Politecnico di Milano

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING
Master of Science – Space Engineering

Multi-objective Pareto Monte Carlo Tree Search for informative path planning of planetary rovers

Supervisor
Prof. Mauro Massari

Candidate
Lorenzo Pisani – 920419

Academic Year 2020 – 2021

Abstract

Planetary exploration rovers represent an essential tool for gathering and studying scientific information on the surface of other planets and satellites. However, these vehicles are, currently, controlled by operators on ground, since they do not have enough autonomy to plan their operations in safety. Nevertheless, the communication is strongly constrained by delays and limited time windows, leading to a very slow exploration process, in which the vehicle awaits for commands from ground for long periods of time.

In this context, the work of this thesis aims to develop a system capable of increasing the decision making autonomy of rovers, in order to get rid of the time constraints related to communication and to maximise their possibilities of exploration.

In particular, a navigation method has been studied for a multi-agent system composed by a helicopter/drone, used as a scout to map the area of interest, and a rover. The efforts have, then, been focused on the development of a multi-objective path planning system able to determine, based on the data previously gathered by the helicopter, a trajectory that could evaluate simultaneously competing objectives such as the maximisation of the scientific return and the optimisation of the path.

For global planning, which is the true decision making part of the system, a multi-objective Monte Carlo Tree Search (MCTS) based on the generation of a Pareto front has been developed.

On a local level, the artificial potential fields method has been used to determine the path among obstacles and dangerous areas. The problem of the generation of local minima, that usually arises with this method, has been dealt with the introduction of asymmetric vortex fields, that have been summed up to the repulsive ones used for obstacles.

Finally, a sensitivity analysis has been carried out to verify efficiency, quality and safety of the trajectories proposed by the system.

Sommario

I rover per l'esplorazione planetaria sono uno degli strumenti fondamentali per lo studio e la raccolta di informazioni scientifiche sulla superficie di altri pianeti e satelliti. Attualmente, però, la loro gestione è controllata da operatori a Terra, in quanto tali veicoli non possiedono abbastanza autonomia per poter pianificare le loro operazioni in sicurezza. Tuttavia, poichè la comunicazione è fortemente vincolata da ritardi e finestre limitate, il processo di esplorazione risulta molto lento, in quanto il rover rimane fermo in attesa di comandi da Terra per lunghi periodi di tempo.

Il presente lavoro di tesi si inserisce in questo contesto e ha l'intento di sviluppare un sistema capace di aumentare l'autonomia decisionale dei rover, al fine di eliminare i vincoli temporali legati alla comunicazione e massimizzarne, così, le possibilità di esplorazione.

In particolare, è stato studiato un metodo di navigazione per un sistema multi-agente composto da un elicottero/drone, utilizzato come scout per mappare l'area di interesse, e da un rover. Il lavoro si è, poi, incentrato sullo sviluppo di un sistema di pianificazione multi-obiettivo del percorso del rover che potesse, sulla base delle informazioni rilevate dall'elicottero, determinare una traiettoria considerando contemporaneamente obiettivi in contrasto tra loro quali la massimizzazione del ritorno scientifico e l'ottimizzazione del percorso.

Per la pianificazione globale, che rappresenta il vero elemento decisionale del metodo, è stato elaborato un sistema di ricerca multi-obiettivo ad albero Monte Carlo basato sulla generazione del cosiddetto fronte di Pareto.

A livello locale, invece, il metodo dei potenziali artificiali è stato utilizzato per generare una traiettoria tra ostacoli e zone potenzialmente pericolose. Il problema dei minimi locali, solitamente legato a questo tipo di sistema, è stato affrontato con l'introduzione di potenziali asimmetrici a vortice, che sono stati sommati a quelli repulsivi degli ostacoli.

Infine, è stata condotta un'analisi di sensitività e robustezza per verificare l'efficienza, la qualità e la sicurezza delle traiettorie proposte dal sistema.

Contents

List of Figures

List of Tables

Acronyms

ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
ASTIA	Autonomous Science Target Identification and Acquisition
BN	Bayesian Networks
CPP	Coverage Path Planning
ESA	European Space Agency
FIRAS	Force Inducing an Artificial Repulsion from the Surface
MCTS	Monte Carlo Tree Search
MER	Mars Exploration Rover
NASA	National Aeronautics and Space Administration
NTZ	Non-Traversable Zone
SARA	Science Assessment and Response Agent
SVM	Support Vector Machine
SVS	Scientific Value Score
TGO	Trace Gas Orbiter
UAV	Unmanned Aerial Vehicle
UCB	Upper Confidence Bound
UGV	Unmanned Ground Vehicle

Chapter 1

Introduction

1.1 Autonomous planetary exploration rovers

Planetary rovers represent a key feature for the exploration of other planets and satellites, as they are able to collect scientific information on ground where men are not yet able to land.

However, their rate of exploration is very slow as they require constant human supervision for decision making, which is heavily constrained by delays and limited communication windows. In 2022, European Space Agency (ESA) is going to launch the ExoMars rover, Rosalind Franklin (Figure 1.1), which will sample the subsurface of Mars in search for traces of life. The Trace Gas Orbiter (TGO), launched in 2016 as first part of the ExoMars mission, will serve as relay for the communication between the rover and the ground. This communication, though, may only happen twice per Sol (Martian day) due to TGO's orbital trajectory and constraints on the allocation of deep space antennas [1]. It is clear how this leads to a very slow exploration process, in which the rover awaits for commands from ground for long periods of time.

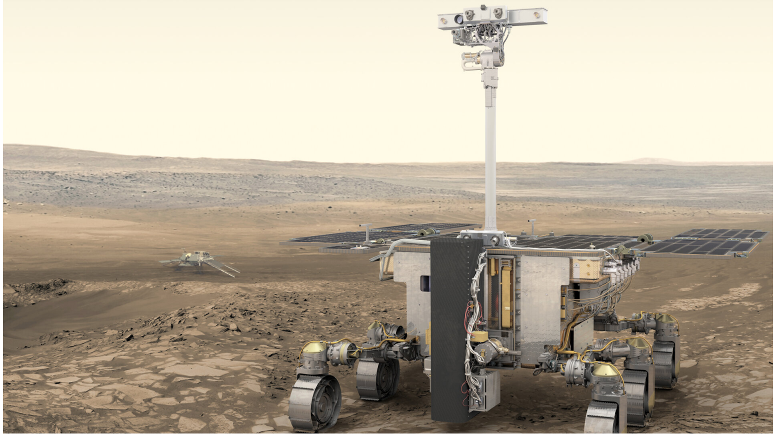


Figure 1.1. Rosalind Franklin rover (ExoMars, ESA)

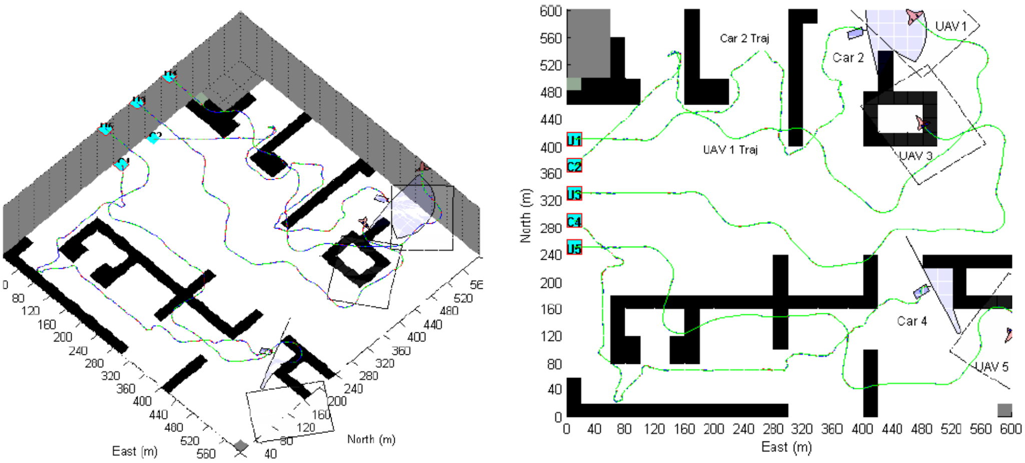
Of course, safety is the major concern for these types of missions, given the amount of work, time and resources invested and it is one of the main reasons why the navigation planning is carried out under human supervision. Avoiding risks results in longevity of missions and extended exploration opportunities.
An autonomous navigation system that could prevent the rover from interacting so frequently with ground and with the capability of keeping the rover safe, thanks to an efficient obstacle avoidance system, would have a great impact on the exploration rate of planetary rovers. Meaning that with one single mission the explored land of the planet or satellite, along with the scientific return, could be way larger.

1.2 Multi-agent decoupled path planning

The aim of this work is to define a method that could make use of the scientific information, obtained by a satellite or a helicopter/drone in an area of interest, to plan the exploration of this zone in an autonomous way, without the need of human intervention (or at least reducing it).
The helicopter case is of particular interest, since NASA's mission to Mars Perseverance, launched in July 2020 and successfully landed in February 2021, is going to carry out, for the first time, some experimental flights of a Mars Helicopter, Ingenuity (Figure 1.2).

Figure 1.2. Mars Helicopter Ingenuity (Perseverance, NASA)

Therefore, the possibility of exploiting a multi-agent coordinated path planning, such as the one presented in [2], combining a helicopter and a rover to explore an unknown area has been analysed. In the quoted study, the ground and aerial vehicles are considered as comparable, except for their ability of overcoming obstacles, as can be seen in Figure 1.3.
However, the flight endurance of the Mars Helicopter is very limited (about 90 s)



(a) (b)

Figure 1.3. (a) 3D and (b) 2D trajectories of an exploration team of 2 UGV and 3 UAV [2]

and also for its improved version, called Mars Science Helicopter and currently under development at NASA, is still less than 7 min [3] [4]. The helicopter could cover large distances in a small amount of time, while the rover is much slower. As a consequence, it would not be reasonable to consider the two vehicles as similar in their motion. Moreover, the cooperative exploration would make more sense if the two vehicles were carrying the same instruments on board, so that each scientific target could be visited indiscriminately either by the ground or the aerial vehicle, as in the framework proposed in [5]. However, this is not the case since, in general, they will definitely be equipped with different kind of instruments.

For these reasons, the concept of the mission will be to utilise the helicopter to map the area with cameras and scientific instruments able to detect useful information, that will be then used by the rover for its path planning. A similar solution, in which the images taken from an aerial vehicle are used to improve the path planning of a ground vehicle, is explained in [6].

In this way, as shown in Figure 1.4, the rover path planning and exploration phase will take place after the helicopter has landed and the gathered information has been processed ($T_0 \rightarrow T_h$: helicopter phase, $T_h \rightarrow (T_h + T_r)$: rover phase). Therefore, the multi-agent path planning can be decoupled, which means that it will be performed separately for helicopter and rover, as their exploration will not be coordinated.

Moreover, since the helicopter will explore an unknown or little known area (only satellite imagery may be available) there will not be any privileged direction or sub-area to investigate. So, the helicopter should perform a Coverage Path Planning (CPP), that consists of finding the route which covers every point of a

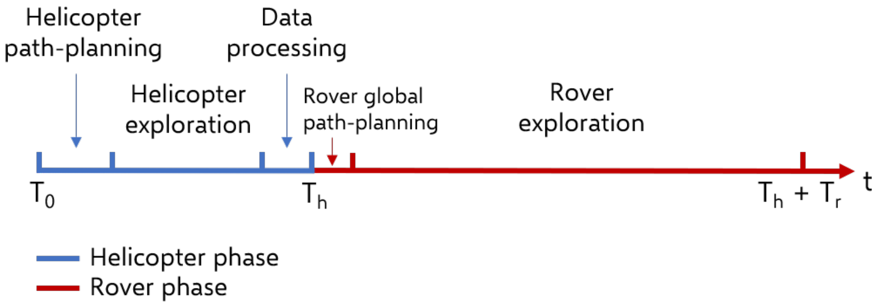


Figure 1.4. Decoupled team exploration timeline

certain area of interest.
Many studies have been carried out in this field in order to find ways to define efficient trajectories that could cover uniformly a given area. The article in [7] explores and analyses the existing studies in the literature related to the different approaches employed in CPP problems. Geometric flight patterns based on energy, such as the energy-aware spiral described in [8], and more complex grid-based solutions are explained. The review also considers different shapes of the area of interest. Some examples of these CPP methods are depicted in Figure 1.5.

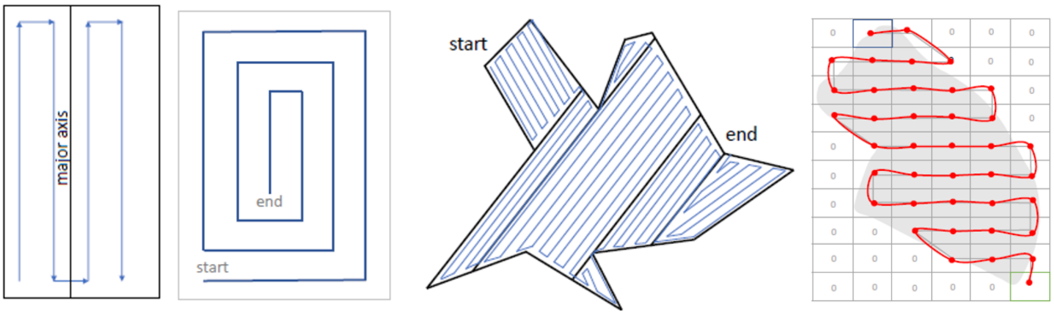


Figure 1.5. Coverage Path Planning examples [7]

Eventually, it has been decided to focus only on the more interesting and relevant rover path planning, with the hypothesis of already having the map generated with the information detected by a helicopter/drone or by a satellite.

1.3 Multi-objective path planning system

In the context of increasing the autonomy in decision making of planetary rovers, this thesis proposes an autonomous path planning method that aims to the maximisation of the scientific return of the mission.
The planning will be, at first, based on the information gathered by a helicopter or by a satellite, which will then be updated and fused with the new data that

the rover will find out during its route.

In addition to the maximisation of science, the method should also consider other important parameters such as the optimisation of the path. Indeed, it would be interesting to combine these two competing requirements trying to maximise the scientific gain and, at the same time, generate a smart short route for the rover to follow.

In general, there are two main alternatives that may be examined to face this problem. The first option is to consider it as a constrained optimisation problem, by defining a limit on the resources (such as a maximum path length or energy) and optimising the scientific value only. Instead, the second possibility may be to use a method that could perform an optimisation over multiple objective functions, such as scientific return and path length or energy used.

In literature, the most common approaches that are used for decision making involve the formation of a tree, which is explored by the algorithm in order to find the most promising solutions. Some examples of these methods are the branch and bound approach and the Monte Carlo Tree Search (MCTS).

Regarding the constrained optimisation problem, different solutions have been proposed. In the article in [9], the robot is constrained to a total sensing budget, which includes the energy of both its movements and of its sensing actions (different sensing modalities are evaluated). In this paper, the MCTS is used to plan paths and sensing actions that maximise the information gained in respect of the energetic constraint. A path planning algorithm that makes use of a branch and bound approach to coordinate multiple robots, each having a resource constraint, to maximise the "informativeness" of their visited locations is proposed in [10]. Actually, tree algorithms are not the only tool that has been chosen as a solution to this constrained optimisation problem. For instance, dynamic programming has been used in [11] to plan data collection tours of a robot with an energy constraint.

In relation to the second type of approach, the most commonly used method is the multi-objective Monte Carlo Tree Search. The authors of the papers in [12] and [13] both use a MCTS combined with the generation of a Pareto front to manage the multi-objectivity of the problem. A very interesting aspect of this approach is that, once it is defined how to deal with two objective functions, the algorithm could be easily extended to more than two competing objectives and, so, it may be implemented for more complex decision making problems.

In this study, as it will be better explained in section 2.4, the planning system has been divided into a global and a local segment. Global planning aims to determine the waypoints that the rover must follow to achieve the mission objectives and it represents the actual decision making part of the method. Therefore, for the reasons previously explained, it has been decided to use the multi-objective Pareto MCTS for the development of this segment.

On a local level, instead, a system based on artificial potential fields has been designed to determine the actual path between consecutive waypoints and among potential hazards. The problems related to the generation of local minima, that often come with this method, have been solved with the definition of asymmetric swirling potential fields for the repulsive sources.

1.4 Structure of the thesis

The thesis is organised as follows.
Chapter 2 presents the assumptions defined for the model and the description of the scenario of work with the rover concept of operations, along with an overview of the structure of the selected method in all its parts.
In chapter 3, the multi-objective Pareto Monte Carlo Tree Search used for global path planning is described in detail with its four steps of selection, expansion, simulation and back-propagation.
Chapter 4 explains the artificial potential fields method used for local path planning for the definition of the actual trajectory of the rover around obstacles. The potential fields models utilised are presented as well.
In chapter 5, the results of the application of the studied method to different simulation environments are reported, together with a sensitivity analysis that verifies its effectiveness.
Lastly, chapter 6 presents the conclusions and some possible future developments.

Chapter 2

Description of the model

2.1 Model assumptions

Before the actual set up of the path planning method, it is necessary to properly define the operational framework by establishing some assumptions that will be considered in the model. These assumptions could be broken down into two different categories: some general ones, related to the definition of the path planning method, and some hypotheses on the rover itself. A summary of these assumptions can be found in Table 2.1 and Table 2.2.

General assumptions

First of all, the human level of intervention has been defined to be possible only at high level, in order to increase as much as possible the autonomy of the rover. This means that ground control will decide the long-term direction of exploration, by selecting the ending point within an exploration area. This last point, will then become the starting point of the following exploration sector. Therefore, at least satellite images should be available to ground control, so that the ending point can be selected. It must be said, however, that the algorithm may work also without human intervention, as the rover could select one of the scientific points as its final target. Besides, the role of ground control is crucial from the long-term point of view. Indeed, the algorithm studied in the context of this thesis acts within a limited specific area (already explored by the helicopter/drone). It is the ground control, then, that will define which is going be the next area to be explored by setting, as said, the overall direction of exploration. Figure 2.1 shows this process of selection of the long-term direction of exploration by ground control.

Another important assumption is that the rover will move on a 2-dimensional surface. Altitude variation within the map will not be considered. In section 6.2, a possible extension of this method that also includes the 3-D elevation is

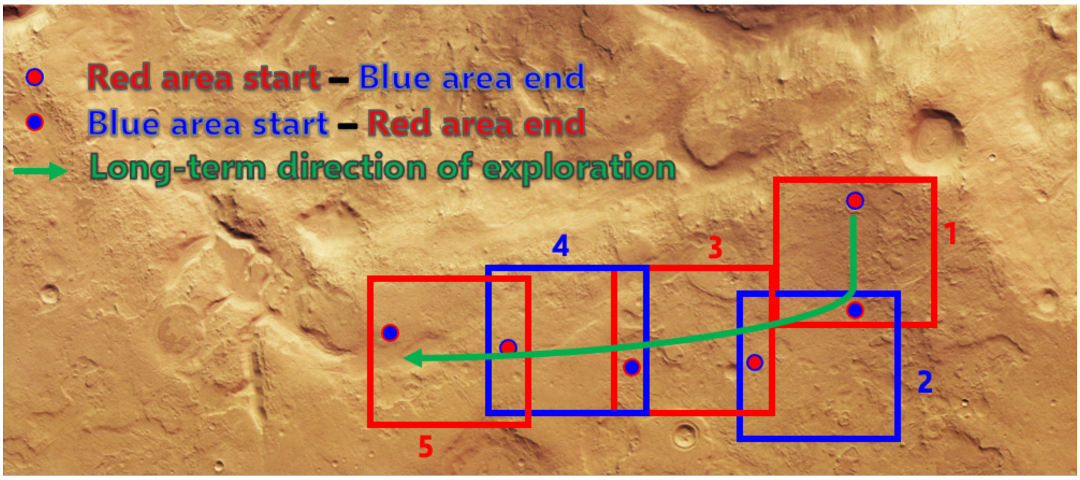

Red area start – Blue area end
Blue area start – Red area end
Long-term direction of exploration
1
2
3
4
5

Figure 2.1. Long-term direction of exploration

explained.

The 2-D map detected by the helicopter will be available to the rover for path planning from the beginning, including the scientific information, and will be then updated continuously with the new data. However, since small obstacles are difficult to be noticed from air, only the ones that are larger than a certain dimension will be marked on the map from the beginning. The others will be detected by the rover along its path.

General assumptions
Human intervention only possible at high level (such as direction of exploration)
Satellite images will be available to ground control
The rover moves within a 2-D map
The map with scientific information will be available to the rover before exploration
Rover is not aware of the obstacles within the map before exploration (except for large ones)

Table 2.1. General assumptions summary

Rover assumptions

In particular, the hazard detection system of the rover is set to operate in a discrete way to save energy and resources. In fact, the rover stops every 3 m along its route to detect any new obstacles.

Moreover, rover localisation and trajectory control are considered to be always satisfied, since they are not the main focus of this study.

Finally, the dynamics of the rover can be neglected given its very low velocity of motion.

Rover assumptions
The hazard detection system works in a discrete way
Localisation and trajectory control are always satisfied
The dynamics of the rover can be neglected, since its velocity is very low

Table 2.2. Rover assumptions summary

2.2 Rover concept of operations

In Figure 2.2, a flow chart that shows the operations that the rover shall carry out to achieve its exploration goal is reported.

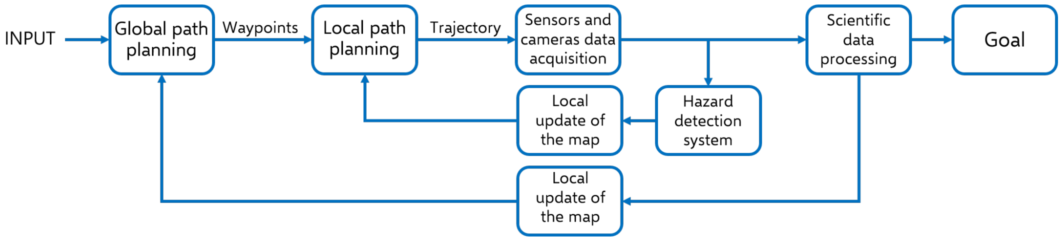


Figure 2.2. Rover concept of operations flow diagram

In this process, the input is represented by the scientific information and by the features of the area to investigate, that have been sent from the helicopter to the rover. The data processing of this initial information may be done either by the rover or by the helicopter depending on their computational and communication capabilities. Indeed, performing the initial data processing on the helicopter itself would prevent the need of sending a large amount of instrumental data from the aerial vehicle to the rover. However, it would then be necessary to have enough computational capabilities on board of the helicopter.

