step forward in the realism captured by search planing models.

In order to validate our methods, we executed two sets of computational experiments. The first of them, was intended to compare the three solution methods, named Epsilon Constraint method, Lexicographic method and Pareto Multi-Agent Ant Colony Optimization (PMAACO). These methods were compared in terms of computational time and solution quality. We found that the Epsilon Constraint method and the PMAACO are both able to find very good approximations to the exact Pareto front. However, the applicability of the Epsilon Constraint method is limited to small instances, where both methods took less than two seconds to conclude. In contrast, the largest run of the Epsilon Constraint method took 15.1 hours, which is a clearly prohibitive time in the SAR context. In contrast, the PMAACO took 61.8 seconds to conclude in the same instance, showing its suitability for real size instances. By its side, the Lexicographic method was always able to find the two extremes of the Pareto front, which was in turn its main purpose. However, its running time for the same large instance was 29.1 minutes, limiting the applicability of this method to small size and medium size instances.

During our second set of experiments, we validated our adaptive planning methodology based on a rolling horizon approach. This time, we increased the size of the instance with respect to the first experiment, reduced the number of ants in the PMAACO algorithm, and simulated 200 missions, locating the missing person at different places of the search region. Our methodology proved to be robust and located the missing person before it was reached by the fire more than 60% of times. The average time required to find the person rounded the 16 seconds and remained under 27 seconds in all trials.

The scenarios considered during our experiments correspond to some of the largest instances tested until now for search problems. From our computational experience, we conclude that our methodology presents an effective solution to the problem of searching in presence of dynamic threats. Furthermore, we conclude that the PMAACO is the first algorithm in the literature able to deal with big size instances of this problem under a reasonable time.

Achieved Results and Products

Research articles:

- *Search Theory: A Taxonomic Literature Review.* Literature review of 83 articles in Search Theory - Submitted to the academic journal Computers & Operations Research
- Academic article exposing our methodology and results - Currently in edition

Conference speeches:

- *Búsqueda de Entidades en Presencia de un Desastre.* Conferencia en Logistica Social LS 2016 - Universidad del Norte, Barranquilla, Colombia.
- Searching for Entities Under Dynamic Emergencies. INFORMS Annual Meeting 2016 - Nashville, USA.

Future Research Lines

We consider the research topic addressed here to be very interesting. The execution of this research provided us with a lot of new knowledge in a field unexplored by us before. Our incursion in search planning, combined with the motivation of being solving a problem never considered in preceding literature, gives us multiple ideas for future research lines. A set of them are described below.

On the problem setting and assumptions

Extension to multiple targets and moving targets: In this thesis, we considered the case of a single stationary target. Given that many search scenarios involve multiple targets and moving targets, both considerations present interesting research avenues. Such a development may have its major impact on maritime search, where the mission often involve multiple targets being dragged by the oceanic currents.

Extension to agents prone to false alarm: Another assumption adopted here is that agents where not prone to advice false target sightings. This assumption may be valid under certain conditions, as in a search with pretty favorable environmental conditions. However, this may not be the case for a search in presence of a non-instantaneous disaster. In those cases, the phenomenon may have an effect on the lectures of the sensing devices employed by the search team. Therefore, a promising future research line consists on relaxing the assumption of false alarm free agents for search planning in the context of the OSPEAD.

Evaluation of alternative objective functions: The mathematical model provided here implements the following two objective functions:

- Maximization of the cumulative Danger collected by the search team
- Maximization of the cumulative Probability of Success collected by the search team

A future research line may consists on the validation of alternative objectives functions for the OSPEAD. As mentioned above, the OSPEAD may involve the influence of the disaster, causing distortions on the lectures of sensing devices. In this respect, a useful development would be to combine the two objectives adopted by us, with the maximization of the information gathered by the agents.

Another valid objective could be the minimization of the expected time to find the target. Such an objective may be suitable for entities presumed critically injured, that may require medical assistance in the shortest possible time.

Development of a formal sampling method for disaster monitoring: In this research we did not provide or employ any formal sampling method to simulate the collection of data about disaster's state. However, real search missions may involve limited resources to track the disaster and it may not be possible to achieve data from all over the search region. That constraint makes necessary, the development of formal sampling methods oriented to decide how to track the disaster. This research line is not only interesting, but also necessary for the application of any OSPEAD oriented methodology in real scenarios.

