UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

Departement of Computer Science and Engineering

Hybrid Navigation Control for Multi-Behaviour Robots

by

Fadi Halal

Thesis submitted as partial fulfillment of the requirements for the doctorate degree in Science and Information Technology

July 2017

UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

Département d’informatique et d’ingénierie

Hybrid Navigation Control for Multi-Behaviour Robots

par

Fadi Halal

Thèse présentée comme exigence partielle du programme de doctorat en sciences et technologies de l'information

Juillet 2017

BOARD OF EXAMINERS
THIS THESIS HAS BEEN EVALUATED
BY THE FOLLOWING BOARD OF EXAMINERS

Prof. **Marek Zaremba**, Thesis Supervisor

Département d'informatique et d'ingénierie, Université du Québec en Outaouais

Prof. **Ahmed Lakhssassi**, President of the Board of Examiners

Département d'informatique et d'ingénierie, Université du Québec en Outaouais

Prof. **Jurek Czyzowicz,** Member of the Jury

Département d'informatique et d'ingénierie, Université du Québec en Outaouais

External Evaluator

Prof. **Jurek Sasiadek**, Member of the Jury

Department of Mechanical and Aerospace Engineering, Carleton University

External Evaluator

Prof. **Adam Skorek,** Member of the Jury

Département de génie électrique et génie informatique, Université du Québec à Trois-Rivières (UQTR)

Contents

List of Figures

List of Tables

Acknowledgments

Since I started my PhD study, I have received help, advice and support from many people, to all of whom I am really thankful.

First of all, I am really grateful to my supervisor, Prof. Marek Zaremba, for his help and guidance throughout the preparation of this thesis.

I wish to thank Noetix Research Inc. and the Winnipeg Lake Research Consortium for making available the Lake Winnipeg data, as well as my colleagues from the Canada-China scientific and technological cooperation project "Blue-Green Algae Blooms Warning System" for their support and fruitful discussions.

I am grateful to the members of the dissertation committee for their time and expertise.

Special thanks go to Trudie Winkelaar, David Sims and Eng. Zaid Sweidan who provided editorial correction.

I would like especially to acknowledge my father for his guidance and his insistence that I should work in the field of research.

Last but not least, I would like to express my deep gratitude to my family for their endless love, incredible support, help and understanding.

To Jamil, Eli & Peter

Abstract

This thesis addresses the issue of designing integrated deliberative-reactive architectures for multi-behaviour robot navigation control. The objective of the study is to devise and investigate a methodology for designing robust planning and control systems equipped with a high level of intelligence and capable of navigating a mobile platform, at a high level of performance, in partly unknown environments, where the mobile robot multi-task operation is subject to different behaviours. Of particular interest in this thesis are deliberative-reactive navigation systems operating in large, complex environments, such as those applied in environmental monitoring, that make use of a variety of remote sensing data. The spatial data are interpreted intelligently using multi-layer feature maps.

In this thesis, we present a formal model for hybrid mobile robot navigation. The model integrates two levels of navigation, deliberative and reactive. The novelty in this model is that the decision component makes a decision depending on the global and local context choosing the suitable behaviour, including conflicting behaviours, and regulates the relation between the deliberative and the reactive navigation via computational intelligence techniques. The presented methodology offers a suitable solution for complex partially known environments, where the mobile robot control produces an overall behaviour for executing the proper action in order to reach the target by employing multi-task navigation. In terms of the multi-behaviour operation, the following behaviours are considered: different tasks for environment data acquisition, and different navigation behaviours. In the latter type of behaviours, three situations are studied: dynamic local path, unreachable local path, and conflicting behaviours in a critical situation.

The experiments presented in the thesis demonstrate the utility of the model in the two fundamental types of the navigation: those with the predominance of the deliberative action, and those with dominant reactive action. The experiments adopted the following methodological approach. First, the deliberative navigation was developed using hybrid genetic algorithm to deal with multi-task navigation. We aimed to build a navigation system which has a flexible and efficient performance along global and local paths. A complete solution for monitoring of water quality in Lake Winnipeg using satellite data

was presented. Second, a multi-behaviour deliberative-reactive navigation scheme was designed to deal with conflicting behaviours using artificial intelligence methods, such as fuzzy systems and genetic algorithms, in a hierarchical configuration. The fuzzy control drives the robot to execute the required behaviour, depending on the robot-specific situation and the characteristics of the environment, in order to reach a given target.