In the chart, the path planning is divided into two different components: global and local. The global path planning is responsible for generating the high-level route through the scientific points of interest, by means of the definition of the trajectory waypoints. During its operational phase, the rover, thanks to its sensors and cameras, will gather more scientific information that will be processed and locally updated on the map. Therefore, a feedback control is needed to replan the path globally any time the scientific data are revised on the map.

The local path planning method, instead, is in charge of the definition of the actual trajectory between the waypoints avoiding the obstacles along the path. The hazard detection system acts as a feedback on the local planning: whenever a new obstacle is detected, it is added to the map and the on-board computer of the rover will then be aware of it for its local path planning.

2.3 Definition of the scientific reward

Before the development of the path planning system, it is important to define how the scientific reward will be evaluated. The challenging part of this operation is to find a method that assigns a numerical value to something that is not really measurable as the scientific return of an object.
Although the work of this thesis is not focused on the feature-to-reward classification problem, it is important to briefly analyse how this process may work to understand which type of data will be available to the rover for path planning.

2.3.1 Feature-to-reward classification in the literature

In literature, different systems have been proposed in this field in order to increase rover autonomy in the analysis and processing of the surrounding environment. In [14], a Support Vector Machine (SVM) trained with a supervised learning has been suggested for performing the classification from the observations to the reward map. In particular, the authors of this study used trained SVM to analyse multispectral satellite images and to classify different types of minerals in the area, depending on their different thermal emission. The scientists will then assign a reward value to the different minerals based on their importance for the mission. For instance, as shown in Figure 2.3, goethite and calcite are selected as minerals of interest, as both are considered to be representative of areas that may have contained water. Magnetite is assigned the highest reward value as its presence was not initially known by the scientists and, as the other two minerals, it provides good indication of presence of water on the surface.

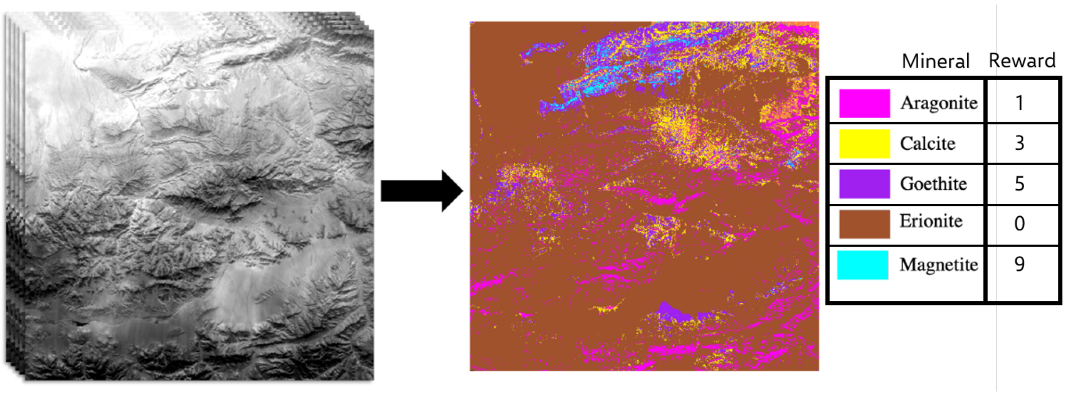


Figure 2.3. High resolution multispectral images obtained from ASTER instrument of Terra satellite, their mineralogical signatures and their respective reward value [14]

The functioning of the SVM and the role of the hyperplane in this method are well explained in [15], where Support Vector Machine is applied to a life science case study.

Another interesting class of methods is based on the emulation of the processes and assessments that a human field geologist would perform to analyse a potential science target. The basic attributes used in classification of geological features are *structure* (e.g. geometric shape, scale, orientation and form), *texture* (e.g. luster, relief, grain size, shape and sorting), and *composition* (e.g. colour, albedo, specularity and mineralogy).

This approach has been used with good results in Science Assessment and Response Agent (SARA) [16] and Autonomous Science Target Identification and Acquisition (ASTIA) [17] methods that with the application of fuzzy logic assign a Scientific Value Score (SVS) to potential science targets in an image, based on rock morphology.

In particular, for the development of SARA, a database of parameters used in terrestrial field geology was compiled for each attribute and each one was assigned a SVS, based on relative geological significance. The overall SVS of a scientific target is derived by combining all the values of the features observed. Therefore, total SVS of an element is function of different parameters:

$$\text{SVS} = f(A_S, A_T, A_C, A_X, Q, B) \tag{2.1}$$

where A_S is the overall structural attribute score, A_T is the overall textural attribute score, A_C is the overall compositional attribute score, A_X is the composite attribute score, Q is a quality factor and B is a bias factor.

A possible method to evaluate SVS could be:

$$\text{SVS} = (\sum A_S + \sum A_T + \sum A_C + A_X) \cdot Q \cdot B \tag{2.2}$$

Finally, there also is a group of methods that uses probabilistic approach.

In [18], Bayesian Networks (BN) calculate the probability that each detected feature is a particular object, based on the attribute measurements. So, in this case, there is not a direct classification of the object depending on its attributes. Indeed, BN define the probability that given those attributes a certain object is, for instance, a certain type of mineral.

A benefit score S_b is assigned to each feature that was identified by the BN using the Equation 2.3, which gives different gains to the possibility of discovering different types of minerals depending on their importance for the mission of the rover.

$$S_b = K \cdot P \cdot \Sigma \cdot [aPr(A) + bPr(B) + ... + nPr(N)] \tag{2.3}$$

In the equation, K is a scale constant, P is the apparent size (also known as angular diameter) of the feature, Σ is the normalised standard deviation of the probabilities, and $a, b, ..., n$ are the importance weights for minerals $A, B, ..., N$.

$Pr(A)$ stands for probability of A. The scale constant K simply scales the scores to be in an appropriate scale [18].

Finally, the authors in [19] introduce a science hypothesis map, as the probabilistic structure in which scientists initially describe their hypotheses, that is then improved as the robot makes new measurements. In this context, it has been considered that sampling the same region repeatedly leads to diminishing expected returns.

2.3.2 Application to the model for path planning

A common aspect of all the reported methods is that their final output is a numerical reward value associated to each element that is observed in the environment. This reward is given to each feature depending on the importance that it has in relation to the mission of the rover.

As already said, since this thesis is focused on the path planning part of the problem, the feature-to-reward process is assumed to be already implemented for hypothesis. Therefore, the input of the decision making method will be a set of scientific points of interest, each one associated with its Scientific Value Score. The SVS will be considered as a number from 0 to 1, where 1 corresponds to the maximum possible scientific return. These scientific points may represent rocks or particular elements or areas of interest for the mission of the rover.

Moreover, the idea that sampling the same region or the same type of features would lead to a decrease in the expected scientific return explained in [19], has been developed as well, but with a different approach. First of all, for each scientific point, in addition to the SVS domain, a category domain has been added to classify also the type of feature that is present in that scientific point, e.g. which kind of mineral. Then, as it will be better explained in section 3.2, instead of gradually reducing the reward of certain features, depending on the ones that have already been sampled, as done in the quoted study, a new objective function was introduced in the multi-objective MCTS: the number of categories visited by the rover. In this way, the algorithm will try to maximise the number of visited categories together with the other objective functions and, so, the scientific points that present categories that have not been visited yet will be favoured.

Since, in general, there is a connection between the type of feature (category) and the scientific reward, as for instance a certain mineral is more valuable than others, the two domains have been linked through Equation 2.4.

$$\text{SVS} = (\text{rand}(0,1) + (cat - 1)) \cdot \frac{1}{n_{cat}} \tag{2.4}$$

Where rand(0, 1) is a random number between 0 and 1, $\frac{1}{n_{cat}}$ is the maximum value of SVS (which is 1) divided by the total number of existing categories and *cat* is the number of category of that specific scientific point. Indeed, the categories are defined as numbers, for instance from 1 to 10, if there are 10 different types of scientific points. As a consequence, the reward for each category will be in a certain range and category 1 will be the one with the lowest scientific return.

An example of a reward map that contains 20 scientific points of interest (blue circles), belonging to 10 different categories (although not all of them are present within the map), and 50 obstacles (red circles) is depicted in Figure 2.4. Each scientific point in the graph is reported together with its SVS and its category (SVS - category: e.g. 0.35 - 4).

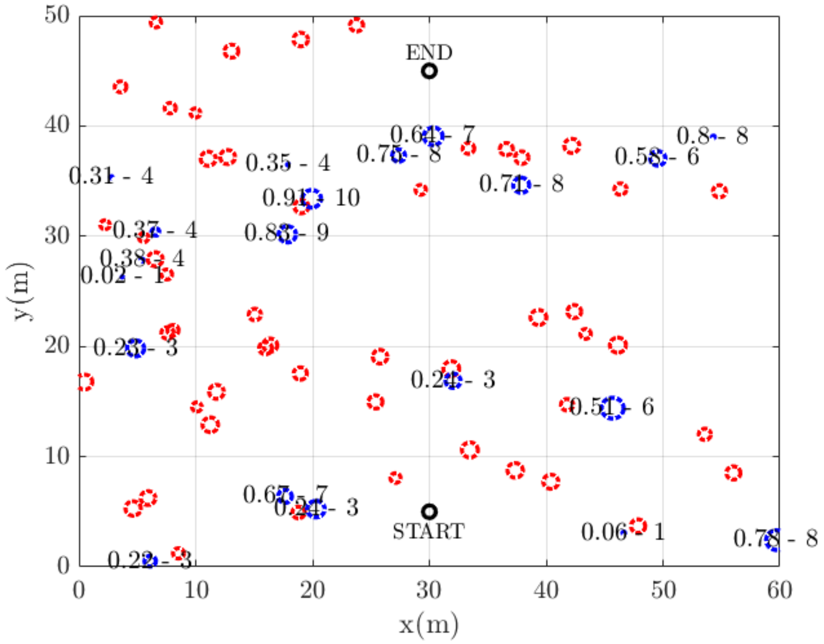


Figure 2.4. Example of the reward map with 20 scientific points and 50 obstacles

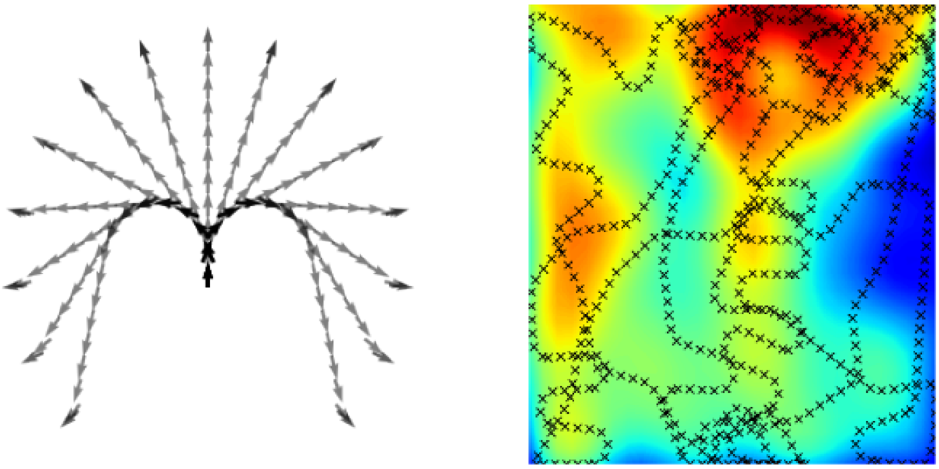
2.4 Overview of the selected method

Now that the conceptual operations have been described and the scientific reward has been established, it is possible to explain which methods have been used to fulfil the different parts of path planning problem. As previously stated in section 1.3, the multi-objective Pareto Monte Carlo Tree Search has been selected for global path planning.

Concerning the local planning, a navigation system based on artificial potential fields has been developed.

2.4.1 Global planning with multi-objective Pareto MCTS

Global path planning represents the decision making part and so the most important and innovative segment of the system.
In [12], Pareto MCTS is used to choose the most promising move between the primitive paths of a robot, as reported in Figure 2.5a. In the cited study, the vehicle moves through the environment shown in Figure 2.5b, in which the hotspots represent high interest areas. In this case, the two competing objectives considered in the Pareto MCTS path planning method are the maximisation of information gain and of the exploration of the area. It is possible to notice how the hotspots are visited and sampled by the robot more frequently than the cold ones, showing a certain consistency of the method. Moreover, the fact that the decision making tool chooses the next move amongst the possible primitive paths of the robot means that the output of the system will not be a set of waypoints that the robot should follow, but its actual trajectory through the environment. In other words, global and local path planning are fused together.



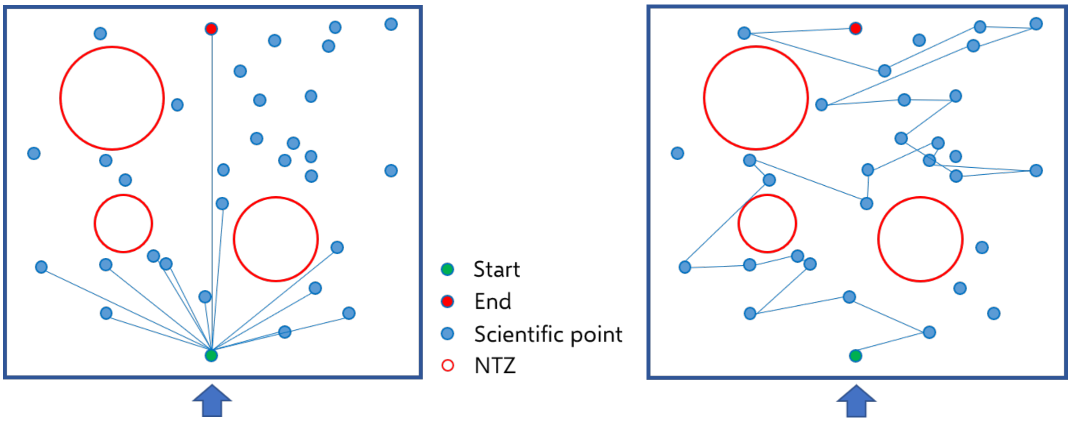
(a)

(b)

Figure 2.5. (a) Primitive paths for a wheeled rover. (b) Path and collected samples determined with Pareto MCTS [12]

In this thesis work, given the discretisation that has been made on the scientific reward, previously illustrated in section 2.3, the approach has been changed. The most promising next move will not be selected among some primitive paths the rover may follow, but between a certain number of the scientific points that are closer to the current position of the rover, as it is depicted in Figure 2.6a. Also, the ending point will be added every time as a possible next move. Therefore, the waypoints will be formed as a sequence of the most promising scientific points found at each step by the algorithm. An example of this idea is represented in Figure 2.6b. In the map of these two figures small obstacles are not depicted for clarity. Only three large Non-Traversable Zones (NTZ), that represent areas in

which the rover is not able to move through, are shown.
As already said, the trajectory between these waypoints will then be found thanks to the local path planning system.



(a)

(b)

Figure 2.6. (a) Overview of the possible next moves of the rover. (b) Example of global path planning waypoints output

2.4.2 Local planning with artificial potential fields

The artificial potential fields approach has been chosen for the computation of the local trajectory of the rover among the obstacles, since this system is very fast and does not require great computational effort. These are good pros for a system that will be placed aboard a planetary rover. On the other hand, an important problem related to this method is the possibility to get stuck in local minima, but there are ways to avoid or reduce the occurrence of these type of events. This issue will be addressed more in detail in chapter 4.
The idea of the artificial potential fields method is based on the analogy with natural conservative force fields, that can be derived as the gradient of a potential function that only depends on the position. In the same way, the motion of the rover through the map is driven by a force, that is evaluated in each point as the gradient of the sum of different potential fields, that represent the different features of the environment. In particular, the target point will be modelled as an attractive field, while the obstacles will be shaped as repulsive fields. An example of a trajectory generated with the artificial potential fields approach is depicted in Figure 2.7. In this figure the contour lines of the attractive potential field are represented as well.

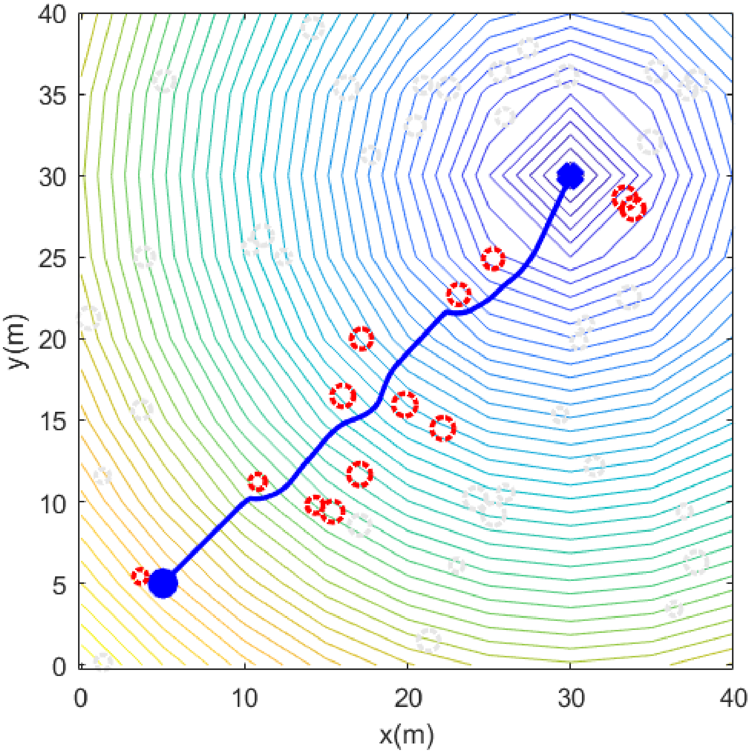


Figure 2.7. Example of a trajectory obtained with an artificial potential fields method

Chapter 3

Multi-objective Pareto MCTS

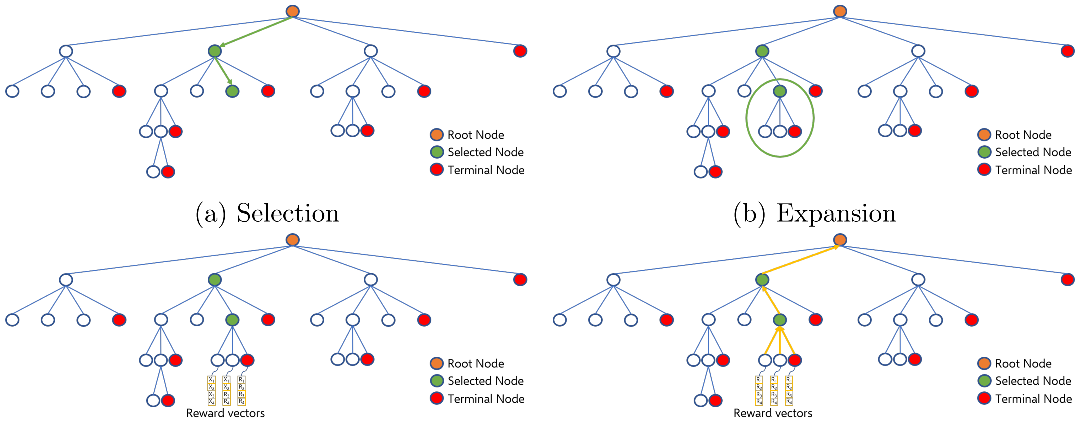
3.1 MCTS general overview

The classical version of Monte Carlo Tree Search (MCTS) is a heuristic search algorithm used for decision making and often applied to software playing of traditional board games like *Go* or *chess*. The main purpose of this algorithm is to choose, given a state of the game, the most promising next move.

With this method a tree search is built, node-by-node, according to the outcomes of the simulations that allow to determine its most promising branches [20]. Each node of the tree represents a different state of the system, that in the context of this thesis could be either a certain scientific target or the start/end point (cf. Figure 2.6a). The root of the tree is the node that represents the state in which the search starts.

The process of MCTS can be broken down into the four steps that are depicted in Figure 3.1: *selection, expansion, simulation* and *back-propagation.*

Selection is the core of the search method, since it determines the traversals



(a) Selection

(b) Expansion

(c) Simulation

(d) Back-propagation

Figure 3.1. MCTS steps of the process

down the tree and, so, how the tree is explored. A single traversal is a path from the root to a not expanded node, called leaf node. The selection policy (cf. section 3.3.1) defines, at each step of the traversal, which one is the best child node to be explored. In Figure 3.1a, green nodes are the selected nodes for the traversal that is ongoing, which is represented by the green arrows.
Once a leaf node is found, if it is not a terminal node (an ending point, the red nodes in the figure), it undergoes the process of *expansion* and its children are added to the tree (Figure 3.1b). These new nodes are then chosen one by one to become the root for a single *simulation/rollout* (Figure 3.1c), which consists in a propagation up to a terminal node based on a simulation policy (cf. section 3.3.3). The results of these rollouts are then back-propagated up to the root node. In other words, during the *back-propagation* process, the algorithm updates the total simulation reward vector $\mathbf{X}_k$ and the total number of visits n_k of each node k of the traversal path (Figure 3.1d) [21]. These variables are attributes of the nodes of the tree and they are defined as follows.

- $\mathbf{X}_k$ - *Total simulation reward vector* of a node k is the sum of the simulation results that passed through the considered node (in a single-objective MCTS, the reward vector is scalar).

- n_k - *Total number of visits* of a node k is the counter of how many times the node has been on the back-propagation path and so it counts how many times $\mathbf{X}_k$ has been updated.

The process is iterated until a predefined time/computational/memory budget is exhausted. Afterwards, the most promising child of the root node is chosen depending on the attribute n_k: the most visited child of the root node will be selected as the next move, since the number of visits indicates how promising is the node. Indeed, the most visited one has resulted to be picked as best child for more times than the others by the selection policy.