On the model

Disjunctive constraint pairs: The MILP developed in Section 4.2.6 was effectively solved during our experiments. Nonetheless, the processing time was clearly prohibitive for big instances of our algorithm. The induction of disjunctive constraint pairs in our model will significantly reduce the number of binary variables and constraints. It is known that the addition of integer variables in a model makes the processing time to increase exponentially. Then, in a reverse process, we propose to reduce the number of integer variables, seeking for exponential reduction on processing time.

On the solution methods

Local search: Local improvement heuristics were not evaluated in this thesis. We propose the application of local improvement algorithms for the following two purposes:

- Improving the solutions found by each colony in the PMAACO algorithm seeking for a reduction in the distance to the exact Pareto front in big instances. It may also allow to reduce the population size required to find good solutions and consequently, the processing time.
- Take extreme Pareto solutions found by the Lexicographic method, and search for additional Pareto efficient solutions in the region in a short time. This procedure may increase the diversity in the Pareto front found by this method.

Epsilon-constraint with meta-heuristics: The Epsilon-constraint method presented here was supported by the solution of the MILP problem developed in Section 4.2.6. The Epsilon-constraint methodology can also be supported by a meta-heuristic able to solve the mono-objective problem at each iteration. This research line will attempt to test the Epsilon-constraint method combined with different meta-heuristic methods such as Genetic Algorithms, Simulated Annealing or Ant Colony.

Diversification of construction procedure on PMAACO: In the PMAACO developed here, all the ants build their solution based on the same type of construction procedure. This research line would attempt to define multiple different construction procedures within the same PMAACO and distribute them over the ants. Some of those construction steps may be greedy. For instance, some ants may build their solution by selecting the most reliable partial agent and choosing next cells in its sequence until the agent become complete. The use of multiple construction procedures may help the algorithm to find attractive solutions on a short time. The combination of this diversification on the construction procedure with the application of local improvement is another promising research line.

On the simulation of the disaster

Enhancement of the disaster simulation method: As it was mentioned in Section 5.5, we encountered several difficulties in our search for a simulation tool available for our purposes. As far as we know, there exists advanced simulation methods based on the Cellular Automata approach implemented here. This research line would consist on the improvement of the disaster simulation method embedded in our algorithm. The availability of an accurate forecasting tool will be determinant on the effectiveness of the search methodology proposed here.

On the validation of the methodology

Comparison with current practice and SAR manuals: One of the most revealing experiments to test the performance of a new development is to compare it with the current practice and the manuals in the matter. A future step in development of the OSPEAD may involve the contrast between the procedures implemented nowadays by emergency response agencies and our methods. Such a procedure may evidence strengths in both approaches and potential opportunities of improvement.

Field tests: We are aware of the fact that implementing our methods in real scenarios require multiple future developments. Some of them involve the coordination between the systems that may take part in our search methodology, described in Section 4.1.1. However, we find it necessary to reach a cohesion between theoretical and practical developments in order to reach the desired objective of improving existent search methodologies. In this respect, we propose the validation of our methodologies in physical simulated search scenarios as a future research avenue.

References

Abi-Zeid, I., & Frost, J. R. (2005). Sarplan: A decision support system for canadian search and rescue operations. *European Journal of Operational Research*, *162*(3), 630–653.

Ablavsky, V., & Snorrason, M. (2000). Optimal search for a moving target: A geometric approach. In *Aiaa guidance, navigation, and control conference and exhibit* (p. 4060).

Adams, S. M., & Friedland, C. J. (2011). *A survey of unmanned aerial vehicle (uav) usage for imagery collection in disaster research and management.* publisher not identified.

Assaf, D., & Sharlin-Bilitzky, A. (1994). Dynamic search for a moving target. *Journal of applied probability*, 438–457.

Baiocchi, V., Dominici, D., Milone, M. V., & Mormile, M. (2013). Development of a software to plan uavs stereoscopic flight: An application on post earthquake scenario in laquila city. In *Computational science and its applications–iccsa 2013* (pp. 150–165). Springer.

Bekmezci, I., Sahingoz, O. K., & Temel, Ş. (2013). Flying ad-hoc networks (fanets): A survey. *Ad Hoc Networks*, *11*(3), 1254–1270.