Résumé

Cette thèse s'adresse au problème de la conception des architectures délibératives et réactives pour le contrôle de la navigation du robot à multiples comportements. La conception et l'investigation d'une méthode pour une planification robuste ainsi que pour un système de contrôle sont les objectifs de cette étude. En effet, ces robots sont équipés d'un haut niveau d'intelligence capable d'une navigation mobile, permettant ainsi de favorisant le haut niveau de performance dans un environnement partiellement inconnu, en y incluant des opérations multitâches possédant des conflits de comportement. L'intérêt particulier de cette thèse est les systèmes de navigation délibératifs et réactifs opérant dans un milieu complexe, comme le monitorage des environnements qui utilisent plusieurs ressources de données de télédétection. Les cartes multicouches sont utilisées pour la représentation et le traitement des données spatiales.

Cette thèse présente alors un modèle formel pour les navigations des robots mobiles hybrides. Le modèle intègre deux niveaux de navigations soit délibératif et réactif. La nouveauté de ce modèle est que la composante décisionnelle rend la dépendance de la décision dans le contexte global et local. Ceci fait en sorte qu'il est possible de choisir la suite comportementale en y incluant des conflits ainsi que des régulations de la relation entre la navigation délibérative et réactive via la technique d'intelligence de calcul. La présente technologie offre une suite de solution pour les environnements partiellement connus et complexes, où le contrôle de robots mobiles produit un comportement général pour l'exécution de l'action propre dans le bût de trouver le trajet, utilisant la navigation multitâche pour ajuster les comportements au niveau réactif. Dans le dernier type de comportement, on distingue trois situations qui ont été étudiées, le trajet dynamique local, le trajet inaccessible, et le conflit de comportement dans les situations critiques.

Les expériences présentées dans cette thèse démontrent l'utilité de ce modèle dans deux types fondamentaux de la navigation : ceux avec la prédominance de l'action délibérative et ceux avec l'action réactive comme dominante. En effet, les expériences adoptées suivent les approches méthodologiques. Au début, la navigation délibérative est développée en utilisant l'algorithme génétique hybride pour gérer les navigations

multitâches. On est aspiré à construire un système qui a une performance flexible et efficace tout au long de trajet global et local. La solution complète pour le monitorage de la qualité de l'eau du lac Winnipeg qui utilise des données satellites est présentée. Dans un second lieu, un schéma délibératif et réactif hybride est conçu pour gérer les conflits de comportement en utilisant les méthodes de l'intelligence artificielle, comme les systèmes flous et les algorithmes génétiques, dans une configuration hiérarchique. Le contrôle flou dirige le robot à exécuter le comportement requis, dépendamment de la situation spécifique et des caractéristiques de l'environnement, pour atteindre une cible donnée.

Chapter 1

Introduction

1.1. Mobile Robot Navigation

1.1.1. Definition

Navigation is a field of study that focuses on the process of monitoring and controlling the movement of mobile robots (vehicles) from one place to another. An autonomous robot can navigate and perform its task without human intervention.

Navigation is an active process, and it comprises positioning and guidance (Hofmann-Wellenhof et al. 2003). To navigate through the environment, the robot must be able to perceive its surroundings through various kinds of sensors. Moreover, it should answer the following questions:

- Where am I?
- How do I get to other places from here?
- Where are other places relative to me?

(Baltzakis 2004) answered these questions as follows: a mobile robot must be able to (a) understand its environment, (b) localize itself within it, and (c) purposively move to desired target points. Robot position is given as a set of coordinates in a well-defined coordinate reference frame, needing a convention to determine its origin and orientation. While position determination answers the question "where am I?", the planning process is responsible for defining an appropriate trajectory, and it answers the question "where to go and how to go?".

Navigation is defined as the interaction of positioning and guidance. Navigation considers a robot's position with regard to the other relative objects to determine its position to answer the question, "Where are other places relative to me".

Mobile robot navigation is a broad area, which comprises such problems as localization, optimal path planning and mapping. In this thesis, we deal with only a part of it, namely autonomous robot navigation in complex dynamic environments where the robot is subject to different requirements in terms of its behaviours.