3.2 Objective functions

The main principle of operation of the multi-objective MCTS is identical to the one that has just been explained in the context of the single-objective. However, in the multi-objective case, the reward $\mathbf{X}_k$ will be a vectorial variable based on the objective functions considered and, during the back-propagation step, each component of the vector will be updated separately with its respective new simulation results (cf. section 3.3.4).
Therefore, it is now necessary to define the four objective functions that will be evaluated in the path planning process, even though three of them have already been presented above in the text. The first variable to be maximised is the total

scientific return of the mission, which is the actual core of the path planning process and is determined as the sum of the Scientific Value Score of each scientific point that is sampled by the rover along its path. A second important function is the path length, which shall be minimised. Since Pareto MCTS method aims to the maximisation of the objective functions, minimisation will be managed as a maximisation of the negative of the variable to be minimised (e.g. minimisation of the path length l_{tot} will be treated as a maximisation of the variable: $-l_{tot}$).

Given these first two objective functions, the algorithm will try to gain as much scientific reward as possible reducing, at the same time, the length of the path that the rover should travel to sample these targets.

The third function that needs to be maximised is the number of different categories of the scientific features that are sampled by the rover. As explained in section 2.3.2, this objective aims to the exploitation of all the different pieces of information that are present in the environment.

Finally, the last variable has been introduced to force the rover to stay away from the borders of the area explored by the helicopter/drone and, so, to prevent it from going outside the (partially) known domain. This function has been defined as the sum of the distances d_i between the visited scientific targets and the point C at the centre of the exploration area, as depicted in Figure 3.2 (Equation 3.1).

$$f_4 = \sum_i d_i \tag{3.1}$$

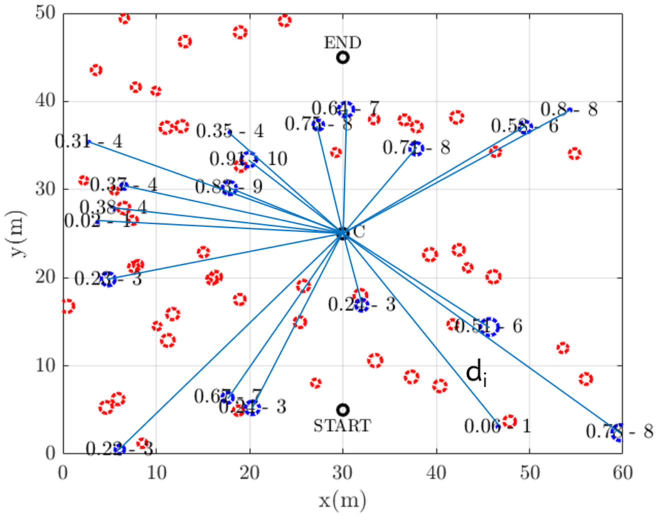


Figure 3.2. Map showing the distances d_i between the scientific targets and the central point of the known area

Indeed, introducing the minimisation of this function f_4 into the objectives of the MCTS will penalise scientific points that are very close to the borders (such as point 0.78 - 8 at the bottom-right of Figure 3.2) as they have greater values of d_i. Therefore, the planning method may prefer to visit other targets rather than the ones at the borders.

In Table 3.1, a summary of the four objective functions is displayed.

Function	**Description**	**Max/min**
f_1	Total SVS	Maximise
f_2	Total path length	Minimise
f_3	Number of visited categories	Maximise
f_4	Sum of distances from centre	Minimise

Table 3.1. Summary of the objective functions

The different functions are scaled, even though it is not actually necessary, in order to be more or less of the same order of magnitude. For instance total path length is divided by 100.

Before proceeding, it must be highlighted that the MCTS is not a global optimiser. Indeed, it works step by step, determining at each step the most promising next move with a statistical approach, nevertheless the global trajectory will be, in general, sub-optimal. Moreover, in the multi-objective scenario, the method has to reach a compromise between the different objective functions and, so, the overall path will be sub-optimal also for this reason.

3.3 Multi-objective Pareto MCTS steps

The Monte Carlo Tree Search starts from the root node. In the first cycle, the selection step ends immediately, as the root node is already a leaf node and, so, its children are added to the tree through the expansion process. Afterwards, simulation and back-propagation are performed from each one of them in sequence. The overall process of selection, expansion, simulation and back-propagation is then iterated until the budget is over.

In the following sections, these steps will be described more in detail starting from the driving force of the whole method: selection.

3.3.1 Selection

During the selection process, the algorithm has to determine at each step, by means of a selection policy, the most promising node, in order to perform a traversal along the tree until an unexpanded node is found (Figure 3.3).

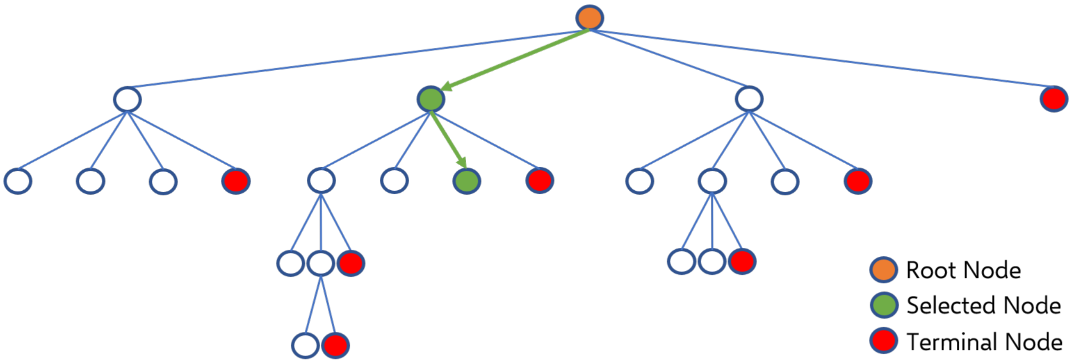


Figure 3.3. Selection step

Pareto front policy for multi-objective evaluation

In case of scalar reward, the most promising node is simply the one with highest expected reward. However, in the multi-objective scenario, the reward is a vector based on the objective functions. Therefore, the optima are not unique and there is a set of optimal choices that are considered equally best, as some of them may be good for part of the objectives but worst for others. This set of equally optimal nodes is defined Pareto optimal node set.

Before establishing a definition for this set of nodes, it is necessary to specify the concept of dominance. Referring to the definition reported in [12], let $\mathbf{X}_k$ be a D-dimensional reward vector of a choice k and $X_{k,d}$ be the d-th element of the vector. A choice is one of the possible child nodes to select between. The i-th choice is better than, or *dominating*, another choice j, denoted as $i \succ j$ or $j \prec i$, if and only if the following conditions are satisfied:

1. any element of $\mathbf{X}_i$ is not smaller than the corresponding element in $\mathbf{X}_j$: $\forall\, d = 1, 2..., D,\ X_{i,d} \geqslant X_{j,d}$;
2. at least one element of $\mathbf{X}_i$ is greater than the corresponding one in $\mathbf{X}_j$: $\exists\, d \in \{1, 2, ..., D\}$ such that $X_{i,d} > X_{j,d}$.

As a consequence, it is said that choice j is *non-dominated* by choice i if and only if there exists at least one component d of the reward vectors such that $X_{j,d} > X_{i,d}$ ($j \not\prec i$ or $i \not\succ j$). Moreover, two choices i and j are defined *incomparable* ($i \| j$) if and only if there exists one dimension d_1 such that $X_{i,d_1} > X_{j,d_1}$ and a dimension d_2 such that $X_{i,d_2} < X_{j,d_2}$.

By means of these concepts, a proper mathematical definition of the Pareto optimal node set can be expressed as follows [12]. Given a set of nodes V, a subset $P^* \subset V$ is the Pareto optimal node set, in terms of expected reward, if and only if:

$$\begin{cases} \forall\, v_i^* \in P^* \text{ and } \forall\, v_j \in V,\ v_i^* \not\prec v_j \\ \forall\, v_i^*, v_j^* \in P^*,\ v_i^* \| v_j^* \end{cases} \tag{3.2}$$

The first condition states that the nodes belonging to the Pareto front are not dominated by any other, while the second one means that these nodes are incomparable with each other. A Pareto optimal node set in a two dimensional reward case is shown in Figure 3.4a. In this example, the two objective functions are the SVS and the path length of the rover. It can be seen that the two conditions needed for the Pareto set are satisfied for the green nodes. In the 2-dimensional case, the Pareto front is approximately placed on a hyperbola, as in the inversely proportional relation.

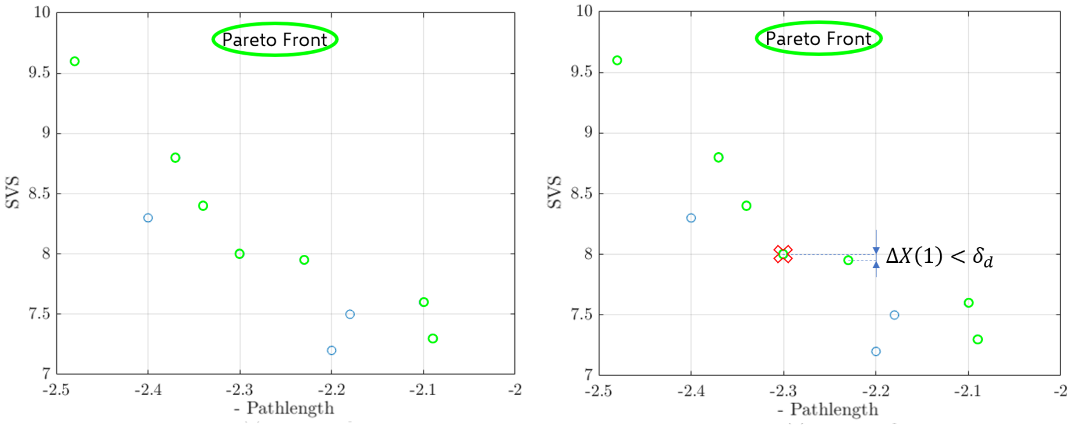
Once this set of optimal nodes has been defined and computed at the selection step during each cycle, the most promising child node is chosen uniformly at random among the nodes of the Pareto set. Indeed, there is not a specific criteria to choose between them, as they are all considered equally optimal.

However, if there are objectives that are slightly more important than others, the Pareto front could be built using non-absolute dominance. Meaning that, for the objective functions of lower importance and, so, for their respective dimension d of the reward vector, $X_{i,d} > X_{j,d}$ will not be considered enough to verify the second condition of dominance, that has been previously described. A stronger condition will be necessary: $|X_{i,d} - X_{j,d}| > \delta_d$, where δ_d is a positive value, that has to be defined for each objective function of lower importance. In other words, the node i, in order to be considered better than the node j in relation to the dimension d of the reward vector, shall have not only a value of $X_{i,d}$ greater than $X_{j,d}$, but also a difference between these variables larger than the defined δ_d.

For instance, in a 2-objectives scenario, if having a shorter path is considered a bit more important than the total Scientific Value Score collected, for this last objective a non-absolute dominance condition may be introduced. In this way, as shown in Figure 3.4b, nodes in which the reward value related to the SVS of one node is greater than the other's but their difference is smaller than δ_d are considered equal from the SVS point of view. Therefore, the one with higher reward value connected to path length will be added to the Pareto optimal node set, while the other will be discarded.

It is clear how, if δ_d is set to a higher value, the algorithm will evaluate more nodes as comparable, in relation to the reward of the scientific return, and will include in the Pareto set only the one of these with the shortest path length. Therefore, longer paths will be privileged only if they allow the rover to achieve a consistent increase in the scientific return.

In section 5.3.2, some simulations carried out with the introduction of the concept of non-absolute dominance have been reported. Different values of δ_d have been assigned to the four components of the reward vector, penalising one or the others in the various tests.



(a) (b)

Figure 3.4. Pareto front in case of (a) absolute and (b) non-absolute dominance

Upper Confidence Bound (UCB)

The computation of the Pareto optimal set is based on the Upper Confidence Bound (UCB) $\mathbf{U}_k$, which is a vector calculated for each child node k from the total simulation reward vector $\mathbf{X}_k$ according to the Equation 3.3. In the context of this study, the UCB is a vector of four components, since four objective functions have been considered.

$$\mathbf{U}_k = \frac{\mathbf{X}_k}{n_k} + \mathbf{C}\sqrt{\frac{4 \ln n_{tot} + \ln D}{2\, n_k}} \tag{3.3}$$

The first term of the equation represents the average reward that is obtained from child k, as it is the division between the total simulation reward and the total number of visits of the node. The second term is added to let the algorithm search along the whole tree and not get stuck in the nodes that have been found most promising in the beginning. Indeed, the term increases the UCB value of the nodes that have been visited a lower amount of times, since the denominator is $2\, n_k$. In the equation, n_{tot} is the total number of visits of the parent node, so the total number of visits that have all the children together. D is the number of components of the vector $\mathbf{U}_k$, which is four in this case, and $\mathbf{C}$ is a vector of scaling constants. In Figure 3.5, is reported an example of the trend of this second term of the equation in relation to the number of visits of the node. It can be noticed how as the visits decrease the term increases. The number of visits is never 0, because, every time a new node is added, a simulation starts from it and, so, in the back-propagation step of the process the variable n_k is updated to 1.

Vector $\mathbf{C}$ is used to scale the second term according to the different order of magnitude of the several objective functions, although, as explained in section 3.2,

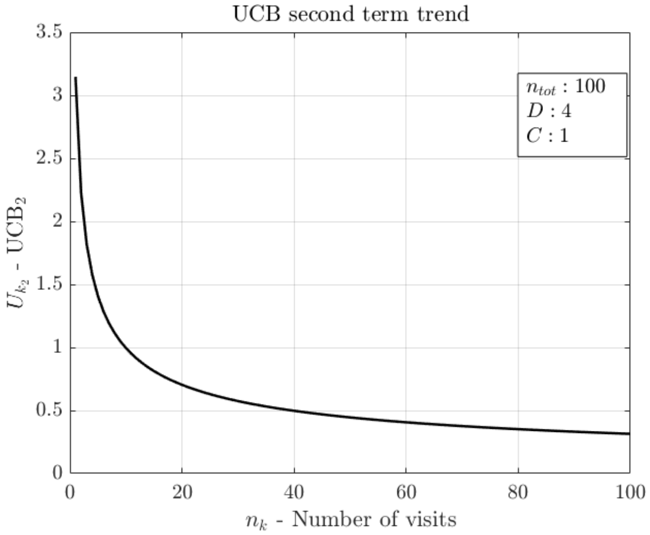


Figure 3.5. Trend of the second term of the UCB

the variables themselves have already been scaled.

In conclusion, the generation of the Pareto optimal node set is based not directly on the simulation reward vector $\mathbf{X}_k$, but on the four components of the vector $\mathbf{U}_k$, that are compared to the other children's $(i, k...)$ vectors $(\mathbf{U}_i, \mathbf{U}_j...)$ according to the two conditions previously defined.

3.3.2 Expansion

Once a leaf node has been found through the search process driven by the selection policy, its children are determined and linked to it with the expansion step, as depicted in Figure 3.6. However, if the selected leaf node is a terminal node (ending node), expansion is not possible. Therefore, simulation (that will actually stop immediately, as explained in section 3.3.3) and back-propagation steps will be directly performed from it.

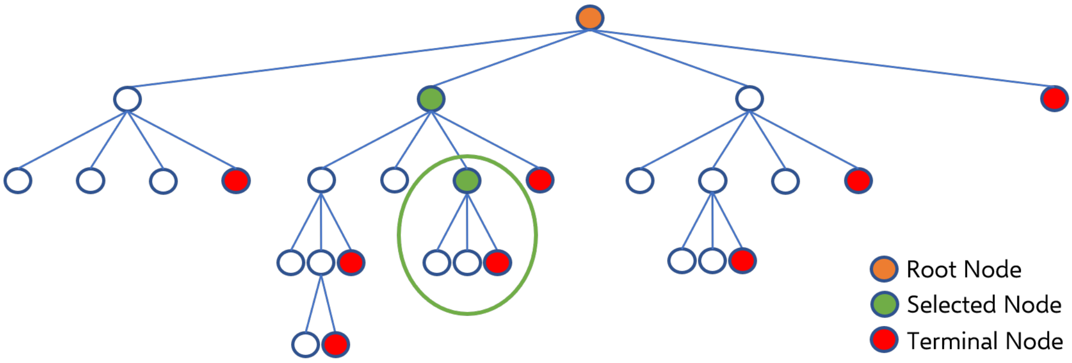


Figure 3.6. Expansion step

The children of each node represent all the possible states that can follow the state of the parent. As explained in section 2.4.1 and shown in Figure 2.6a, the possible next states are defined as a certain percentage of the closest scientific points, plus the ending point. In particular, the percentage has been set to the 50% of the elements of the vector of the possible states considered at the beginning of the search minus one (that is the current state). Therefore, if the state vector has 21 elements, the algorithm will select the closest 10, plus the ending point. So, eleven children will be added to the tree. This number will remain constant during the construction of the tree, as it is based on the length of the vector of possible states available at the beginning of the search. It may happen that, in the most expanded branches, the states that are yet to be visited are less than eleven. In this case all of them will be added as children.

The ending point is always linked to any node of the tree, to allow the algorithm to always analyse the reward vector obtained from the possibility of moving the rover to the ending point after the state of the parent node. In this way, the rover can decide to conclude its route after having visited any of the scientific targets of the area.

It must be highlighted that during the time in which the tree is built the rover is not moving, it is idle at the state position of the root node. The nodes are explored by the Monte Carlo Tree Search algorithm and not by the rover. Indeed, the method will determine the most promising node to be next visited by the autonomous vehicle. Section 3.4 will better explain of how the succession of MCTS computation and rover moving will take place.

To evaluate the distances between the different points in the area and, so, to determine the targets that are closer to the root node, the artificial potential fields method has been used. Of course, this analysis is performed with the data that the rover knows on the environment at that point of the exploration (not all obstacles are probably known yet). The distances from each state point of the map to every other target are saved in a matrix (named path length matrix) before the starting of MCTS, so that this information is available to the decision making system.

With respect to other MCTS algorithms, it has been decided to directly expand all the children of the parent node at once, instead of adding only one child. In this way, the knowledge on how promising is the node is immediately more effective than having the information of just one child.

Once the children are added, a simulation process is carried out from each new node and the data are then back-propagated up to the root node.

3.3.3 Simulation

The role of the simulation step, also called rollout, is to propagate, according to a policy, the rover moves from the point in which the simulation starts up to the ending node and evaluate the reward vector of that path (Figure 3.7). This process is important to estimate the potential reward of that node and so to understand how promising is.

If the newly added child node is a terminal node, the simulation will stop immediately, since the ending point has already been reached. Therefore, the reward vector will be only based on the data of the path up to this node.

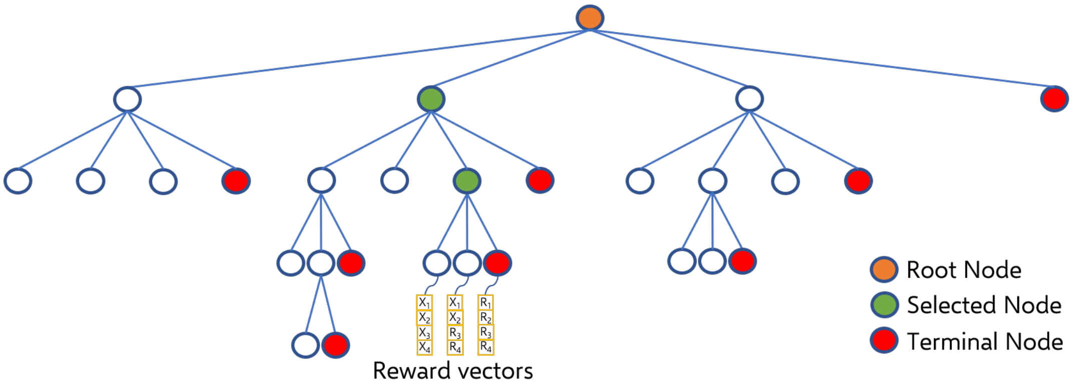


Figure 3.7. Simulation step

The simulation policy determines how the next nodes are chosen during each rollout. In particular, the next state will be selected each time at random among the two closest not visited points in the map. In this way, a quite regular path of the rover within the area will be simulated.

The output of each rollout is a reward vector $\mathbf{R}$ of four components built on the four objective functions.

3.3.4 Back-propagation

Back-propagation is the last step of the cycle and its function is to update the total simulation reward vector $\mathbf{X}_k$ and the total number of visits n_k of each node along the back-propagation path, which is the inverse of the search path up to the root node, as shown in Figure 3.8 for the three children nodes that have been linked to their parent.

In each process of back-propagation, the two following calculations are performed for each node k: the reward vector is added to the total simulation reward (Equation 3.4) and the number of visits is increased by one (Equation 3.5).

$$\mathbf{X}_k = \mathbf{X}_k + \mathbf{R} \tag{3.4}$$

$$n_k = n_k + 1 \tag{3.5}$$

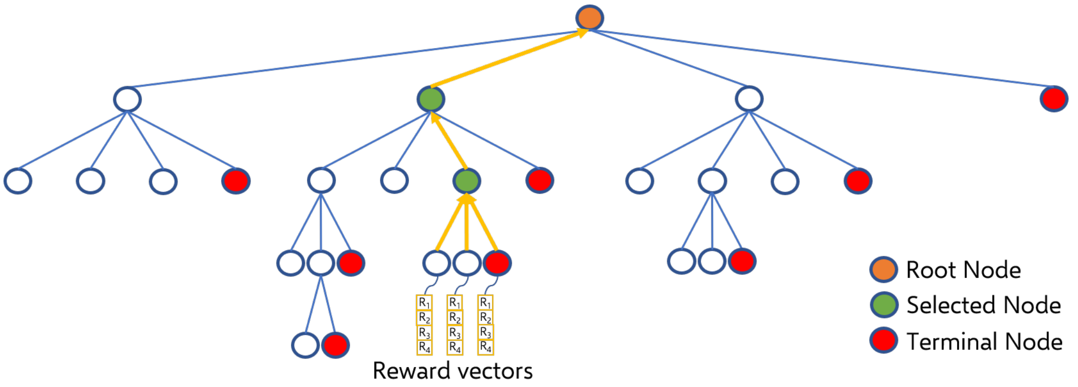


Figure 3.8. Back-propagation step

3.4 Exploration strategy

3.4.1 Planning-moving sequence

Once the budget is exhausted, the Monte Carlo search stops and the child of the root node with the greatest n_k is chosen as next move.

It is now necessary to establish how the MCTS is integrated with the exploration of the rover. When does the rover stop for computation and how many waypoints are defined before the rover moves to the next target?

There are two main possibilities. The first option is to determine, at the beginning, all the waypoints from the start up to the end, by performing the search multiple times using the next node as root node of the following search. Another possibility is that the rover could determine its route point by point and, so, move to the next point as soon as it is determined.