Beltagy, M., & Abd Allah El-Hadidy, M. (2013). Parabolic spiral search plan for a randomly located target in the plane. *ISRN Mathematical Analysis*, *2013*.

Benkoski, S. J., Monticino, M. G., & Weisinger, J. R. (1991). A survey of the search theory literature. *Naval Research Logistics (NRL)*, *38*(4), 469–494.

Berger, J., & Lo, N. (2015). An innovative multi-agent search-and-rescue path planning approach. *Computers & Operations Research*, *53*, 24–31.

Berger, J., Lo, N., & Barkaoui, M. (2016). Static target search path planning optimization with heterogeneous agents. *Annals of Operations Research*, 1–18.

Berger, J., Lo, N., & Noel, M. (2013). Exact solution for search-and-rescue path planning. *International Journal of Computer and Communication Engineering*, *2*(3), 266.

Berger, J., Lo, N., & Noel, M. (2014). A new multi-target, multi-agent search-and-rescue path planning approach. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, *8*(6), 978–987.

Bonnie, D., Candido, S., Bretl, T., & Hutchinson, S. (2012). Modelling search with a binary sensor utilizing self-conjugacy of the exponential family. In *Robotics and automation (icra), 2012 ieee international conference on* (pp. 3975–3982).

Borghetti, A., D'Ambrosio, C., Lodi, A., & Martello, S. (2008). An milp approach for short-term hydro scheduling and unit commitment with head-dependent reservoir. *IEEE Transactions on Power Systems*, *23*(3), 1115–1124.

Bourgault, F., Furukawa, T., & Durrant-Whyte, H. E. (2003). Coordinated decentralized search for a lost target in a bayesian world. In *Intelligent robots and systems, 2003.(iros 2003). proceedings. 2003 ieee/rsj international conference on* (Vol. 1, pp. 48–53).

Bourgault, F., Furukawa, T., & Durrant-Whyte, H. F. (2004a). Decentralized bayesian negotiation for cooperative search. In *Intelligent robots and systems, 2004.(iros 2004). proceedings. 2004 ieee/rsj international conference on* (Vol. 3, pp. 2681–2686).

Bourgault, F., Furukawa, T., & Durrant-Whyte, H. F. (2004b). Process model, constraints, and the coordinated search strategy. In *Robotics and automation, 2004. proceedings. icra'04. 2004 ieee international conference on* (Vol. 5, pp. 5256–5261).

Brown, S. S. (1980). Optimal search for a moving target in discrete time and space. *Operations Research*, *28*(6), 1275–1289.

Carpin, S., Burch, D., Basilico, N., Chung, T. H., & Kölsch, M. (2013). Variable resolution search with quadrotors: Theory and practice. *Journal of Field Robotics*, *30*(5), 685–701.

Carpin, S., Burch, D., & Chung, T. H. (2011). Searching for multiple targets using probabilistic quadtrees. In *Intelligent robots and systems (iros), 2011 ieee/rsj international conference on* (pp. 4536–4543).

Casbeer, D. W., Beard, R. W., McLain, T. W., Li, S.-M., & Mehra, R. K. (2005). Forest fire monitoring with multiple small uavs. In *Proceedings of the 2005, american control conference, 2005.* (pp. 3530–3535).

Charnes, A., & Cooper, W. W. (1958). The theory of search: optimum distribution of search effort. *Management Science*, *5*(1), 44–50.

Chrobocinski, P., Makri, E., Zotos, N., Stergiopoulos, C., & Bogdos, G. (2012). Darius project: Deployable sar integrated chain with unmanned systems. In *Telecommunications and multimedia (temu), 2012 international conference on* (pp. 220–226).

Chung, T. H. (2009). On probabilistic search decisions under searcher motion constraints. In *Algorithmic foundation of robotics viii* (pp. 501–516). Springer.

Chung, T. H., & Burdick, J. W. (2007). A decision-making framework for control strategies in probabilistic search. In *Robotics and automation, 2007 ieee international conference on* (pp. 4386–4393).

Chung, T. H., & Burdick, J. W. (2008). Multi-agent probabilistic search in a sequential decision-theoretic framework. In *Robotics and automation, 2008. icra 2008. ieee international conference on* (pp. 146–151).

Chung, T. H., & Burdick, J. W. (2012). Analysis of search decision making using probabilistic search strategies. *Robotics, IEEE Transactions on*, *28*(1), 132–144.