1.1.2. Importance

The navigation issue is encountered in many different applications, mainly in all types of terrain exploration and transportation, with the involvement of different kinds of transportation means, such as mobile robots, jet aircrafts, cars and boats. Navigation in the transportation domain usually attempts to find an optimal or sub-optimal path planning for a predefined mission. The optimal navigation problem has been solved in many transportation applications by defining it in terms of the Vehicle Routing Problem (VRP) or the Travelling Salesman Problem (TSP) using a variety of algorithmic and heuristic optimization methods. TSP deals with a list of given cities where the distances between each pair of cities are provided to obtain the shortest possible route that visits each city once and returns to the origin city. VRP generalizes TSP and is considered a combinatorial optimization problem seeking the optimal set of routes for a fleet of vehicles to traverse in order to deliver to a given set of customers.

When executing a navigation task, the risk of any collision has to be eliminated by exploring the neighbourhood of the robot and making a decision to avoid any obstacle where the robot performs its tasks in a partially unknown environment.

Monitoring of environmental phenomena, which is of particular interest in this thesis, requires measuring physical processes, such as nutrient concentration, wind effects, and solar radiation across the entire spatial domain in a range of oceanographic, terrestrial, and atmospheric applications (Tso and Mather, 2001). Given the cost of acquiring a large number of in-situ measurements, traditionally used to gather water pollutant information, remote detection techniques provide significant advantages in terms of spatial and temporal coverage and cost-efficiency. Navigation using environment monitoring has been used to observe and to predict the risk of damage from certain (predefined) phenomena. Environment monitoring often employs remote sensing (RS) techniques that use aircraft or satellites measurements. Its objective is to study a variety of phenomena in different areas, some of which are listed below:

- Water Quality
- Surface Water Hydrology
- Meteorology and Weather Monitoring

Optimal navigation of mobile robotic platforms in dynamic environments is a complex issue subject of both practical and theoretical interest. Navigation requires information about the trajectory of the moving objects in order to deal with dynamic environments.

1.1.3. Global/Local Navigation

In terms of terrain exploration, we distinguish between local and global navigation. Mobile robot navigation is an advanced technique where static, dynamic, known and unknown environments are involved (Yun et al. 2011). In terms of the operational paradigm and the level of control, navigation is divided into two categories, deliberative and reactive navigation, corresponding *grosso modo* to global and local navigation. Deliberative navigation requires a known environment model (world model) and uses the traditional architecture sense-plan-act. Reactive navigation is used for unknown environments (Garcia et al. 2009) and employs the sense & act architecture which has direct connection between the sensors and the actuators, which is known as the behaviour-based architecture (Brooks 1986).

The hybrid deliberative-reactive navigation architecture is suitable for dealing with a partially-unknown environment. This control architecture combines two levels where the deliberative navigation has a higher level and supervises reactive navigation. These hybrid systems are intensively used in complex and dynamic environments.

(Giesbrecht 2004) classifies the local and global path characteristics as in Table 1.

Table 1: Global Path Planning vs. Local Navigation

Global Path Planning	Local Navigation
Uses accumulated and a priori information	Uses immediate sensor data only
Concerned with the global environment including hills, rivers, canyons, forests, roads, buildings, etc.	Concerned with objects and conditions in the vicinity of the robot
Plans for long distances and time periods	Plans for the immediate vicinity for a short time ahead
Slow, deliberative process	Fast and reactive
Allows robot to avoid getting trapped	Allows robot to travel safely
Plans to reach a goal in the most efficient manner	Plans to travel as fast as possible
Simple model of vehicle (point robot)	Complex vehicle model (dynamics and kinematics)

1.2. Sample Acquisition Mission in Large-Scale Environments

1.2.1. Sample Acquisition Mission

Investigations presented in this thesis have been to a large extent motivated by research challenges posed by the problem of designing an optimal navigation solution for large-scale multi-behaviour environment monitoring requiring an effective use of remote sensing technologies (RS). A large–scale environment is usually unstructured and consists of a large number of variables. Depending on the **mission** complexity, first, the robot searches for an optimal global path in the world model by dealing with the large environment entities such as forest, river and lakes. Navigation in such conditions should change the robot's behaviour to deal with different object classes in an adaptive way. Thus, the robot should be able to plan the mission using many different **strategies** performing different **behaviours**. Multi-behaviour navigation missions need a wide range of behaviour to comply with the environment entities and the objective of the missions.