Both strategies have pros and cons. In the first case, the main issue is related to the poor knowledge the rover has on the environment at the starting point. Indeed, during its route it will discover new elements such as obstacles or scientific points or gain new information about already known targets, that would definitely force it to change its plan, discarding the old one that becomes useless. On the other hand, the second scenario totally lacks of long-term planning.

With regard to the computer memory, both ideas are feasible, as only waypoints, path length matrix and part of the tree will be stored. Indeed, the actual trajectory will be, in both cases, computed step by step, as the rover moves through the area, with the artificial potential field method, as will be explained in section 4.1.

After these considerations, a compromise solution has been adopted for which, as soon as the two following waypoints are computed, the rover moves to the first one and does its sampling. Then another waypoint is determined and added as second next and the rover travels to the new first. The process is iterated until the ending point is reached by the vehicle. Figure 3.9 illustrates an example of this sequence of computation and travelling.

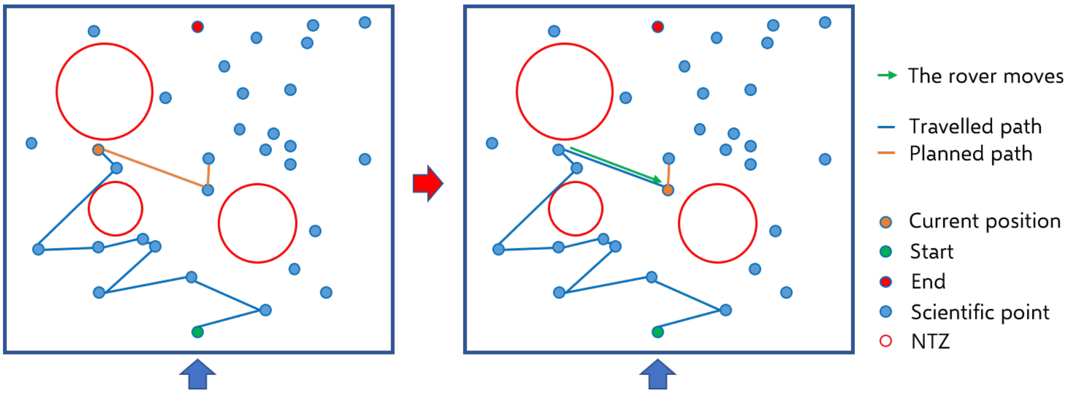


Figure 3.9. Sequence of MCTS computation and rover moving

3.4.2 Memory management

As said, the memory of the on-board computer will not be particularly challenged by this method, since only two waypoints, a path length matrix and part of the tree will be kept in memory.
In particular, each new Monte Carlo search will be built over the tree that has already been produced in previous computations, while the old parts will be discarded to empty the memory, as shown in Figure 3.10. In this way, the tree is just enlarged and updated and not started from scratch every time. As soon as the rover approaches the first next waypoint, the part of the tree upstream of that node is erased from the memory.

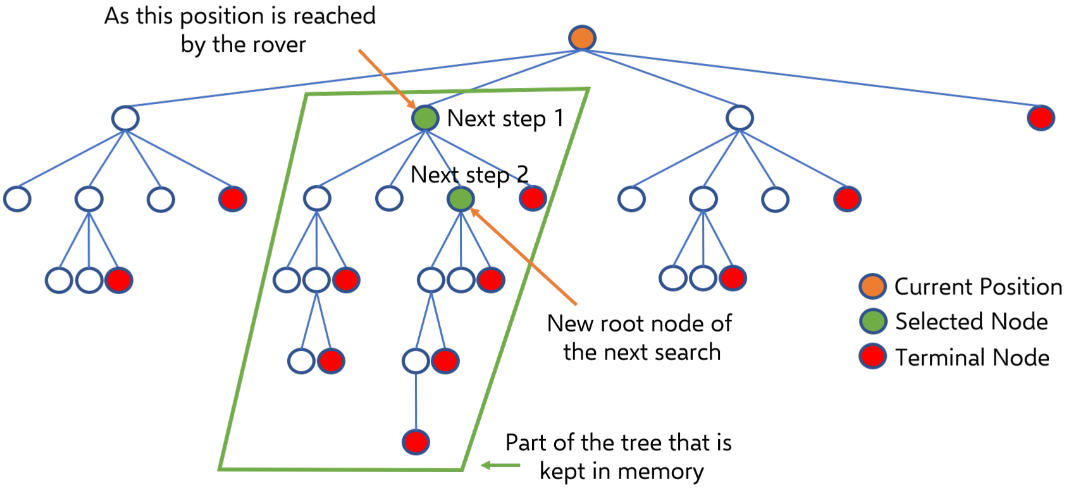

As this position is reached by the rover
Next step 1
Next step 2
New root node of the next search
Part of the tree that is kept in memory
Current Position
Selected Node
Terminal Node

Figure 3.10. Memory management

However, as will be explained in the following paragraphs, there are cases in which the tree has to be completely deleted and regenerated. Indeed, as discussed regarding the rover concept of operations (cf. section 2.2), a change in the scientific information will act as feedback on the global path planning system, whereas local re-planning will be needed when new obstacles are detected.

Change in the scientific information

As the rover performs the trips from one point to another, it stops every 3 m to observe its surroundings. During these analyses the knowledge about scientific elements in the area might be locally updated. In particular, these changes may concern different aspects:

- the scientific object the rover is about to reach is located in a too dangerous position as it is very close to an obstacle, that had not been previously detected;
- a generic scientific object along the path is located in a too dangerous position as it is very close to an obstacle, that had not been previously detected;
- a new interesting scientific target is observed;
- a scientific point belongs to a category or has a reward that is different than what expected from the helicopter data.

In these situations, the tree needs to be re-built, because the scientific information in the scenario has changed. Therefore, the decision making algorithm will be run again from scratch.

In the first case, the rover will stop at the point in which it has detected the new obstacle, since the next scientific point is not reachable, and will perform a new decision making process. As it is shown in Figure 3.11a and b, as the rover approaches the scientific target 0.95-10, a new obstacle is detected close to that point, that is then evaluated as too dangerous. Therefore, the rover stops there, a new MCTS is run and when two new next waypoints are determined, the vehicle moves to the first one.

In every other situation, the vehicle can complete its route towards the element that is going to. So, it does not stop immediately, but it reaches that scientific point and samples it. Only after that, a completely new search tree is built.

In Figure 3.11c and d, an example of the second situation is depicted. During the traversal from point 0.98-10 to 0.44-5, the rover finds out that the target 0.49-5 is too dangerous and so it deletes it from the targets to be visited. As a consequence, after the sampling in 0.44-55 a new tree is built. In green it is possible to see the already sampled points and the first next planned waypoint.

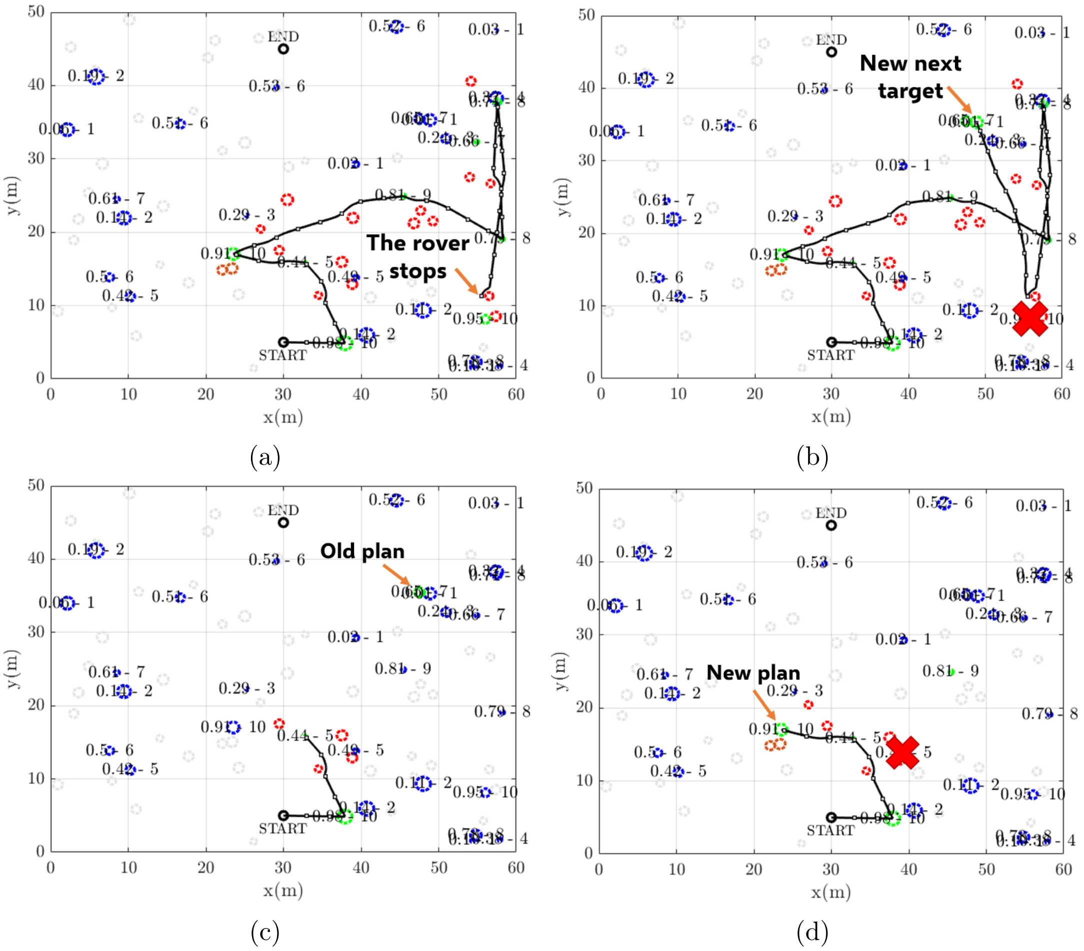


Figure 3.11. (a)-(b) Rover path before and after the global re-planning due to the discovery of an obstacle that is too close to the target where the rover is going. (c)-(d) Rover path before and after the global re-planning due to the discovery of a too dangerous scientific point

It can be noticed that the point the rover eventually samples in Figure 3.11d (that is 0.91-10) is different than the one that was planned in c, as a proof of the fact that the old tree and plan have been totally discarded, because of the change in the scientific information.

In case the rover discovers that the ending point is too close to an obstacle, the planning algorithm is run in the same way and the vehicle eventually stops its exploration at the last scientific point before the end.

Detection of a new obstacle

The situation is different in case newly detected obstacles do not interfere with the scientific targets of the map. In this case, global re-planning is not needed, because the scientific knowledge has not changed. Local re-planning will be carried out step by step, thanks to the artificial potential fields method (cf. section 4.1).

Moreover, after rover traversals in which new obstacles have been found, the path length matrix is updated with the new distances of the paths among the obstacles, while the search tree is kept in memory.

Chapter 4

Artificial potential fields method

4.1 Step by step algorithm

There are two different categories of local path planning systems, the first one in which the rover computes the whole trajectory between two waypoints at the beginning of the traversal and the second one in which there is not a predefined trajectory, as the rover determines its path step by step while it is moving.

The first type of algorithms is used in the situations in which the map is well known. Therefore, the rover can plan the path among the obstacles from the beginning and only a hazard detection and avoidance system will be used to modify it, if new obstacles are detected.

The second class of methods is able to navigate the rover in a less known environment. The trajectory is computed considering at each step the current position of the rover, the target location and all the obstacles in the surroundings.

In this thesis, as previously stated, the artificial potential fields method has been chosen for local path planning. This algorithm belongs to the second class. Indeed, in the defined scenario the rover has enough scientific knowledge to plan its global path, but only largest obstacles are determined by the helicopter. Therefore, the local navigation system shall allow the vehicle to move around obstacles that are detected step by step.

More in detail, the rover discovers the environment thanks to its cameras and instruments. In order to save energy and also because of the need of computational time, these detectors are not continuously active. Indeed, the rover stops every 3 m to take images and measurements. The data are then processed and, if the target is still reachable, the vehicle will start moving again.

In particular, as established in [22], two different zones are defined, a *known zone* and a *shadow zone*. The *known zone* is the section of the map that can be observed by the rover every time it stops during the traversal. Its geometry is dependent on the properties of the artificial vision system and of the instruments on board. In the context of this thesis, it has been considered as a circular area around the rover with a radius of 5 m. On the other hand, the *shadow zone* is the part of the map the rover does not know, because it is outside of this 5 m radius circle and so can not be seen by the instruments.

As the rover moves towards the *shadow zone* and, so, as it reaches the borders of the *known zone* it needs to stop and make new observations of the surroundings. That is why in the algorithm, the rover has been set to stop every 3 m to analyse the environment. Indeed, after this distance the rover gets close to the borders of the *known zone* (2 m far from it) and moving becomes dangerous. Figure 4.1 shows an example of this sequence of observation and motion during the traversal. The small grey squares with black edges along the path represent the points in which the rover stops to make new measurements; from each one of these locations a 5 m radius circular area is observed by on board instrumentation (sky blue circles). It can be noticed how the point in which the rover stops is pretty close to the borders of the previous *known area*. Moreover, in this example the rover finds out that the target where it is going to, which is 0.91-10, is located too close to an obstacle and, so, it is in a too dangerous position. Therefore, the vehicle does not continue its traversal towards that point and starts a completely new global planning process to determine new next waypoints.

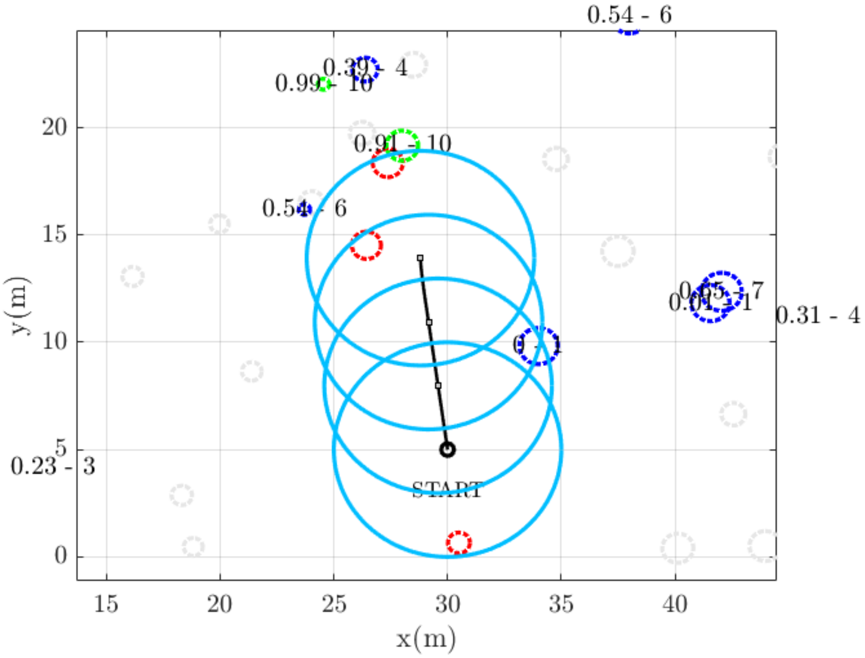


Figure 4.1. Sequence of observation and motion during the traversal of the rover towards the next waypoint. Sky blue circles represent the *known zones*

In real systems, the width or the size of the vehicle shall be considered, but, as stated in [23], in many algorithms, the width of the rover is added to the dimensions of the obstacle and the actual size of the vehicle is then shrunk to zero.

In this study, a generic small rover of 60 cm width was considered. In the code, however, this method of shrinking the dimensions of the vehicle to zero has been used. Therefore, the radius of each obstacle has been increased of 30 cm, which is half-width of the rover, plus other 10 cm as a safety factor. In each plot that will be presented, every obstacle will be shown with these enlarged dimensions and the rover will be considered as a point.
Of course, an obstacle will be detected within the *known zone* only if it falls within this area with its actual dimensions and not with the enlarged ones.
Since the navigation algorithm is realised considering a 2-dimensional map, the information on the height of the obstacles is lost. As a consequence, during the data processing, rocks and elements has to be defined as hazards or obstacles only if their height is enough to interfere with the rover motion. As depicted in Figure 4.2, the danger of a rock is related to how high it is, compared to the clearance of the rover.

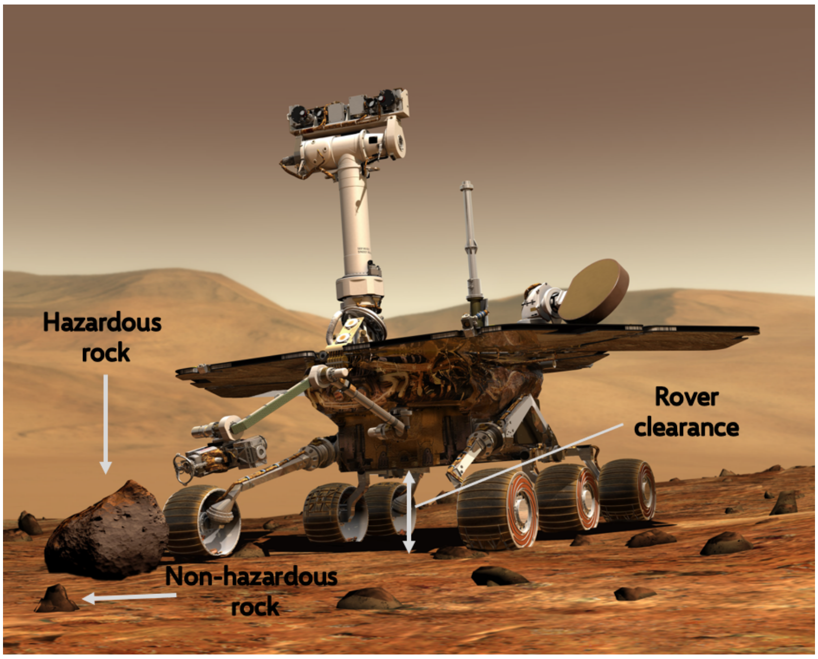


Figure 4.2. Example of hazards for NASA's Mars Exploration Rover Opportunity [24]

4.2 Potential fields definition

The local navigation algorithm defines a potential field U in each position of the map. The obstacles detected by the rover are considered as a source for the potential, while the final point, which is the next waypoint, is defined as a sink. In the next paragraphs, the fields associated to sources and sink will be described in detail. These single fields represent how the rover feels the different objects in the environment: it is repulsed by the obstacles and attracted by the target.

The total potential field is, then, defined as the sum of all the single fields of the elements and is used in Equation 4.1 to compute the 2-dimensional force vector $\mathbf{F}$ in each point of the trajectory, that will be then inserted in the dynamic equations (Equation 4.3) to determine the next step of the path towards the target.

$$\mathbf{F} = -\nabla U \tag{4.1}$$

The state vector of the dynamic system is reported in Equation 4.2. The first two components are simply the coordinates x and y of the rover in the map and the other two represent their time derivatives.

$$\mathbf{X} = \begin{Bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{Bmatrix} \tag{4.2}$$

The dynamic equations that need to be integrated are defined as follows:

$$\dot{\mathbf{X}} = \begin{Bmatrix} X_3 \\ X_4 \\ F_x/m \\ F_y/m \end{Bmatrix} - \begin{Bmatrix} 0 \\ 0 \\ K_d X_3/m \\ K_d X_4/m \end{Bmatrix} \tag{4.3}$$

where m is the mass of the rover, or actually a dummy mass used to perform the calculations (in this case it has been set to 80 kg). Indeed, the only real output of the integration of the dynamic equations is the trajectory, and so a sequence of coordinates x and y. The obtained velocity $v = (\dot{x}, \dot{y})$ is not the actual speed of the rover in each point of the traversal, but only a dummy velocity that is the result of the applied force field.

Moreover, the two terms $-K_d X_3$ and $-K_d X_4$ are dampers. These damping terms are needed to prevent the system from oscillating harmonically around the attractive point. Indeed, if only conservative forces were present, the system would behave like an ideal spring. Dampers allow the system to lose energy in the same way as air drag does with an oscillating pendulum.

Before proceeding with the description of the different potential fields functions, it must be said that, in the code, instead of calculating the negative gradient of the sum of all different potential fields acting in the rover position of the map, it has been preferred to separately evaluate the negative gradient of the potential field of each attractive or repulsive element and, then, to sum up the forces. The total force $\mathbf{F}$ will be then the sum of the total attractive, $\mathbf{F}_a$, and of the total repulsive force, $\mathbf{F}_r$, acting in that point, as shown in Equation 4.4.

$$\mathbf{F} = \begin{Bmatrix} F_x \\ F_y \end{Bmatrix} = \begin{Bmatrix} F_{a_x} \\ F_{a_y} \end{Bmatrix} + \begin{Bmatrix} F_{r_x} \\ F_{r_y} \end{Bmatrix} \tag{4.4}$$

In the next sections, sink and source artificial potential fields used will be described.

4.2.1 Sink model

The attractive potential field used is a combination of two different functions: a conic linear potential field and a Gaussian field, as also proposed in [22]. Their mathematical expression is shown respectively in the first and in the second term of Equation 4.5:

$$U_a = K_a \rho_a - K_{1_a} e^{(-0.5 K_{2_a} \rho_a^2)} \tag{4.5}$$

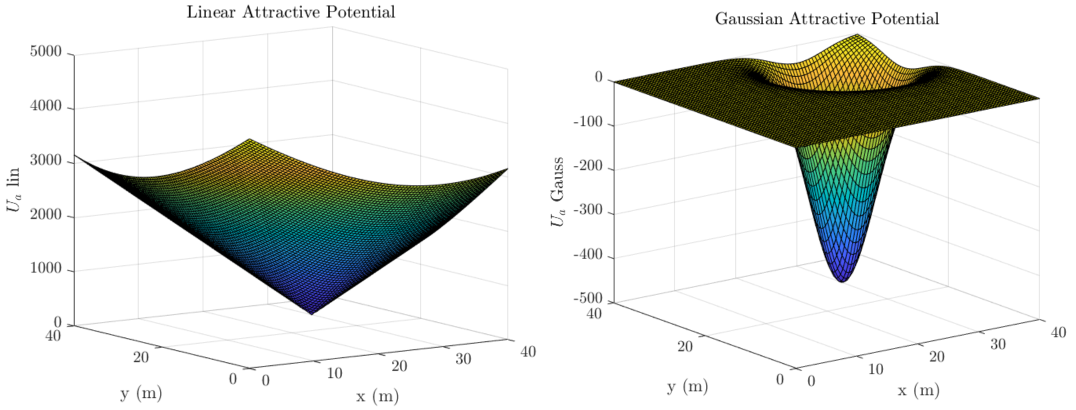
in which K_a, K_{1_a} and K_{2_a} are weighting constants (the values used in this study are reported in Table 4.1) and ρ_a is the distance between the current position of the rover (x, y) and the attractive point (x_a, y_a) and is defined as follows:

$$\rho_a = \sqrt{(x_a - x)^2 + (y_a - y)^2} \tag{4.6}$$

The choice of the conic field has been made because the rover must feel the attractive field in all the map, in order to be always drawn towards the final point and not get trapped between two repulsive fields. Quadratic parabolic potential field has also been tested, but, at greater distances from the attractive point, the gradient of this potential became too large and influential with respect to the repulsive ones, making hazard avoidance difficult. Figure 4.3a shows the 3-D plot of the conic field.