Chung, T. H., & Carpin, S. (2011). Multiscale search using probabilistic quadtrees. In *Robotics and automation (icra), 2011 ieee international conference on* (pp. 2546–2553).

Chung, T. H., Kress, M., & Royset, J. O. (2009). Probabilistic search optimization and mission assignment for heterogeneous autonomous agents. In *Robotics and automation, 2009. icra'09. ieee international conference on* (pp. 939–945).

Collins, G. E., Riehl, J. R., & Vegdahl, P. S. (2007). A uav routing and sensor control optimization algorithm for target search. In *Defense and security symposium* (pp. 65610D–65610D).

Dambreville, F., & Le Cadre, J.-P. (2002). Detection of a markovian target with optimization of the search efforts under generalized linear constraints. *Naval Research Logistics (NRL)*, *49*(2), 117–142.

Dambreville, F., & Le Cadre, J.-P. (2004). Spatio-temporal multi-mode information management for moving target detection. *Information Fusion*, *5*(3), 169–178.

Daou, M. P., Cabal, A., Coulet, C., Bertrand, O., Blayo, E., Rousseau, A., & Degroof, A. (2014). Modelling crisis management for improved action and preparedness (crisma): Modelling submersion on the charente-maritime coast. In *Colloque international ≪connaissance et compréhension des risques côtiers: Aléas, enjeux, représentations, gestion≫.*

DasGupta, B., Hespanha, J. P., & Sontag, E. (2004). Aggregation-based approaches to honey-pot searching with local sensory information. In *American control conference, 2004. proceedings of the 2004* (Vol. 2, pp. 1202–1207).

Dell, R. F., Eagle, J. N., Alves Martins, G. H., & Garnier Santos, A. (1996). Using multiple searchers in constrained-path, moving-target search problems. *Naval Research Logistics*, *43*(4), 463–480.

Dobbie, J. M. (1968). A survey of search theory. *Operations Research*, *16*(3), 525–537.

Dodin, P., Minvielle, P., & Le Cadre, J.-P. (2007). A branch-and-bound algorithm applied to optimal radar search pattern. *Aerospace science and technology*, *11*(4), 279–288.

Dondo, R., & Cerdá, J. (2007). A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research*, *176*(3), 1478–1507.

Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical computer science*, *344*(2), 243–278.

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, *1*(1), 53–66.

Dorigo, M., & Stützle, T. (2010). Ant colony optimization: overview and recent advances. In *Handbook of metaheuristics* (pp. 227–263). Springer.

Eagle, J. N., & Yee, J. R. (1990). An optimal branch-and-bound procedure for the constrained path, moving target search problem. *Operations research*, *38*(1), 110–114.

Enslow, P. H. (1966). A bibliography of search theory and reconnaissance theory literature. *Naval Research Logistics Quarterly*, *13*(2), 177–202.

Favaretto, D., Moretti, E., & Pellegrini, P. (2007). Ant colony system for a vrp with multiple time windows and multiple visits. *Journal of Interdisciplinary Mathematics*, *10*(2), 263–284.

Flushing, E. F., Gambardella, L., & Caro, G. (2012). Gis-based mission support system for wilderness search and rescue with heterogeneous agents. In *Iros12 workshop on robots and sensors integration in future rescue information system (rosin12).*

Flushing, E. F., Gambardella, L. M., & Di Caro, G. A. (2013). Strategic control of proximity relationships in heterogeneous search and rescue teams. *arXiv preprint arXiv:1312.4601*.

Flushing, E. F., Gambardella, L. M., & Di Caro, G. A. (2014). A mathematical programming approach to collaborative missions with heterogeneous teams. In *2014 ieee/rsj international conference on intelligent robots and systems* (pp. 396–403).

Forsmo, E. J., Grøtli, E. I., Fossen, T. I., & Johansen, T. A. (2013). Optimal search mission with unmanned aerial vehicles using mixed integer linear programming. In *Unmanned aircraft systems (icuas), 2013 international conference on* (pp. 253–259).

Frew, E. W., & Brown, T. X. (2009). Networking issues for small unmanned aircraft systems. *Journal of Intelligent and Robotic Systems*, *54*(1-3), 21–37.