A typical application example is the monitoring of environmental phenomena in large areas, both aquatic and terrestrial, which requires measuring a variety of physical processes, such as nutrient concentration, wind effects, and solar radiation. Acquisition of a large number of in-situ measurements by a mobile sensor platform is a basic task in the process of monitoring biological and chemical pollutants. Remote detection (RS) techniques provide significant advantages in terms of spatial and temporal coverage and cost-efficiency.

Environment maps of large areas are often obtained through processing of multi-spectral satellite imagery. Remote sensing data often have to be augmented and updated by *in situ* measurements due to the need for precise local measurements, for the calibration of satellite imagery in varying water conditions, and for the purpose of precise local decision making.

A path planning system generates an optimal path with the goal of maximizing the number and the value of the collected samples during the acquisition mission. In order to obtain the best and richest data set, an appropriate metric should be defined over the sampling field. Path planning waypoints identify water sample candidates. Thus, many strategies can be applied to water pollutant patches by adopting different search behaviours. An example of the mobile platform used for collecting samples of an aquatic environment is the Namao ship, belonging to the Winnipeg Lake Research Consortium, shown in Fig. 1.1.

Figure 1. 1. Lake Winnipeg Research Vessel, Namao, is the platform used by the Lake Winnipeg Research Consortium to acquire water samples in Lake Winnipeg.

1.2.2. Environment Representation from RS Data

The main source of information on large-scale environments is satellite imagery. Multispectral satellite images are those who obtain information in several bands, as shown in Fig. 1.2. The image in each band consists of a matrix of pixels which contains a numeric value obtained from the sensors when they capture the amount of energy reflected by the objects on the Earth's surface (Morón Hernández et al. 2016). The size of the area covered by one pixel corresponds to the resolution of the satellite image, e.g., 260 m x 290 m for MERIS pixel and 1 km x 1 km for MODIS pixel - see Appendix 1.1 and 1.2.

The combination of bands is achieved under certain rules to obtain new environmental features, such as characteristics of some water pollutant classes that are explained in detail in Chapter 5.

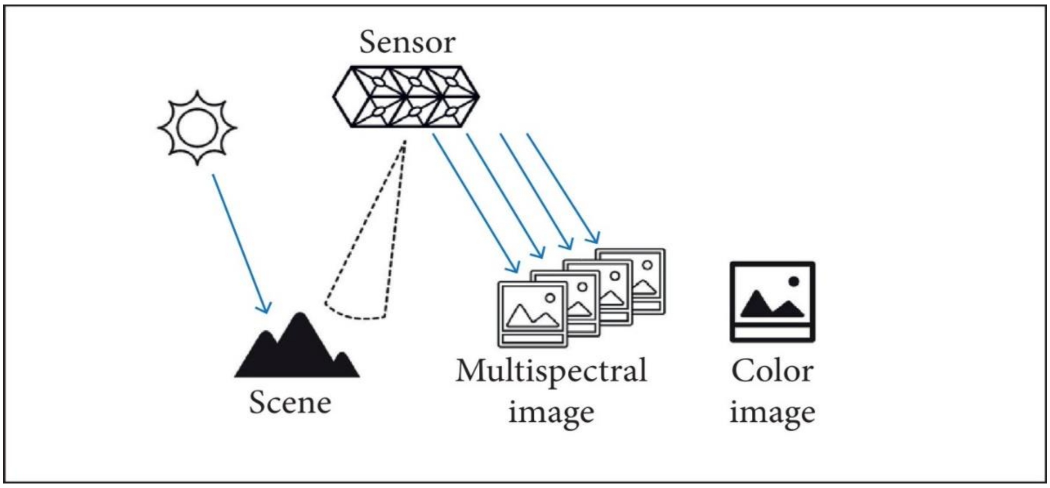


Figure 1. 2. An image obtained from a four-band multispectral sensor

The multi-spectral data can be subsequently processed by using a variety of classification and regression techniques to obtain multi-layer maps which present different classes of features.

Navigation in large-scale environments poses different difficulties for the navigation. In order to guide the search towards the valuable zones, pruning the search zone can be employed to reduce the optimization search time and to obtain a high-quality solution. The pruning zone can be considered as a Region of Interest (ROI) which can be represented in one or more-layer maps.