On the other hand, the Gaussian field (Figure 4.3b) is the one that will be used, with an opposite gradient, also for repulsive elements. It mainly acts locally, close to the attractive sink or to the repulsive source. It is a compromise between a triangular field and an exponential field, that solves both their problems. Indeed, the triangular field only acts in a well defined area around the element, while in the exponential field, that guarantees coverage of the whole exploration map, the gradient becomes infinite in the position of the object. The Gaussian function, instead, as shown in Figure 4.4, takes only the advantages of these two functions. The role of this Gaussian field is to increase the attraction of the final point in its proximity, where the presence of some obstacles may lead to getting stuck in local minima.

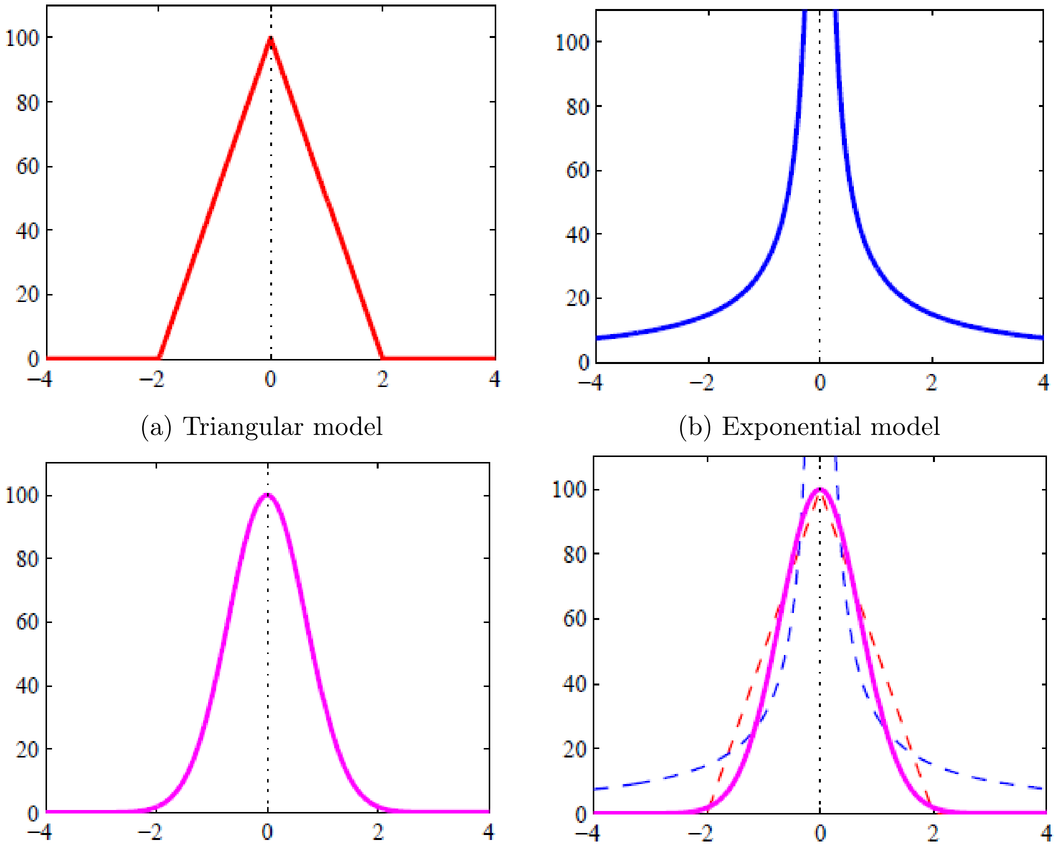
It is possible to observe the total (conic and Gaussian) attractive potential field



(a)

(b)

Figure 4.3. (a) Linear attractive potential field, (b) Gaussian attractive potential field (attractive sink in coordinates: $x = 30\,\mathrm{m}, y = 30\,\mathrm{m}$)



(a) Triangular model

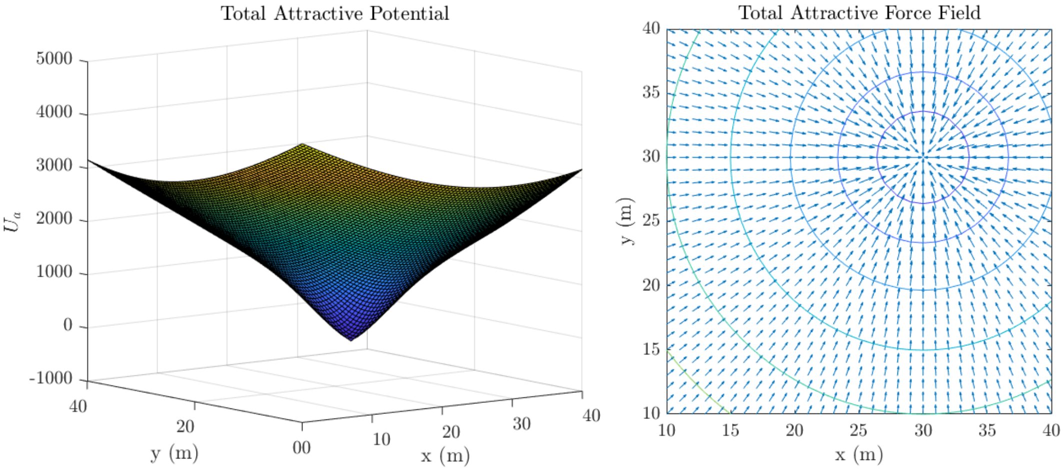
(b) Exponential model

(c) Gaussian model

(d) Comparison

Figure 4.4. Comparison between triangular, exponential and Gaussian potential field functions [22]

in Figure 4.5a, while Figure 4.5b shows the distribution of the attractive force field. Indeed, each arrow represents the negative gradient of the potential. It can be noticed that the magnitude and direction of the force field are constant and symmetric in the all map (because of the conic linear field, that has constant gradient), with changes in proximity of the attractive point due to the presence of the Gaussian field.



(a) (b)

Figure 4.5. (a) 3-D total attractive potential field and (b) its resulting force field (attractive sink in coordinates: $x = 30\,\mathrm{m}, y = 30\,\mathrm{m}$)

Finally, Equation 4.7 shows the mathematical formula for the total attractive force, that is the one actually used in the code at each computational step. The expression is easily obtained as the negative of the 2-dimensional spatial gradient of the potential field function.

$$\mathbf{F}_a = -(\frac{K_a}{\rho_a} + K_{1_a} K_{2_a} e^{(-0.5 K_{2_a} \rho_a^2)}) \cdot \begin{Bmatrix} x - x_a \\ y - y_a \end{Bmatrix} \tag{4.7}$$

A summary of the values, that have been used in the computations, for the different parameters of the attractive potential field is reported in Table 4.1.

Variable	K_a	K_{1_a}	K_{2_a}	K_d
Value	100	500	0.05	300

Table 4.1. Values of the parameters of the attractive potential fields

4.2.2 Source model

Moving to the definition of the repulsive potential field, as previously explained, again, a Gaussian field has been chosen. In this case, however, the potential function is different for each obstacle i, as it is dependent on its enlarged radius ρ_{o_i}. Indeed, two new radius-dependent weight parameters have been introduced according to Equation 4.8 [22].

$$\beta_i = K_{1_r}\rho_{o_i}, \quad \gamma_i = \frac{K_{2_r}}{\rho_{o_i}} \tag{4.8}$$

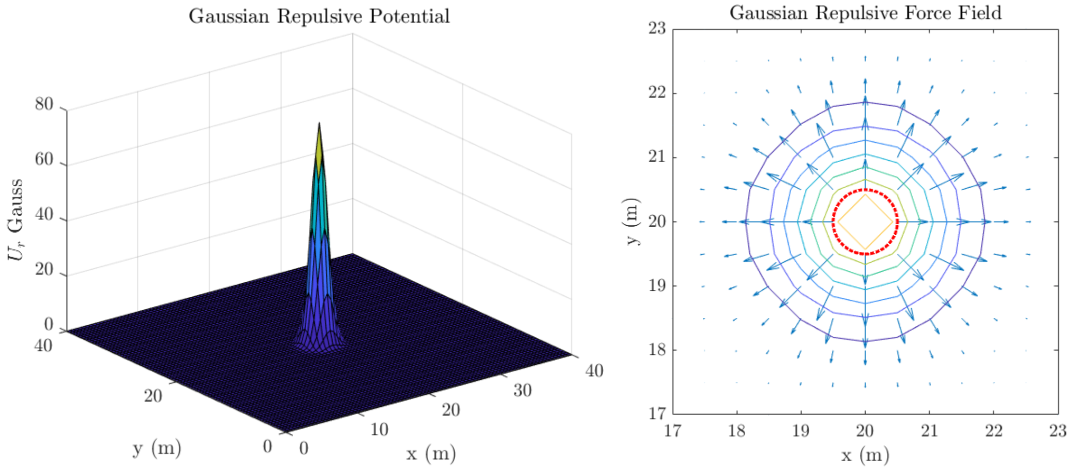
So, the repulsive potential field function is defined as follows:

$$U_{r_i} = \beta_i e^{(-0.5\gamma_i\rho_{r_i}^2)} \tag{4.9}$$

in which ρ_{r_i} is the distance between the current position of the rover (x, y) and the i-th obstacle (x_{o_i}, y_{o_i}):

$$\rho_{r_i} = \sqrt{(x - x_{o_i})^2 + (y - y_{o_i})^2} \tag{4.10}$$

The parameters of the Gaussian function, reported in Table 4.2, are defined in order to model the field in such a way to have very high gradients close to the borders of the obstacles to prevent the rover from crossing them. In Figure 4.6, both repulsive potential and force fields are shown. It can be noticed how the value of the force becomes really high outside the borders of the obstacle (red dashed lines).



(a) (b)

Figure 4.6. (a) 3-D Gaussian repulsive potential field and (b) its resulting force field (repulsive source with $\rho_{o_i} = 0.5\,\mathrm{m}$ in coordinates: $x = 20\,\mathrm{m}, y = 20\,\mathrm{m}$)

As for the attractive field, here is the Gaussian repulsive force used in the code and derived from the potential.

$$\mathbf{F}_{r_i} = K_{1_r}K_{2_r}e^{(-0.5\gamma_i\rho_{r_i}^2)} \cdot \begin{Bmatrix} x - x_{o_i} \\ y - y_{o_i} \end{Bmatrix} \tag{4.11}$$

Vortex field

The main problem of the artificial potential fields method is related to the possibility of getting stuck in local minima. Indeed, there may be some local stationary points in which the sum of all forces is zero. This problem is mainly due to the sum of symmetric fields [25].

Another issue is related to a situation in which the target is within the effect region of an obstacle, such that the repulsive force of the obstacle will push away the rover from the goal point, as depicted in Figure 4.7 [26].

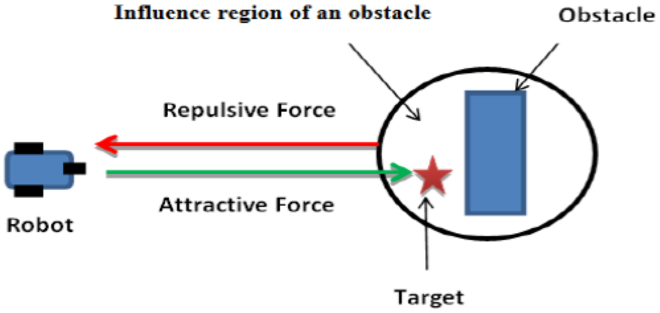


Figure 4.7. Goals non-reachable with obstacle nearby problem [26]

In the quoted article, as well as in [27], a modified artificial potential field is defined with the introduction of the distance between robot and goal into the repulsive force function to guarantee that the global minimum is at the target position. In this way, the repulsive potential field reduces as the rover gets closer to the final point as does the attractive potential field, which is quadratic in these two studies and so its gradient becomes smaller as the rover comes closer to the final point.

In the context of this thesis, instead, this last problem has been solved thanks to different expedients. First of all, the conic linear attractive field assures that the rover feels the same gradient in all the environment allowing a proper tuning of the repulsive fields, which do not need to be oversized to oppose the gradient of a quadratic field, that gets really high at large distances from the attractive point. In this way, also, a strong sink gradient is present in proximity of the target point and the attraction is even increased by the addition of the Gaussian field. Moreover, this situation in which the target is in the range of influence of a hazard does not, actually, occur so frequently in the scenario of this study, as target points that are too close to obstacles are discarded, because considered too dangerous.

Regarding the symmetry problem that leads to local minima, different solutions have been proposed in literature. There are two main possibilities. The first one, as explained in [28], is to introduce a tangential repulsive force that guides the rover along the contours of the obstacle, towards the goal. The second option is to define a swirling field, that is an asymmetric field generated through a combination of radial (repulsive) and tangential (vortex) fields [25].

In this work, it has been decided to use the swirling field idea, which seemed best as it both makes the field asymmetric and it lets the rover turn around the obstacles.

According to [29], the vortex force field for the i-th obstacle is defined as:

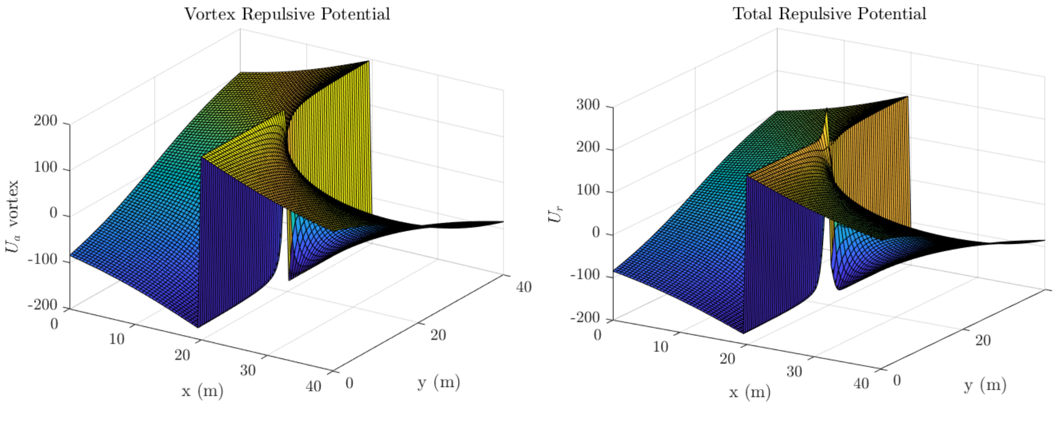
$$\mathbf{F}_{v_i} = \left\{ \begin{matrix} -K_3 \frac{y-y_{o_i}}{\rho_{r_i}^2} \\ +K_3 \frac{x-x_{o_i}}{\rho_{r_i}^2} \end{matrix} \right\} \tag{4.12}$$

where K_3 is a weight parameter. This force field assigns in each position (x, y) of the map a vector which is tangent to the circle centred in (x_{o_i}, y_{o_i}) and is oriented in the counterclockwise direction.

The primitive function $(-U_v)$ of the linear differential $F_{v_x} dx + F_{v_y} dy$ is not globally defined, as the components of $\mathbf{F}_{v_i}$ are defined in a domain which is not simply connected ((x_{o_i}, y_{o_i}) must be removed) and so the integral over a circle with centre in O is nonzero [29]. As a result, the potential U_{v_i} is a multi-valued function that can be expressed as:

$$U_{v_i} = -K_3 \arctan\left(\frac{y - y_{o_i}}{x - x_{o_i}}\right) + c \tag{4.13}$$

where c is a constant. As shown in Figure 4.8, U_{v_i} represents a helical surface. It is possible to notice how the gradient in some regions would be infinite, since the value of the potential function instantly changes, because of the domain problem.

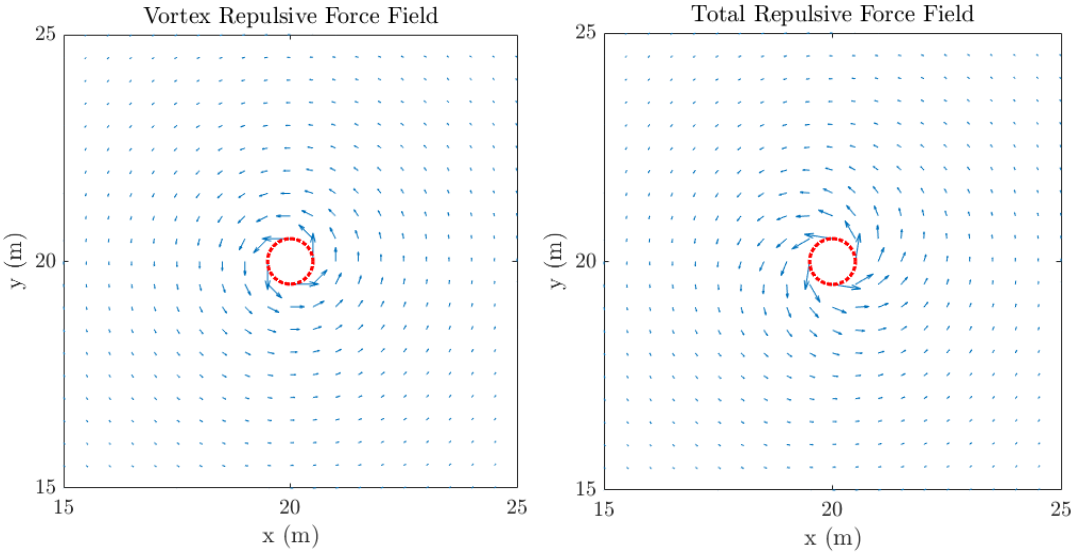


(a)

(b)

Figure 4.8. (a) Vortex potential field, (b) total (Gaussian and vortex) repulsive potential field (repulsive source in coordinates: $x = 20\,\text{m}, y = 20\,\text{m}$)

For these issues related to the integration of the vortex force field, its components are directly added as forces with Equation 4.12 and not derived from the potential. In Figure 4.9a, an example of vortex force field is presented. It is possible to notice that the arrows are tangential circles centred in O, as expected. Figure 4.9b shows the total (Gaussian and vortex) repulsive swirling force field of the same obstacle.



(a)

(b)

Figure 4.9. (a) Vortex force field, (b) total (Gaussian and vortex) repulsive force field (repulsive source with $\rho_{o_i} = 0.5\,\mathrm{m}$ in coordinates: $x = 20\,\mathrm{m}, y = 20\,\mathrm{m}$)

It is, also, important to properly define the direction of vortex rotation. In order to avoid problems of dynamic oscillations due to continuous changes in the direction of rotation, a system of activation-deactivation of the vortexes has been developed. The vortex of an obstacle is turned on only within a circular area of interest, centred in O and with a radius of $r_i = 1.3 \cdot \rho_{o_i} + 0.5\,\mathrm{m}$, and the direction of vortex rotation is kept constant until the rover exits the circular area and the vortex is turned off.

Two versors are defined: $\hat{v}_{o-t}$ and $\hat{v}_{r-o}$, which represent respectively the direction between the obstacle and the target and between the rover and the obstacle. The rover is in the position in which it has entered the area of interest of the obstacle. The cross product of these two versors is evaluated:

$$\hat{b} = \hat{v}_{o-t} \times \hat{v}_{r-o} \tag{4.14}$$

The third component of $\hat{b}$ determines the direction of rotation:

$$\begin{cases} if\ b_3 > 0 \longrightarrow \theta = +1 \longrightarrow \text{Counter-Clockwise} \\ if\ b_3 < 0 \longrightarrow \theta = -1 \longrightarrow \text{Clockwise} \end{cases} \tag{4.15}$$

Where θ is a coefficient that multiplies $\mathbf{F}_{v_i}$ determining the direction of rotation of the vortex. If $\theta = +1$ the direction is the standard one, which is counter-clockwise, as stated before.
Figure 4.10 depicts the reasoning behind this method. Given the position of the rover in Figure 4.10a, it is clear how it would be easier for the vehicle to pass on the right side of the obstacle to get to the target. Therefore, the definition of the vortex rotation as counter-clockwise will force the rover to do so. The cross product is simply used as a way to determine relative positions of rover, obstacle and target and to evaluate the best direction to avoid the obstacle. Of course, in Figure 4.10b the opposite situation is shown. As said, once the vortex is activated its direction of rotation remains the same until it is turned off once the rover exits the circular area of interest.