FrontlineSolvers. (n.d.). *Optimization problem types - linear and quadratic programming.* http://www.solver.com/linear-quadratic-programming. (Accessed August 15, 2016)

Frost, J. (1998). The theory of search: a simplified explanation. *Report by Soza & Company Ltd. and Office of Search and Rescue US Coast Guard*.

Frost, J., & Stone, L. D. (2001). *Review of search theory: advances and applications to search and rescue decision support* (Tech. Rep.). DTIC Document.

Furukawa, T., Mak, L. C., Durrant-Whyte, H., & Madhavan, R. (2012). Autonomous bayesian search and tracking, and its experimental validation. *Advanced Robotics*, *26*(5-6), 461–485.

Gan, S. K., Fitch, R., & Sukkarieh, S. (2012). Real-time decentralized search with inter-agent collision avoidance. In *Robotics and automation (icra), 2012 ieee international conference on* (pp. 504–510).

Golden, B. L., Levy, L., & Vohra, R. (1987). The orienteering problem. *Naval research logistics*, *34*(3), 307–318.

Goodrich, M. A., Morse, B. S., Gerhardt, D., Cooper, J. L., Quigley, M., Adams, J. A., & Humphrey, C. (2008). Supporting wilderness search and rescue using a camera-equipped mini uav. *Journal of Field Robotics*, *25*(1-2), 89–110.

Grundel, D. A. (2005). Constrained search for a moving target. In *Collaborative technologies and systems, 2005. proceedings of the 2005 international symposium on* (pp. 327–332).

Hirokawa, R., Kubo, D., Suzuki, S., Meguro, J.-i., & Suzuki, T. (2007). A small uav for immediate hazard map generation. In *Aiaa infotech@ aerospace conf.*

Ho, D.-T., Grøtli, E. I., Sujit, P., Johansen, T. A., & Borges Sousa, J. (2013). Cluster-based communication topology selection and uav path planning in wireless sensor networks. In *Unmanned aircraft systems (icuas), 2013 international conference on* (pp. 59–68).

Hohzaki, R. (2006). Search allocation game. *European Journal of Operational Research*, *172*(1), 101–119.

Hohzaki, R., & Iida, K. (1997). Optimal strategy of route and look for the path constrained search problem with reward criterion. *European journal of operational research*, *100*(1), 236–249.

Hohzaki, R., & Iida, K. (2000). A search game when a search path is given. *European Journal of Operational Research*, *124*(1), 114–124.

Hohzaki, R., & Iida, K. (2001). Optimal ambushing search for a moving target. *European Journal of Operational Research*, *133*(1), 120–129.

Hong, S.-P., Cho, S.-J., & Park, M.-J. (2009). A pseudo-polynomial heuristic for path-constrained discrete-time markovian-target search. *European Journal of Operational Research*, *193*(2), 351–364.

Hong, S.-P., Cho, S.-J., Park, M.-J., & Lee, M.-G. (2009). Optimal search-relocation trade-off in markovian-target searching. *Computers & Operations Research*, *36*(6), 2097–2104.

Hubenko, A., Fonoberov, V. A., Mathew, G., & Mezić, I. (2011). Multiscale adaptive search. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *41*(4), 1076–1087.

Iida, K., Hohzaki, R., & Sakamoto, T. (2002). An optimal distribution of searching effort relaxing the assumption of local effectiveness. *Journal of the Operations Research Society of Japan-Keiei Kagaku*, *45*(1), 13–26.

Israel, M., Khmelnitsky, E., & Kagan, E. (2012). Search for a mobile target by ground vehicle on a topographic terrain. In *Electrical & electronics engineers in israel (ieeei), 2012 ieee 27th convention of* (pp. 1–5).

Jotshi, A., & Batta, R. (2008). Search for an immobile entity on a network. *European Journal of Operational Research*, *191*(2), 347–359.

Kassem, M. A. E.-H., & El-Hadidy, M. A. A. (2014). Optimal multiplicative bayesian search for a lost target. *Applied Mathematics and Computation*, *247*, 795–802.

Khan, A., Yanmaz, E., & Rinner, B. (2014). Information merging in multi-uav cooperative search. In *2014 ieee international conference on robotics and automation (icra)* (pp. 3122–3129).

Koopman, B. (1956a). The theory of search 1. kinematic bases. *Operations research*, *4*(3), 324–346.