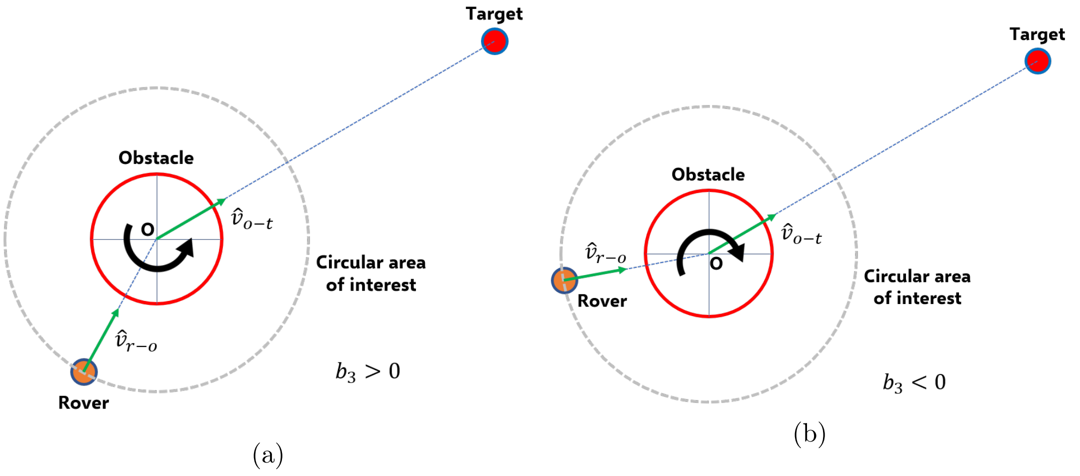


Figure 4.10. Illustration of how the system of selection of the vortex direction of rotation works. In (a) the rover will pass on the right side of the obstacle as the vortex is activated in the counter-clockwise direction, in (b) the vehicle will move on the left side of the obstacle because of the clockwise rotation of the vortex

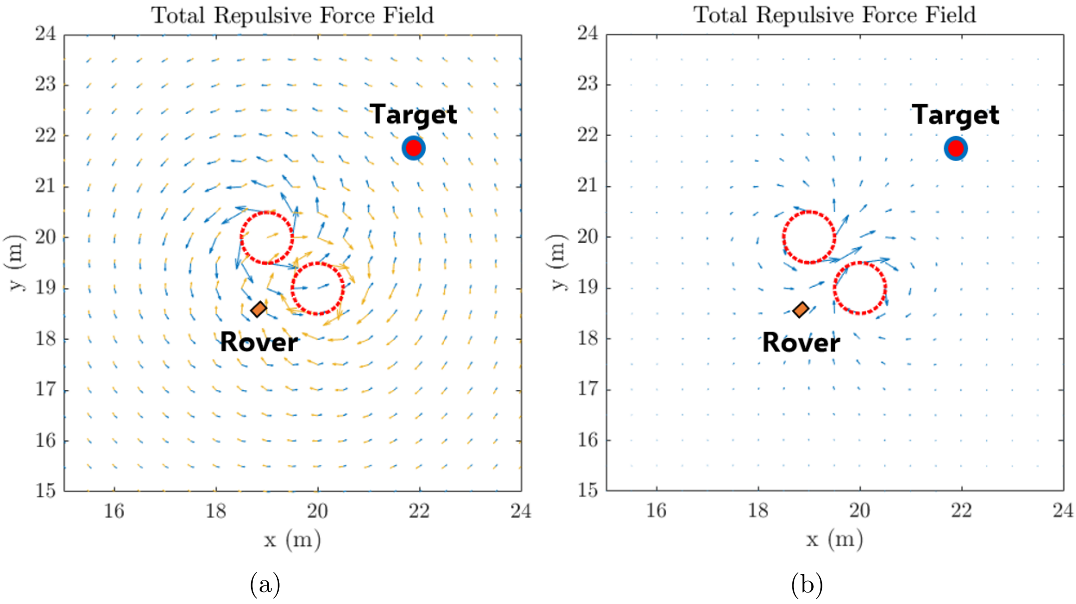
In Table 4.2, the values of the parameters of the repulsive potential fields used in the code are reported. It can be noticed that different values have been set, depending on the radius of the obstacles. Indeed, for larger hazards, specific tuning of the parameters of the fields has resulted to be necessary, based on the ratio between repulsive and vortex field.

	$\rho_i < 2\,\mathrm{m}$			$\rho_i > 2\,\mathrm{m}$		
Variable	K_{1_r}	K_{2_r}	K_3	K_{1_r}	K_{2_r}	K_3
Value	160	0.63	120	190	0.55	150

Table 4.2. Values of the parameters of the repulsive potential fields

Close obstacles

Another problem to be addressed concerns the necessity of keeping the rover away from narrow passages between two obstacles avoiding dangerous manoeuvres. For how the swirling field is defined, the rover is actually encouraged to pass between hazards. In fact, if the vehicle is positioned as in Figure 4.11, once in the circular area of interest of the two obstacles, the vortexes are activated in opposite directions (Figure 4.11a) and their summation field pushes the rover between them, as depicted in Figure 4.11b. Indeed, according to the system shown in Figure 4.10, it would be better to pass on the right side of one obstacle and on the left of the other one.



(a)

(b)

Figure 4.11. (a) Swirling potential fields of the two obstacles activated in opposite directions, according to the position of the rover, and (b) the resulting field

This system is good for obstacles that are far enough from each other, because it allows the algorithm to generate smooth trajectories letting the vehicle pass also between obstacles. However, for hazards that are too close to each other (closer than a defined distance), a method that permits the rover to avoid them completely shall be provided to prevent dangerous paths.

Swirling potential fields have been exploited also in this case. Indeed, by activating both vortexes in the same direction, the repulsive field that is generated induces the rover to circumnavigate both the hazards instead of passing between them, as shown in Figure 4.12.

From a practical standpoint, it usually happens that the rover enters the circular area of interest of one obstacle first. Therefore, in the beginning, just the vortex of this obstacle is activated and then, when the other needs to be turned on, it is

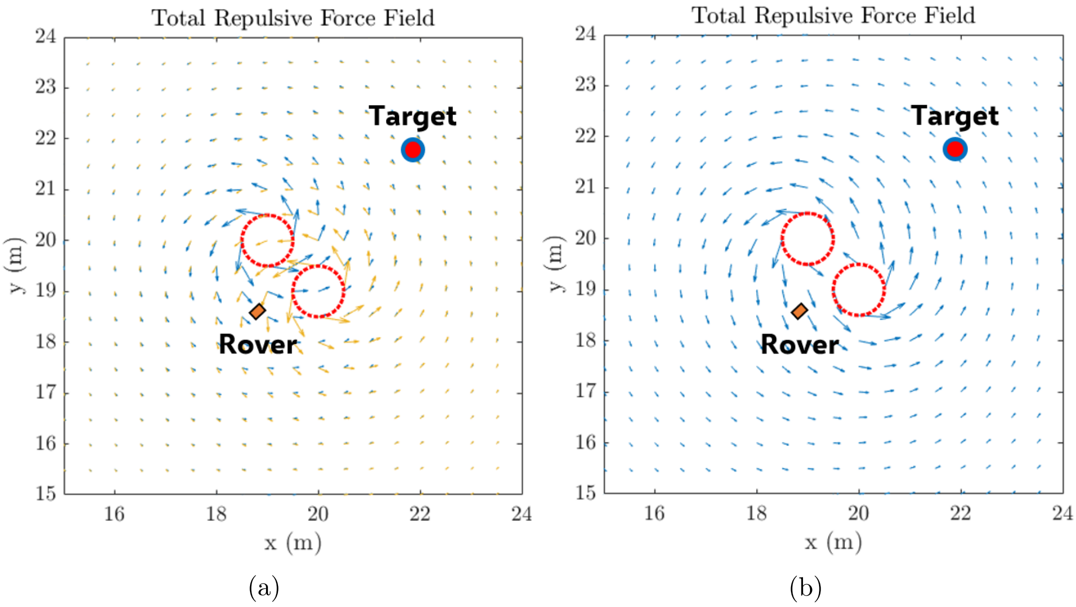


Figure 4.12. (a) Swirling potential fields of the two obstacles activated in the same direction, according to the position of the rover, with the new system used if the hazards are close to each other and (b) the resulting field

set in the same direction of rotation of the previous one. Instead, if the vehicle enters both regions simultaneously, the direction of rotation is chosen with the system described in the previous paragraph, but considering the barycentre of the two obstacles (based on their radii) as point O of Figure 4.10.

If more than two obstacles are in this condition of proximity, all their vortex fields will have the same direction of rotation.

Chapter 5

Simulations and results

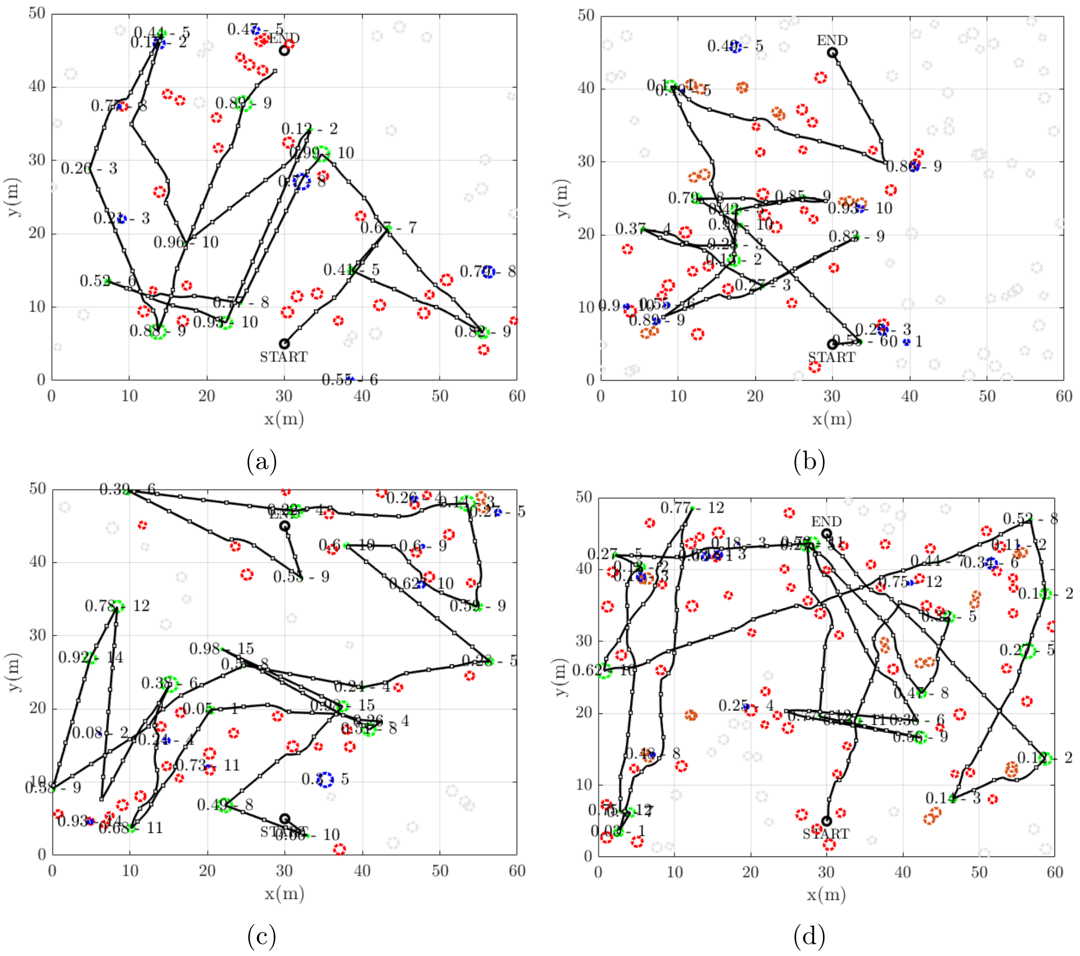
5.1 Simulation environment

Before presenting the results, it is necessary to explain some preliminary concepts regarding how the simulations have been carried out.
Obstacles and scientific points are randomly generated in the 2-D map in all their properties in a number defined by the user. Of course, category and Scientific Value Score of the targets are related to each other, as defined in Equation 2.4. As said, at the beginning of the exploration, the rover is aware of the scientific targets but not of the hazards, except for the large Non-Traversable Zones (NTZ).
Computational time has been used as budget of each Monte Carlo search and has been defined to be proportional to the number of scientific targets still to be reached.
In section 5.2, the results obtained from the application of the path planning algorithm to different environments and scenarios are presented. In the second place, a statistical sensitivity analysis, carried out on the data obtained by multiple runs of the algorithm, verifies its effectiveness and robustness.

5.2 Multi-objective Pareto MCTS results

5.2.1 General

The general scenario refers to cases in which only small hazards and "traditional" scientific targets are present within the map. The term "traditional" is used as an opposite of "high SVS" point, which is a scientific point that has a reward higher than the defined maximum value of 1. Indeed, in section 5.2.3, a high SVS point is introduced to represent an element that has such a great importance for the mission and that is so uncommon that would be a mistake not to sample it. An example may be a very rare indicator for possible traces of life. The



(a)

(b)

(c)

(d)

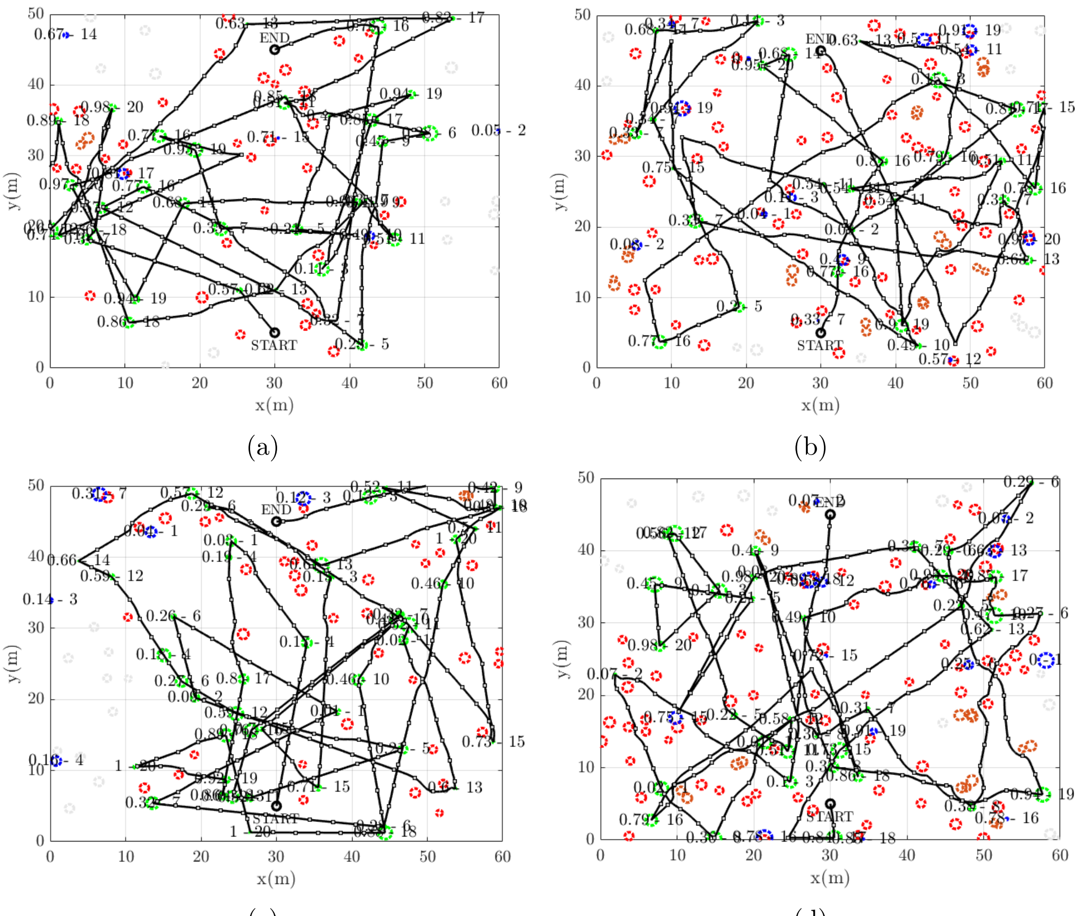
Figure 5.1. Global paths obtained in different general scenarios (Part 1)

behaviour of the algorithm in this scenario has been evaluated as well.

In Figure 5.1 and Figure 5.2, some examples of the application of the algorithm to different environments with increasing number of scientific points, categories and obstacles are reported. In all the figures, the obstacles that are brown coloured (instead of red) represent close hazards (cf. section 4.2.2) and the grey ones are the obstacles not yet discovered by the vehicle.

Once again, it must be highlighted how the global planning method is not a global optimiser, but determines at each cycle the most promising next move, by considering all the competing objective functions. Therefore the trajectory will be globally sub-optimal. This explains why, especially for high numbers of scientific points, the global path may not seem to be the best option to follow. The trajectory is a compromise between the different objectives.

Table 5.1 shows the numerical results of the path planning method applied to the scenarios in the figures in terms of total number of scientific points visited by the rover ($\mathbf{n_{sci}\ vis}$), number of different categories of targets sampled ($\mathbf{n_{cat}\ vis}$), total reward ($\mathbf{rew_{tot}}$) and path length ($\mathbf{path\ l.}$).



(a)

(b)

(c)

(d)

Figure 5.2. Global paths obtained in different general scenarios (Part 2)

Scenario				Results			
	$\mathbf{n_{sci}}$	$\mathbf{n_{cat}}$	$\mathbf{n_{obs}}$	$\mathbf{n_{sci}}$ **vis**	$\mathbf{n_{cat}}$ **vis**	$\mathbf{rew_{tot}}$	**path l.**
Figure 5.1a	20	10	50	13/19	8/8	8.66/11.48	303 m
Figure 5.1b	20	10	100	11/13	9/9	5.58/6.07	223 m
Figure 5.1c	30	15	50	21/25	12/13	10.79/11.68	410 m
Figure 5.1d	30	15	100	21/22	11/11	8.80/9.14	483 m
Figure 5.2a	40	20	50	35/37	16/17	21.95/23.11	705 m
Figure 5.2b	40	20	100	27/27	13/13	15.16/15.16	714 m
Figure 5.2c	50	20	50	45/49	19/19	22.68/23.14	782 m
Figure 5.2d	50	20	100	36/39	18/18	16.79/16.93	702 m

Table 5.1. Results obtained in different general scenarios

If a certain target is found to be in a too dangerous position, it is removed from the interesting points. This explains why, in the environment of Figure 5.1b, for instance, which has been generated with $n_{sci} = 20$, only 13 of them are actually considered as reachable targets. Of course, the same process happens for the number of categories. In this case, however, the categories located within the area might be less than n_{cat} from the beginning. Indeed, for example, $n_{cat} = 15$ is the maximum number of possible categories that can be randomly assigned to the targets, but it is not certain that all of them will be actually present in the area.

Considering the results, it can be noticed how the scientific points are rarely all sampled, since visiting all of them strongly competes with the objective of having a short path length. Moreover, the targets that are usually not reached are the ones that are located in proximity of the borders of the exploration area (see for instance Figure 5.2c), showing the influence of the objective function f_4 (cf. section 3.2). On the other hand, the rover nearly always visits targets belonging to all the different reachable categories.

Figure 5.1a displays an example of a situation in which the rover discovers that the ending point is not in a safe position as it is approaching it and, so, it stops without reaching it.

5.2.2 Non-Traversable Zones

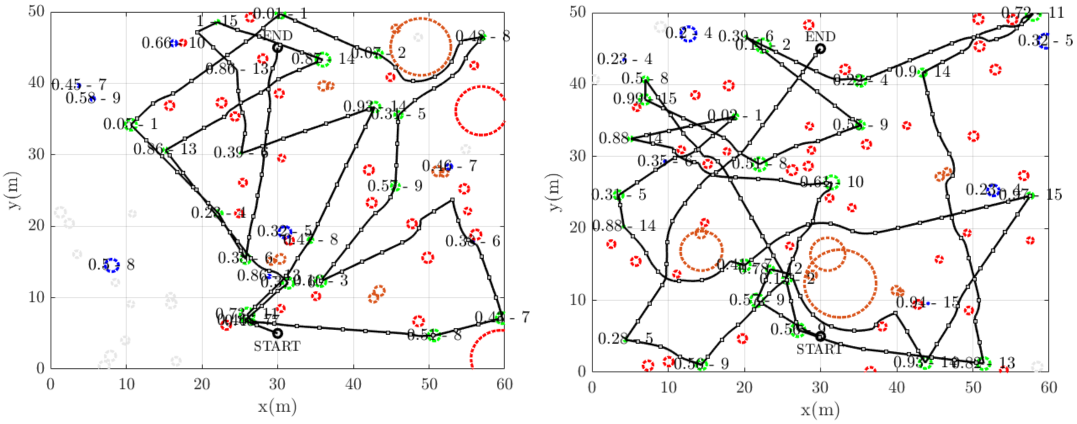
Non-Traversable Zones (NTZ) are large areas that the rover knows from the beginning of the exploration and that represent larger scale features like big rocks, craters, steep hills and others that could be hazardous for the vehicle to move through.

The algorithm has been tested in these environments to prove its capability of avoiding also these larger type obstacles. Two examples are reported in Figure 5.3, along with their numerical results in Table 5.2.

	Scenario			**Results**			
	$\mathbf{n_{sci}}$	$\mathbf{n_{cat}}$	$\mathbf{n_{obs}}$	$\mathbf{n_{sci}}$ **vis**	$\mathbf{n_{cat}}$ **vis**	$\mathbf{rew_{tot}}$	**path l.**
Figure 5.3a	30	15	50	22/26	13/13	10.62/12.81	381 m
Figure 5.3b	30	15	50	24/29	14/14	13.62/15.66	511 m
Figure 5.4	30	15	50	25/27	12/13	12.87/12.92	559 m

Table 5.2. Results obtained in different scenarios with NTZ

By looking at the figures, it can be noticed how the rover is able to avoid these wide areas during the exploration. Nevertheless, further analyses to evaluate the satisfaction of the safety constraint have been performed by applying the planning method to a higher number of different scenarios and the results are



(a)

(b)

Figure 5.3. Global paths obtained in different scenarios with NTZ

reported in section 5.3.

Despite the introduction of the fourth objective function f_4 (cf. section 3.2), that induces the rover to avoid targets that are located close to the borders, it still may happen that the force produced by the artificial potential fields pushes the vehicle outside the domain, as depicted in Figure 5.4. Of course, this may lead to a dangerous situation, since the rover crosses the borders of the exploration area. However, the vehicle itself has instruments able to detect new elements and hazards in this unknown environment and, since the local path planning algorithm has been designed with the step by step approach, it would be able to navigate also in this region without increasing the risk too much.

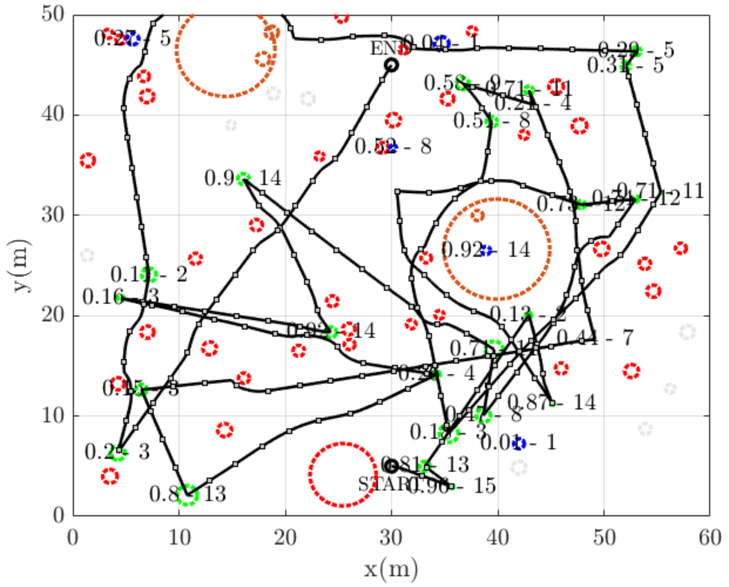
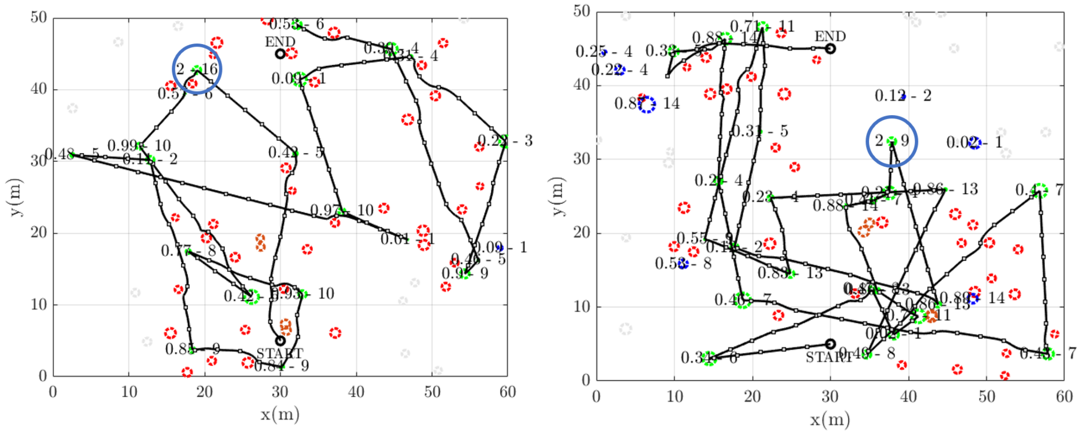


Figure 5.4. Situation in which the rover is forced to pass outside of the area previously explored by the helicopter/drone

5.2.3 High SVS

As previously explained, the algorithm has been tested also in situations in which a target with a very high scientific reward is present. In the scenarios depicted in Figure 5.5, a point that has a SVS two times higher than the maximum of 1 has been introduced. In these cases the global path planning method nearly always decides to sample that target. Moreover, the vehicle often chooses to visit this valuable point pretty soon in the exploration.
Table 5.3 reports the numerical results related to the two examples of Figure 5.5.



(a)

(b)

Figure 5.5. Global paths obtained in different scenarios with a high SVS point

Scenario				Results			
	$\mathbf{n_{sci}}$	$\mathbf{n_{cat}}$	$\mathbf{n_{obs}}$	$\mathbf{n_{sci}}$ **vis**	$\mathbf{n_{cat}}$ **vis**	$\mathbf{rew_{tot}}$	**path l.**
Figure 5.5a	20	10	50	20/21	10/10	12.26/12.35	347 m
Figure 5.5b	30	15	50	24/29	11/11	13.58/14.77	424 m

Table 5.3. Results obtained in different scenarios with a high SVS point

Figure 5.5a also presents the particular situation in which the rover finds out that the ending point is not reachable, before it is actually moving towards it. Indeed, in this case the obstacle that is close to the ending point is detected when the vehicle is in the vicinity of the target 0.09-1, as it passes near the ending point. Therefore, the rover understands that it will not be able to reach the ending point, since it is not safe, but it continues its exploration-planning phases until it reaches the point before the end and it stops there.

5.3 Sensitivity analysis

5.3.1 Definition of the parameters of merit

In order to verify the robustness of the designed system, it is necessary to apply it to a large number of different scenarios and to analyse its performances, by evaluating some established parameters of merit.
In particular, these parameters need to assess two main aspects: the efficiency and quality of the planning algorithm and its safety.
Efficiency and quality aim to evaluate whether the designed trajectory reflects and respects the objective functions. Therefore, three parameters have been considered. The first one is the percentage of the scientific points that have been sampled, with respect to the total number of the ones that are located in safe positions according to the knowledge of the rover. The second one is the percentage of visited categories. These two parameters verify how the method is able to exploit the scientific information present within the exploration area.
The last parameter is the total path length, which is connected with the energy needed by the rover to follow the designed trajectory and should be minimised.
The safety of the system, instead, is evaluated by analysing how the rover is able to perform its operations among the hazards without crashing with them. The parameter that has been considered is the minimum distance d between vehicle and obstacle in each close passage. Close passages are defined as situations in which the rover passes at a distance lower than two times its width from the actual borders of an obstacle. Remembering how the enlarged radius of a hazard is defined:

$$\rho_o = r_{obs} + hw_{rov} + sf \tag{5.1}$$

where r_{obs} is the actual radius of the obstacle, hw_{rov} is the half-width of the rover (0.3 m) and sf is the safety factor (0.1 m), the vehicle performs a close passage to a certain hazard if the distance d between its centre and the border $r_{obs} + hw_{rov}$ of the obstacle is lower than $3 \cdot hw_{rov} = 0.9\,\mathrm{m}$:

$$\text{close passage if: } d \leq 0.9\,\mathrm{m} \tag{5.2}$$

The boundary situation in which $d = 0.9\,\mathrm{m}$ is depicted in Figure 5.6, along with the plot of the different definitions of the radius of the obstacle.
In order to satisfy the safety factor constraint, the distance d shall be always greater than $sf = 0.1\,\mathrm{m}$.
Finally, a summary of the parameters used for the sensitivity analysis is reported in Table 5.4.

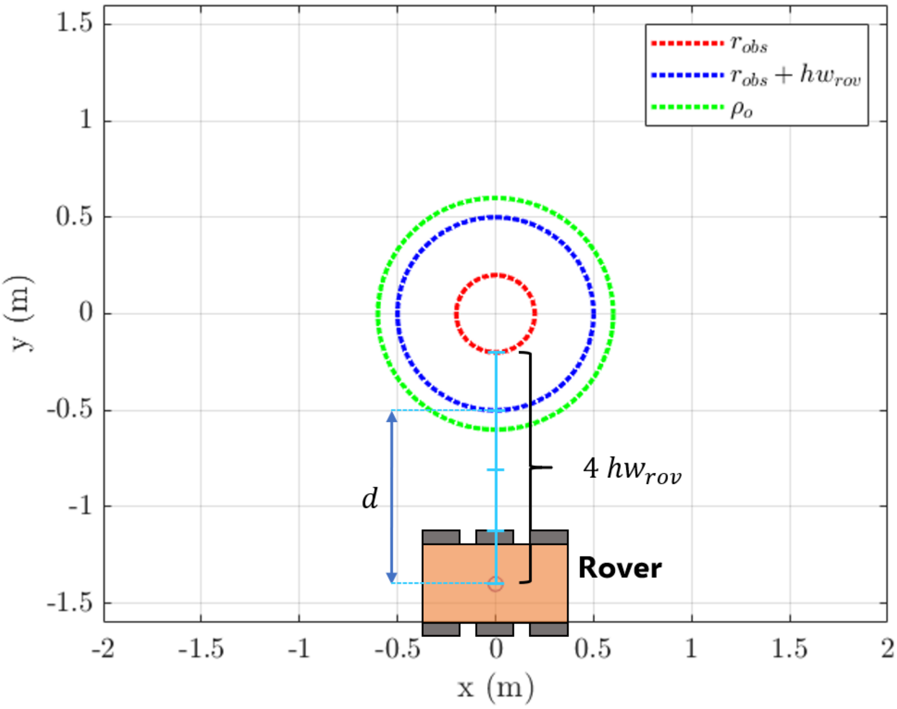


Figure 5.6. Close passage definition in relation to the enlarged radius of the obstacles

Efficiency and quality			Safety
Sampled points (%)	Sampled categories (%)	Path length (m)	$d ><= sf$ (m)

Table 5.4. Summary of the parameters used for the sensitivity analysis

5.3.2 Results

The sensitivity analysis has been performed in different ways to evaluate the response of the algorithm. In the following sections, several applications are presented along with their results.

Different environments

First of all, the system has been tested on variable scenarios, by running 100 simulations on different environments with a predefined number of scientific targets, categories and obstacles.

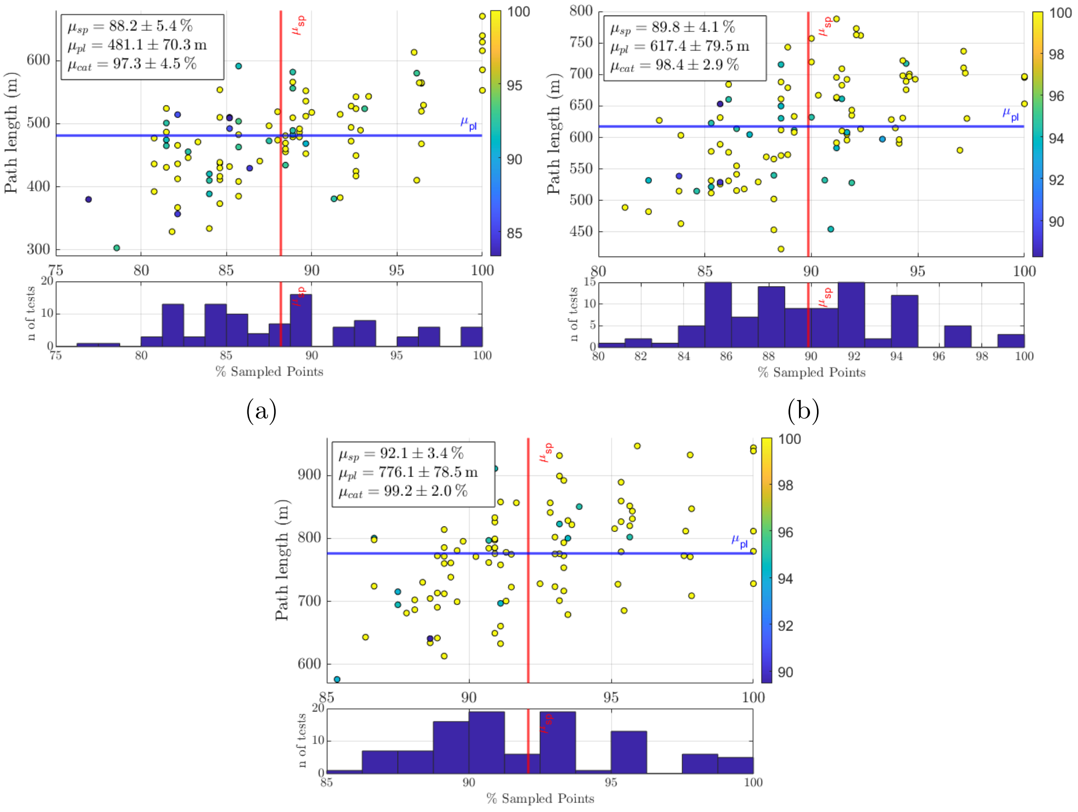
Figure 5.7 depicts the results of the simulations concerning quality and efficiency of the method. Indeed, each point in the graphs represents a run of the algorithm from the starting to the ending point on a randomly generated environment and it is positioned in the diagram according to the percentage of sampled targets (x axis) and the total path length (y axis). Moreover, the percentage of visited categories is shown by the colour of the point according to the colour-bar on the right. The plots also display the mean values of percentage of sampled points (μ_{sp}), path length (μ_{pl}) and percentage of sampled categories (μ_{cat}), ± their standard deviation.

The point distributions, as could be expected, show a conflicting connection between the percentage of the sampled points and the total path length. The simulations in which all scientific targets have been visited, generally, present a total path length greater than the average. Moreover, most of the points are approximately located over a sloped line that assigns greater values of path length as the percentage of visited targets increases. In any case, the mean value μ_{sp} is about 90%, which is a good value. The standard deviation is quite small and decreases as the number of scientific points present within the area becomes greater.

Although the targets are rarely all sampled, the rover often takes samples from all the different reachable categories in the area. Indeed, μ_{cat} is always greater than 95%, with a low value of standard deviation.

Of course, the mean of path length presents higher values as the number of scientific points in the area increases, while its standard deviation remains almost constant (about 70 m).

Regarding the safety of the method, Figure 5.8 shows that the percentage of satisfaction of the safety factor constraint ($d > 0.1\,\mathrm{m}$) is always higher than 95%. Indeed, only in 2 or 3 tests out of 100 the rover happened to pass at a distance lower than 0.1 m from an obstacle. The percentage of hit obstacles is greater in the cases in which more scientific targets are present within the area, since the rover travels longer trajectories and, so, the number of potentially dangerous close passages is larger.

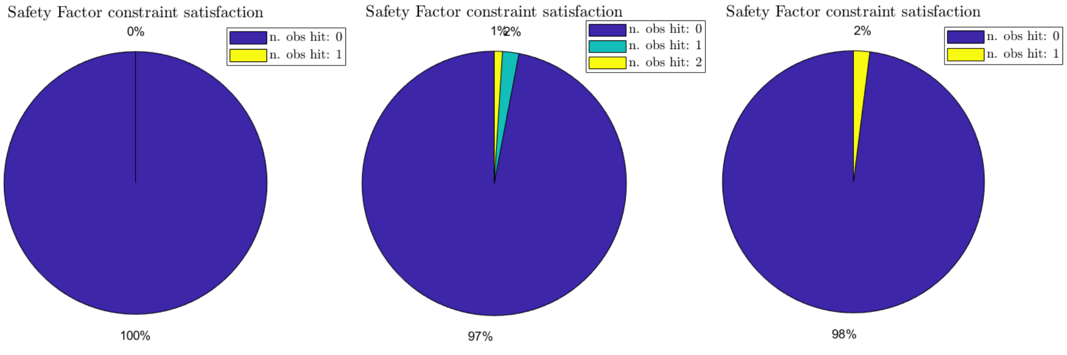


(a)

(b)

(c)

Figure 5.7. Results of the efficiency and quality parameters: percentage of sampled points (x axis), path length (y axis) and percentage of sampled categories (colour-bar). The histogram below displays the distribution of the output parameter: percentage of sampled points. The simulations have been performed on 100 randomly generated environments with: (a) 30 sci, 15 cat, 50 obs; (b) 40 sci, 20 cat, 50 obs; (c) 50 sci, 20 cat, 50 obs



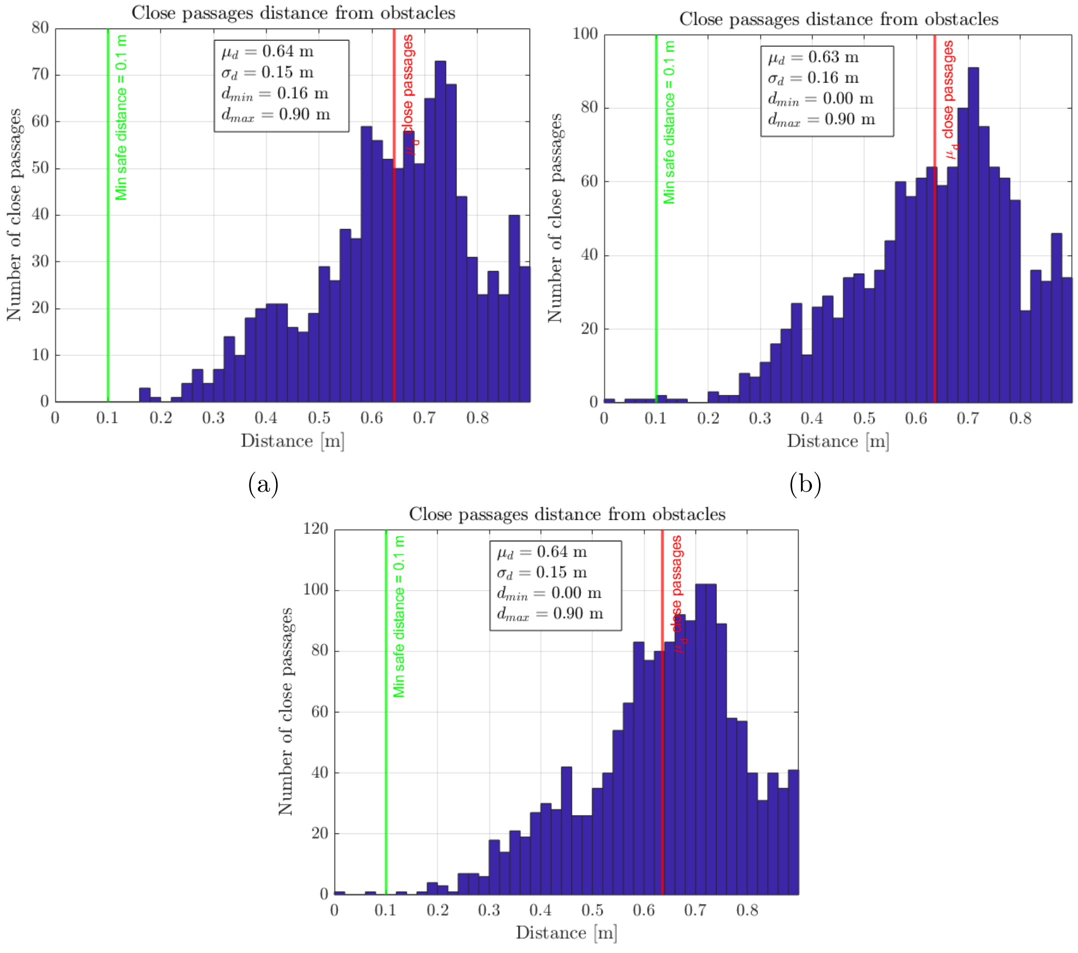
(a)

(b)

(c)

Figure 5.8. Percentage of satisfaction of the safety factor constraint ($d > 0.1\,\mathrm{m}$) in 100 randomly generated scenarios with respectively (a) 30 sci, 15 cat, 50 obs; (b) 40 sci, 20 cat, 50 obs; (c) 50 sci, 20 cat, 50 obs

As shown in Figure 5.9, the average minimum distance μ_d from the borders of the obstacles during close passages is about 0.6 m, with a standard deviation around 0.15 m. In this way, the vehicle usually passes at a good distance (not too far and not too close) from hazards. Indeed, a too far passage would be a waste of energy for travelling more than actually needed to avoid an obstacle.



(a)

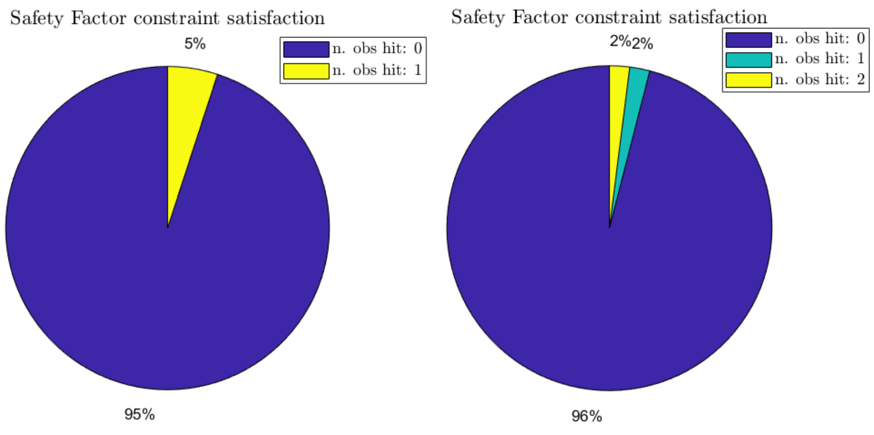
(b)

(c)

Figure 5.9. Close passages distance distribution in 100 randomly generated scenarios with respectively (a) 30 sci, 15 cat, 50 obs; (b) 40 sci, 20 cat, 50 obs; (c) 50 sci, 20 cat, 50 obs

Figure 5.10 and Figure 5.11 depict the percentage of satisfaction of the safety factor constraint in harsher environments. In particular, the plots in Figure 5.10 have been obtained for environments in which large NTZ were present. It is possible to see how the percentage of failures is higher than before, but it still is less than 5%.

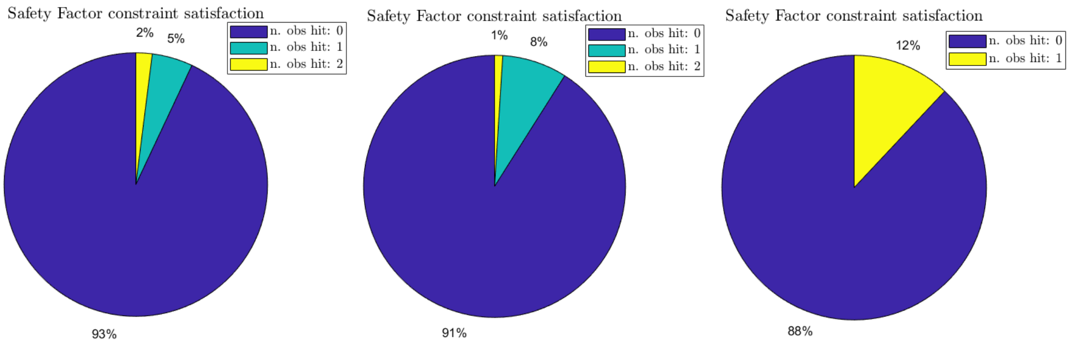
The safety of the algorithm becomes lower as the number of obstacles within the area gets really high. For instance, in environments with 100 obstacles the percentage of tests in which the safety factor constraint is not satisfied at least in one close passage is higher than 5%, as shown in Figure 5.11.



(a)

(b)

Figure 5.10. Percentage of satisfaction of the safety factor constraint ($d > 0.1\,\mathrm{m}$) in 100 randomly generated scenarios with NTZ and respectively (a) 30 sci, 15 cat, 50 obs; (b) 40 sci, 20 cat, 50 obs



(a)

(b)

(c)

Figure 5.11. Percentage of satisfaction of the safety factor constraint ($d > 0.1\,\mathrm{m}$) in 100 randomly generated very harsh environments (100 obstacles) and respectively (a) 30 sci, 15 cat, 100 obs; (b) 40 sci, 20 cat, 100 obs; (c) 50 sci, 20 cat, 100 obs

Variability analysis - same environment

In order to evaluate the variability of the output trajectory, the algorithm has been applied multiple times to the same scenario and the variations on the sensitivity parameters have been analysed.