Koopman, B. (1956b). The theory of search 2. target detection. *Operations research*, *4*(5), 503–531.

Koopman, B. O. (1946). *Search and screening.* Operations Evaluation Group, Office of the Chief of Naval Operations, Navy Department.

Koopman, B. O. (1957). The theory of search 3. the optimum distribution of searching effort. *Operations research*, *5*(5), 613–626.

Koopman, B. O. (1979). Search and its optimization. *The American Mathematical Monthly*, *86*(7), 527–540.

Kriheli, B., Levner, E., Bendersky, M., & Yakubov, E. (2015). A fast algorithm for scheduling detection-and-rescue operations based on data from wireless sensor networks. *Research in Computing Science*, *104*, 9–21.

Kwok, A., & Holsapple, R. (2011). Approximate decentralized sensor fusion for bayesian search of a moving target. In *Aiaa infotech@ aerospace conference.*

Lanillos, P., Besada-Portas, E., Lopez-Orozco, J. A., & de la Cruz, J. M. (2014). Minimum time search in uncertain dynamic domains with complex sensorial platforms. *Sensors*, *14*(8), 14131–14179.

Lanillos, P., Besada-Portas, E., Pajares, G., & Ruz, J. J. (2012). Minimum time search for lost targets using cross entropy optimization. In *Intelligent robots and systems (iros), 2012 ieee/rsj international conference on* (pp. 602–609).

Lanillos, P., Gan, S. K., Besada-Portas, E., Pajares, G., & Sukkarieh, S. (2014). Multi-uav target search using decentralized gradient-based negotiation with expected observation. *Information Sciences*, *282*, 92–110.

Lau, H., Huang, S., & Dissanayake, G. (2005). Optimal search for multiple targets in a built environment. In *Intelligent robots and systems, 2005.(iros 2005). 2005 ieee/rsj international conference on* (pp. 3740–3745).

Lau, H., Huang, S., & Dissanayake, G. (2006). Probabilistic search for a moving target in an indoor environment. In *2006 ieee/rsj international conference on intelligent robots and systems* (pp. 3393–3398).

Laumanns, M., Thiele, L., & Zitzler, E. (2006). An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, *169*(3), 932–942.

Lee, S., & Morrison, J. R. (2015). Decision support scheduling for maritime search and rescue planning with a system of uavs and fuel service stations. In *Unmanned aircraft systems (icuas), 2015 international conference on* (pp. 1168–1177).

Le Thi, H. A., Nguyen, D. M., & Dinh, T. P. (2014). A dc programming approach for planning a multisensor multizone search for a target. *Computers & Operations Research*, *41*, 231–239.

Lin, L., & Goodrich, M. A. (2009). Uav intelligent path planning for wilderness search and rescue. In *2009 ieee/rsj international conference on intelligent robots and systems* (pp. 709–714).

Lo, N., Berger, J., & Noel, M. (2012). Toward optimizing static target search path planning. In *Computational intelligence for security and defence applications (cisda), 2012 ieee symposium on* (pp. 1–7).

Lukka, M. (1977). On the optimal searching tracks for a moving target. *SIAM Journal on Applied Mathematics*, *32*(1), 126–132.

Mohan, B. C., & Baskaran, R. (2012). A survey: Ant colony optimization based recent research and implementation on several engineering domain. *Expert Systems with Applications*, *39*(4), 4618–4627.

Morin, M., Abi-Zeid, I., Lang, P., Lamontagne, L., & Maupin, P. (2009). The optimal searcher path problem with a visibility criterion in discrete time and space. In *Information fusion, 2009. fusion'09. 12th international conference on* (pp. 2217–2224).

Morin, M., Lamontagne, L., Abi-Zeid, I., & Maupin, P. (2010). The ant search algorithm: An ant colony optimization algorithm for the optimal searcher path problem with visibility. In *Canadian conference on artificial intelligence* (pp. 196–207).

Nguyen, J., Lawrance, N., Fitch, R., & Sukkarieh, S. (2013). Energy-constrained motion planning for information gathering with autonomous aerial soaring. In *Robotics and automation (icra), 2013 ieee international conference on* (pp. 3825–3831).

Owen, G., & McCormick, G. H. (2008). Finding a moving fugitive. a game theoretic representation of search. *Computers & Operations Research*, *35*(6), 1944–1962.