Figure 5.12 reports the results obtained from 100 applications of the planning algorithm to the same environment. It can be noticed that the variability of the solutions is pretty high, showing more or less the same values of standard deviations of the case with different environments displayed in Figure 5.7a. This is, mainly, due to the high number of objective functions that compete against each other and to the random selection of the best child from the Pareto front.

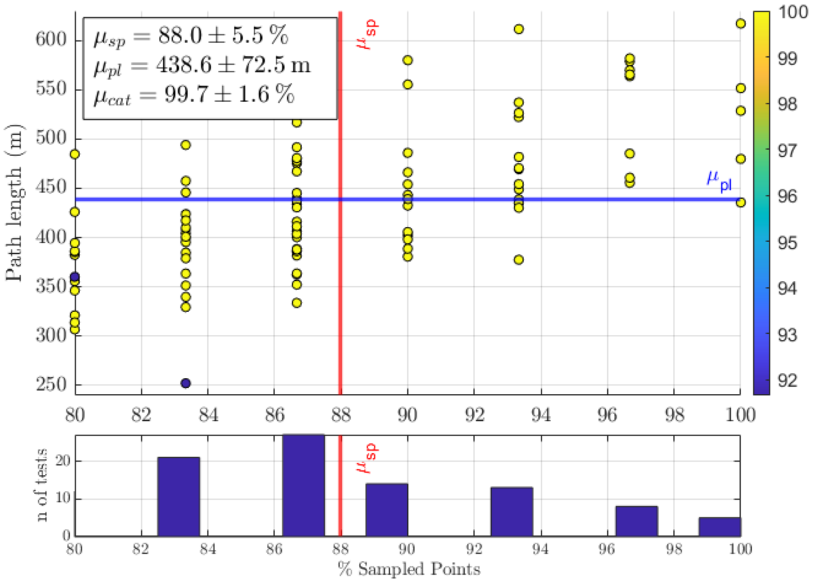


Figure 5.12. Results of the efficiency and quality parameters of 100 simulations performed on the same environment with: 30 sci, 15 cat, 50 obs

It must be highlighted how this variability is referred to the global path that the rover has followed once the exploration phase is over and not to the output of the next best step that is determined by a single run of the algorithm. Indeed, this output is more stable, as shown in Figure 5.13, in which the green numbers represent the number of times a certain scientific point has been selected as next move from the starting point in 100 runs of the algorithm in the displayed scenario. It can be seen how the same two points have been chosen in the 50% of the simulations.

What makes the global path so variable is the sum of all these different possibilities that the algorithm could give at each selection of the next step, leading to a tree of different overall trajectories.

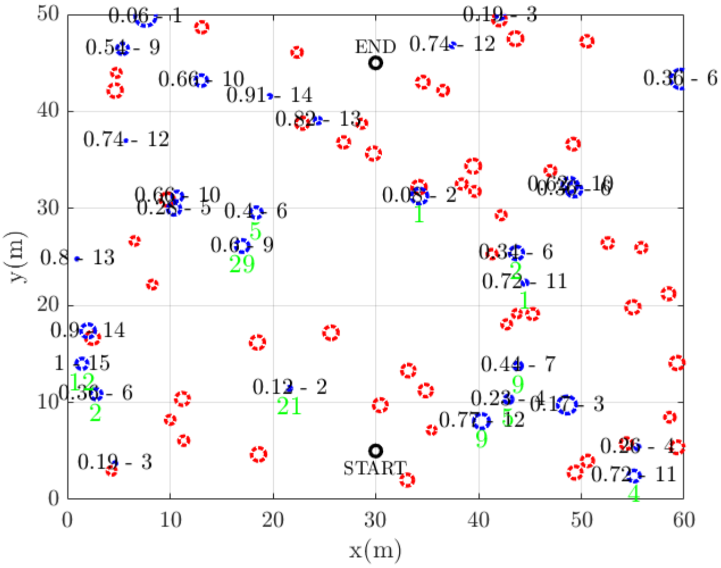


Figure 5.13. Number of times each scientific point in the map has been selected as next step from a single run of the MCTS. If a number is not reported, the target has never been selected

Dependence on initial conditions - start/end switch

Besides, the dependence on the initial conditions has been tested by switching starting and ending point of a scenario, as a way to invert the time direction of exploration, that will be performed in the opposite way.

An example is reported in Figure 5.14, in which triangles represent simulations carried out from the ending to the starting point. It is possible to notice that the results are more or less similarly distributed, showing that there is not a strong dependence on initial conditions and on the direction of exploration. Indeed, the system is not too bonded to the starting position, as the rover can widely explore the area from the beginning.

Non-absolute dominance

Finally, some attempts to analyse the variability of the sensitivity parameters, after the definition of the variable δ_d for the non-absolute dominance, have been carried out.

In particular, four different tests have been performed, each time considering the d-th objective function as less important, by defining $\delta_d = 1$. A comparison between the resulting mean values of the sensitivity parameters of quality obtained with different definitions of dominance is displayed Table 5.5.

Figure 5.15a reports the results obtained in the case in which the objective function f_1, connected with the total SVS collected along the path, is considered to be a less important goal and, so, δ_1 has been set to 1. It is possible to notice how, in this case, the algorithm never reaches 100% of sampled points and how

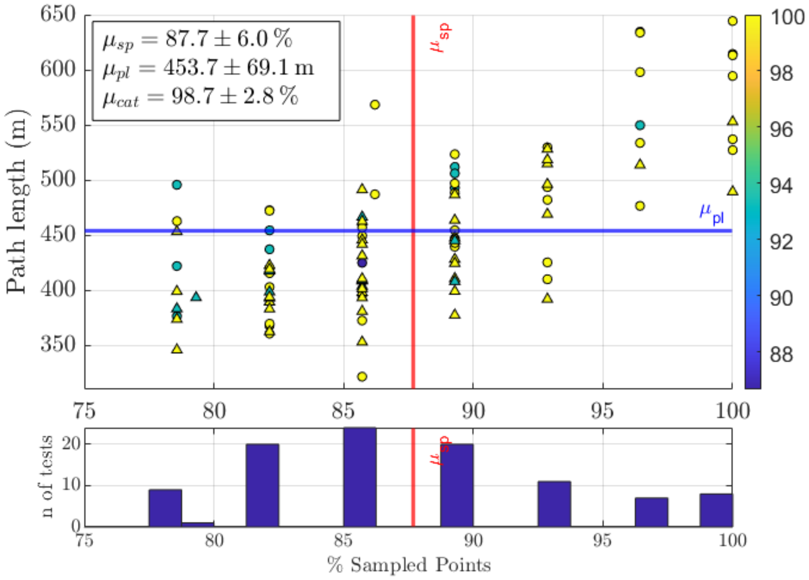
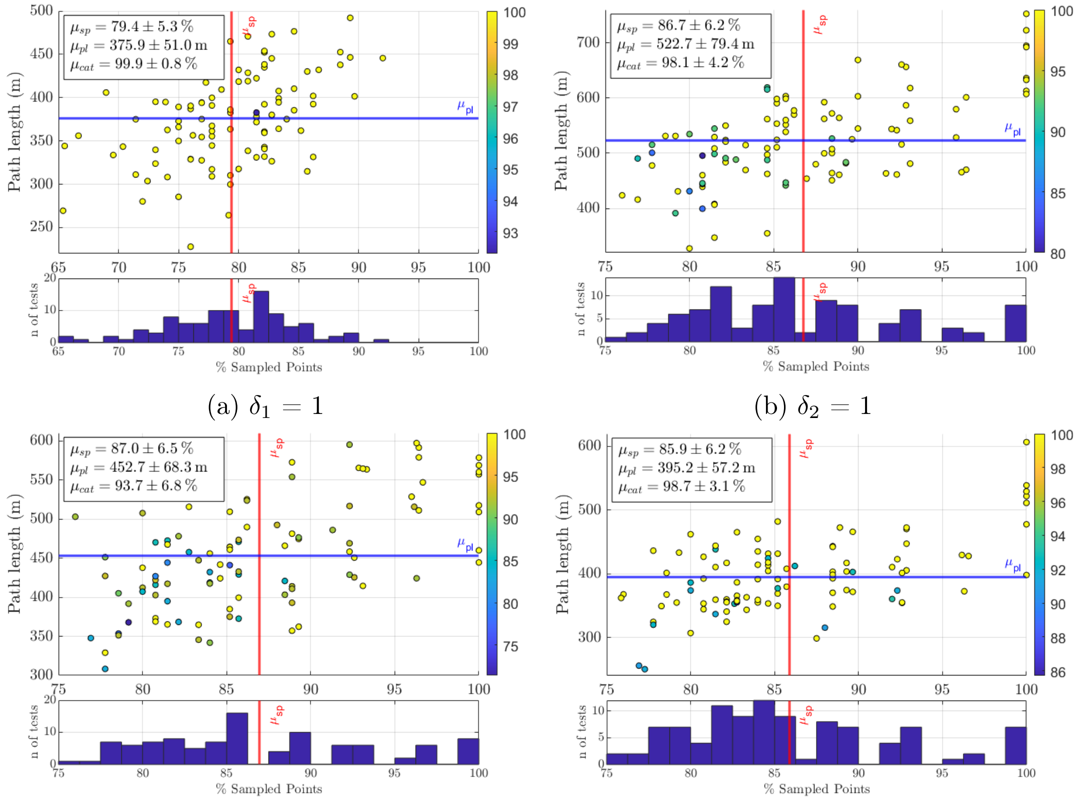


Figure 5.14. Results of the efficiency and quality parameters of 100 simulations performed on the same environment with: 30 sci, 15 cat, 50 obs. 50 simulations were carried out start to end (circular points) and the other 50 were run end to start (triangular points)

Dominance	μ_{sp}	μ_{pl}	μ_{cat}
$\delta_d = 0 \rightarrow$ Absolute dominance	88.2%	481.1 m	97.3%
$\delta_1 = 1 \rightarrow$ Total SVS	79.4%	375.9 m	99.9%
$\delta_2 = 1 \rightarrow$ Total path length	86.7%	522.7 m	98.1%
$\delta_3 = 1 \rightarrow$ N. of visited categories	87.0%	452.7 m	93.7%
$\delta_4 = 1 \rightarrow$ Sum of distances from centre	85.9%	395.2 m	98.7%

Table 5.5. Results of simulations carried out with different concepts of dominance for the different objective functions in randomly generated environments with 30 sci, 15 cat, 50 obs



(a) $\delta_1 = 1$

(b) $\delta_2 = 1$

(c) $\delta_3 = 1$

(d) $\delta_4 = 1$

Figure 5.15. Results of the efficiency and quality parameters of simulations carried out with different concepts of dominance for the different objective functions in randomly generated environments with 30 sci, 15 cat, 50 obs

the average path length, that in the respective scenarios of Figure 5.7a was $\mu_{pl} = 481.1\,\mathrm{m}$ has now reduced to $375.9\,\mathrm{m}$. μ_{sp} is also considerably smaller as it is equal to 79.4%. Moreover, the percentage of visited categories is equal to 100% in almost every simulation.

In Figure 5.15b, path length is of lower relevance and, so, the mean value μ_{pl} turns out to be higher. However, an improvement on the other quality parameters is not actually observed. Only μ_{cat} has increased of about 1%.

Figure 5.15c and Figure 5.15d depict the results in the last two cases.

More studies may be carried out in the future to determine proper values of δ_d to obtain the preferred solution depending on the hierarchy of the different objective functions.

Chapter 6

Conclusions and future developments

6.1 Conclusions

The aim of this work was to study a method capable of performing a multi-objective global and local path planning for a multi-agent system composed by a planetary exploration rover and a helicopter. The efforts have been, then, focused on the definition of the planning system for the ground vehicle only, since it has been decided to employ the aerial vehicle as a scout to map the area of interest before the exploration of the rover.

The multi-objective Pareto Monte Carlo Tree Search designed for global path planning has been proven to be a good system, that is able to manage strongly competing objectives such as maximisation of scientific return of the mission and optimisation of the path. The examination of the parameters defined in the context of the sensitivity analysis has shown that the algorithm generates positive results in terms of efficiency and quality and that, with the use of particular tools such as the non-absolute dominance, the output could be modified according to the preferences. A drawback of the method is related to the not negligible variability of the global trajectory between different runs of the algorithm. On the other hand, the system does not present a marked dependence on the initial conditions.

Once again, the trajectory output is not globally maximised for two main reasons. First of all, the multi-objectivity implies that there can not be a single best path. Indeed, since the objectives compete with each other, the output trajectories will be good in relation to some goals and worst for others and, so, they will be placed on a Pareto front. Secondly, the MCTS method itself is not a global optimiser, but it tries to determine the most promising next move that could be able to maximise the reward value from the point in which the search starts up

to the end.
The local planning problem has been addressed with a step by step artificial potential fields method, which is very fast and light and able to navigate the rover in an only partially known environment. Its problems related to the possible formation of local minima has been solved with the definition of an asymmetric swirling potential field, that is generated with the addition of a tangential vortex field to the Gaussian repulsive one defined for obstacles.
Eventually, it can be stated that the designed system could be an interesting solution for planetary rovers, also because of the low requirements on computational effort and memory. Indeed, both MCTS and artificial potential fields are computationally light methods, which is an important aspect given the low specifications of on-board computers of planetary rovers. Moreover, the system is very fast and could be used for online planning.

6.2 Future developments

Some future developments may be implemented on different aspects of the method both concerning the Monte Carlo Tree Search and the artificial potential fields part. In the following sections, some ideas are presented.

New objective functions

A very interesting improvement to the studied method is connected with the simplicity of adding more objective functions that could consider more aspects of the exploration area.
Indeed, the idea of multi-mode navigation, explained in [30], could be exploited by defining different zones based on the terrain difficulty. A new objective might be to minimise the part of the trajectory that passes in the most hazardous areas in order to reduce risks.
Moreover, the tridimensionality of the map could be introduced by defining an objective function that counts the meters of climb and descent the rover has to make to follow a certain trajectory.

Pareto front evaluation

The Pareto front evaluation with the Upper Confidence Bound is not the only possibility that has been proposed for multi-objective problems. The article in [31] summarises different methods of reinforcement learning used in this field and the authors in [13] apply them to the multi-objective MCTS. However, as was pointed out in [12], this system is computationally prohibitive and could not be used for online planning frameworks.

Nevertheless, in the future it may be interesting to better investigate and improve this option as well, in order to have a less random selection of the most promising child from the Pareto front.

Obstacles with non circular shapes

Finally, another possible development is related to the introduction of non circular obstacles in the map. This represents a challenge in terms of the definition of the repulsive potential fields connected with this objects. Indeed, if a non circular obstacle is to be modelled, it is often not possible to encircle it in a spherical contour without a significant reduction of the workspace.

A possible solution may be to use the Force Inducing an Artificial Repulsion from the Surface (FIRAS) function, as proposed in [32], that can model polygonal shaped obstacles very well in their vicinity. However, it can be responsible for the generation of false local minima far from the obstacle.

In order to avoid this problem, the isopotential contours should assume a circular shape far from the obstacle, while following the obstacle geometry close to it [29]. With this idea, the study in [33] proposes a potential field based on ellipses, which contours are elliptical near the obstacle and circular far from it, thus preventing the formation of local minima at large distances. The drawback of this system is that only obstacles that have one large dominating dimension can be well modelled.

This method has then been improved in [34], where a superquadratic formulation, that generalises the elliptical method to other shapes of objects, is introduced. Therefore, this last solution may be added to this thesis work, even though another option that does not need new concepts could be evaluated. Indeed, a differently shaped obstacle may be modelled as a fusion of smaller circular hazards properly placed on its borders forming a sort of wall. Of course, this would be a less elegant solution, but could be considered as well for its simplicity.

Bibliography

[1] Levin Gerdes et al. “Efficient autonomous navigation for planetary rovers with limited resources”. In: *Journal of Field Robotics* 37.7 (2020), pp. 1153–1170.

[2] Hai Ngoc Pham and Salah Sukkarieh. “A Comprehensive Cooperative Exploration Framework for Ground and Air Vehicles in Unknown Environments”. In: *Proceedings of the 2006 Australasian Conference on Robotics and Automation, Auckland, New Zealand*. Citeseer. 2006.

[3] Bob Balaram et al. “Mars helicopter technology demonstrator”. In: *2018 AIAA Atmospheric Flight Mechanics Conference*. 2018, p. 0023.

[4] Wayne Johnson et al. *Mars Science Helicopter Conceptual Design*. National Aeronautics and Space Administration, Ames Research Center, 2020.

[5] Satyanarayana G Manyam, David W Casbeer, and Kaarthik Sundar. “Path planning for cooperative routing of air-ground vehicles”. In: *2016 American Control Conference (ACC)*. IEEE. 2016, pp. 4630–4635.

[6] Jianqiang Li et al. “A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems”. In: *IEEE Transactions on Vehicular Technology* 65.12 (2016), pp. 9585–9596.

[7] Tauã M Cabreira, Lisane B Brisolara, and Paulo R Ferreira Jr. “Survey on coverage path planning with unmanned aerial vehicles”. In: *Drones* 3.1 (2019), p. 4.

[8] Taua M Cabreira et al. “Energy-aware spiral coverage path planning for uav photogrammetric applications”. In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3662–3668.

[9] Akash Arora et al. “Multi-modal active perception for information gathering in science missions”. In: *Autonomous Robots* 43.7 (2019), pp. 1827–1853.

[10] Amarjeet Singh et al. “Efficient planning of informative paths for multiple robots”. In: *IJCAI*. Vol. 7. 2007, pp. 2204–2211.

[11] Alexandra Meliou et al. “Nonmyopic informative path planning in spatio-temporal models”. In: *AAAI*. Vol. 10. 4. 2007, pp. 16–7.

[12] Weizhe Chen and Lantao Liu. "Pareto Monte Carlo Tree Search for Multi-Objective Informative Planning." In: *Robotics: Science and Systems*. 2019.

[13] Weijia Wang and Michele Sebag. "Multi-objective monte-carlo tree search". In: *Asian conference on machine learning*. PMLR. 2012, pp. 507–522.

[14] Aastha Acharya et al. "Iterative Reward Learning for Robotic Exploration". In: *AIAA Scitech 2020 Forum*. 2020, p. 1377.

[15] William S Noble. "What is a support vector machine?" In: *Nature biotechnology* 24.12 (2006), pp. 1565–1567.

[16] Mark Woods et al. "Autonomous science for an ExoMars Rover–like mission". In: *Journal of Field Robotics* 26.4 (2009), pp. 358–390.

[17] Dave Barnes, Stephen Pugh, and Laurence Tyler. "Autonomous science target identification and acquisition (ASTIA) for planetary exploration". In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2009, pp. 3329–3335.

[18] Marc J Gallant, Alex Ellery, and Joshua A Marshall. "Rover-based autonomous science by probabilistic identification and evaluation". In: *Journal of Intelligent & Robotic Systems* 72.3 (2013), pp. 591–613.

[19] Alberto Candela et al. "Planetary robotic exploration driven by science hypotheses for geologic mapping". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 3811–3818.

[20] Ankit Choudhary. *Introduction to MCTS: The Game-Changing Algorithm behind DeepMind's AlphaGo*. https://www.analyticsvidhya.com/blog/2019/01/monte-carlo-tree-search-introduction-algorithm-deepmind-alphago/. 2019.

[21] Int8. *Monte Carlo Tree Search – beginners guide*. https://int8.io/monte-carlo-tree-search-beginners-guide/. 2018.

[22] Mauro Massari, Giovanni Giardini, and Franco Bernelli-Zazzera. "Autonomous navigation system for planetary exploration rover based on artificial potential fields". In: *Proceedings of Dynamics and Control of Systems and Structures in Space (DCSSS) 6th Conference*. 2004, pp. 153–162.

[23] Jang-Ho Cho et al. "A real-time obstacle avoidance method for autonomous vehicles using an obstacle-dependent Gaussian potential field". In: *Journal of Advanced Transportation* 2018 (2018).

[24] NASA/JPL-Caltech. *Artist's Concept of Rover on Mars*. https://mars.nasa.gov/resources/3904/artists-concept-of-rover-on-mars/.

[25] Giovanni Giardini, Mauro Massari, and Franco Bernelli Zazzera. "Artificial Potential Fields and Digital Elevation Map for a Rover Navigation System". In: *XVIII Congresso nazionale AIDAA*. 2005, pp. 1–12.

[26] Alaa A Ahmed, Turki Y Abdalla, and Ali A Abed. "Path planning of mobile robot by using modified optimized potential field method". In: *International Journal of Computer Applications* 113.4 (2015), pp. 6–10.

[27] Seyyed Mohammad Hosseini Rostami et al. "Obstacle avoidance of mobile robots using modified artificial potential field algorithm". In: *EURASIP Journal on Wireless Communications and Networking* 2019.1 (2019), pp. 1–19.

[28] Rekha Raja and Ashish Dutta. "Path planning in dynamic environment for a rover using Astar and potential field method". In: *2017 18th International Conference on Advanced Robotics (ICAR)*. IEEE. 2017, pp. 578–582.

[29] C De Medio, F Nicolo, and G Oriolo. "Robot motion planning using vortex fields". In: *New Trends in Systems Theory*. Springer, 1991, pp. 237–244.

[30] Róbert Marc, Piotr Weclewski, and Daisy Lachat. "Autonomous multi-mode rover navigation for long-range planetary exploration using orbital and locally perceived data". In: *69th International Astronautical Congress (IAC)*. 2018, pp. 1–5.

[31] Peter Vamplew et al. "Empirical evaluation methods for multiobjective reinforcement learning algorithms". In: *Machine learning* 84.1 (2011), pp. 51–80.

[32] Oussama Khatib. "Real-time obstacle avoidance for manipulators and mobile robots". In: *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.

[33] Richard Volpe and Pradeep Khosla. "Artificial potentials with elliptical isopotential contours for obstacle avoidance". In: *26th IEEE conference on decision and Control*. Vol. 26. IEEE. 1987, pp. 180–185.

[34] Pradeep Khosla and Richard Volpe. "Superquadric artificial potentials for obstacle avoidance and approach". In: *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. IEEE. 1988, pp. 1778–1784.