Rardin, R. L. (1998). *Optimization in operations research* (Vol. 166). Prentice Hall Upper Saddle River, NJ.

Royset, J. O., & Sato, H. (2010). Route optimization for multiple searchers. *Naval Research Logistics (NRL)*, *57*(8), 701–717.

Saito, M., Hatanaka, T., & Fujita, M. (2010). Order formations in multi-agent search problem: A game theoretic approach. In *American control conference (acc), 2010* (pp. 4768–4773).

Saito, M., Hatanaka, T., & Fujita, M. (2011). Decision dynamics in cooperative search based on evolutionary game theory. *Communications in Information and Systems*, *11*(1), 57.

Santos, A. G. (1993). *Using multiple searchers to locate a randomly moving target* (Unpublished doctoral dissertation). Monterey, California. Naval Postgraduate School.

Sarmiento, A., Murrieta, R., & Hutchinson, S. A. (2003). An efficient strategy for rapidly finding an object in a polygonal world. In *Intelligent robots and systems, 2003.(iros 2003). proceedings. 2003 ieee/rsj international conference on* (Vol. 2, pp. 1153–1158).

Schilde, M., Doerner, K. F., Hartl, R. F., & Kiechle, G. (2009). Metaheuristics for the bi-objective orienteering problem. *Swarm Intelligence*, *3*(3), 179–201.

Scott, J. H. (2012). Introduction to wildfire behavior modeling. *National Interagency Fules, Fire, &Vegetation Technology Transfer., Wild Fire Managment RD&A*, 7–15.

Simonin, C., Cadre, J.-P. L., & Dambreville, F. (2008). A common framework for multitarget search and cross-cueing optimization. In *Information fusion, 2008 11th international conference on* (pp. 1–8).

Simonin, C., Le Cadre, J.-P., & Dambreville, F. (2009). A hierarchical approach for planning a multisensor multizone search for a moving target. *Computers & Operations Research*, *36*(7), 2179–2192.

Song, N.-O., & Teneketzis, D. (2004). Discrete search with multiple sensors. *Mathematical Methods of Operations Research*, *60*(1), 1–13.

Stone, L. D. (1977). Search for targets with generalized conditionally deterministic motion. *SIAM Journal on Applied Mathematics*, *33*(3), 456–468.

Stone, L. D. (1979). Necessary and sufficient conditions for optimal search plans for moving targets. *Mathematics of Operations Research*, *4*(4), 431–440.

Stone, L. D. (1989). Or forumwhat's happened in search theory since the 1975 lanchester prize? *Operations Research*, *37*(3), 501–506.

Sun, C., & Duan, H. (2012). A restricted-direction target search approach based on coupled routing and optical sensor tasking optimization. *Optik-International Journal for Light and Electron Optics*, *123*(24), 2226–2229.

Waharte, S., Symington, A. C., & Trigoni, N. (2010). Probabilistic search with agile uavs. In *Icra* (pp. 2840–2845).

Waharte, S., & Trigoni, N. (2010). Supporting search and rescue operations with uavs. In *Emerging security technologies (est), 2010 international conference on* (pp. 142–147).

Wang, H., & Zhou, H. (2015). Searching for a target traveling between a hiding area and an operating area over multiple routes. *American Journal of Operations Research*, *5*(04), 258.

Washburn, A. R. (1997). Branch and bound methods for a search problem.

Wex, F., Schryen, G., Feuerriegel, S., & Neumann, D. (2014). Emergency response in natural disaster management: Allocation and scheduling of rescue units. *European Journal of Operational Research*, *235*(3), 697–708.

Wong, E.-M., Bourgault, F., & Furukawa, T. (2005). Multi-vehicle bayesian search for multiple lost targets. In *Robotics and automation, 2005. icra 2005. proceedings of the 2005 ieee international conference on* (pp. 3169–3174).

Yan, I., & Blankenship, G. (1988). Numerical methods in search path planning. In *Decision and control, 1988., proceedings of the 27th ieee conference on* (pp. 1563–1569).

Yetkin, H., Lutz, C., & Stilwell, D. (2016). Environmental information improves robotic search performance. *arXiv preprint arXiv:1607.05302*.

Yu, W., & Batta, R. (2010). Search for an immobile entity on a unit graph: A mathematical programming approach.