The data of Lake Winnipeg for navigation is obtained in the form of a multi-layer map that is based on different data sources, apart from satellite images also on ancillary environment information. In order to improve forecasts and make the path planning process more adaptable to changing environment conditions, additional meteorological data are required. As an example of ancillary data are the meteorological measurements which are provided each hour by the national data buoy center. Lake Winnipeg holds three buoys: the 45144 buoy is located in the northern basin, the 45145-buoy located in the corridor zone, and the 45140 buoy is situated in the southern basin. The buoy data include wind speed and direction, the wave height and period, air and water temperature, and other variables. Fig. 1.3 shows the wind speed and direction during August 2012 as measured by the C45144 buoy.

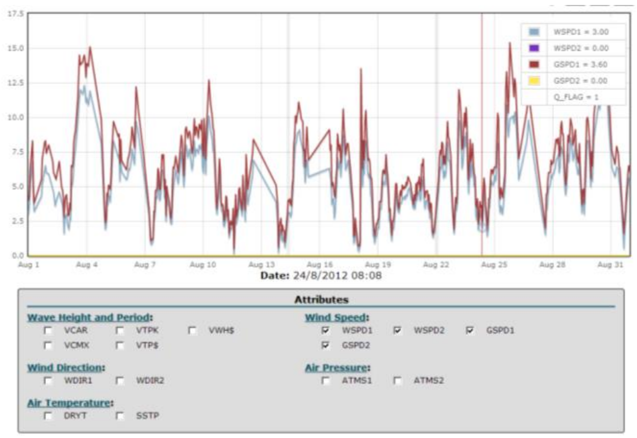


Figure 1. 3. Wind speed and direction data

1.3. Navigation Challenges in Large-Scale Environments

In this thesis, the basic task of a mobile robot is environmental navigation. In general, a mobile robot works in a dynamic and unstructured environment and must also deal with the uncertain and incomplete knowledge of its environment and the effects of its own actions.

1.3.1. Uncertainty in the Information

Navigation in a large-scale environment using remote sensing data faces many uncertainties in the processing of the obtained information. Certain sources of incertitude are as follows:

- Errors in the estimation of water pollutant concentration;
- Top of Atmosphere (TOA) reflectance measurements;
- Motion of moving obstacles;
- Atmospheric scattering and atmospheric absorption;
- Haze and cloud cover.

1.3.2. No Precise Location of Pollutant Patches

Generally, water pollutant types are mixed together, which poses a significant problem for RS spectral reflectance detection.

The application of satellite remote sensing to lake water is constrained by the need for high spatial resolution image data and thus remains limited by spectral resolution capabilities of the satellite imaging sensor (Tyler et al. 2006). Due to the spatial heterogeneity in the water body, the in-situ measurements may not be truly representative of the satellite pixel area. Because of the RS low resolution which reflects a poor representation of the observed area, the water characteristics within each satellite pixel are not truly homogeneous, and thus the in-situ observations are not comparable with the values derived from the satellite image pixels (Moses 2009).

1.3.3. Dynamic Environment

Due to the satellite temporal time coverage, the frequency of environmental changes may be much faster than the frequency of the satellite revisit time. Also, our study zone consists of many significant water pollutant patches which to a large degree are subject to the wind impact. (For more details see section 5.4.6). As a result, the satellite multi-spectral observation represents a partially known environment due to the existence of dynamic changes and objects in the observed zone.

1.3.4. Soft & Hard Obstacles

The problem of obstacles avoidance has to be resolved by considering hard obstacles in the form of islands, coastal areas, ships, and other floating objects, and soft obstacles in the form of haze or cloud patches. Obstacles such as the cloud zones in the satellite image also affect the ship navigation by providing inaccurate prior environmental representation for path planning. Many soft obstacles such as haze and fog can affect both local navigation and global navigation. These obstacles reduce the travel speed and can affect the detection of the target position.

1.4. Multi-Behaviour Operation

In order to navigate in a multiple-class environment, navigation goals should be specified for each class (e.g., the type of pollutant) such that they may be guided by different patterns of behaviour for different purposes. First, some remarks are made regarding the behaviours:

- Mission execution strategy: any mission can be executed by different strategies depending on the robot capabilities. A controller is needed to switch between the strategies regarding the robot situation and the surrounding area conditions.
- Behaviours at the planning phase which considers long-term behaviours executed during the whole local path which deals with the global environment, such as hills, rivers, canyons, forests, roads and buildings. At this level, Behaviour determines the mode of the local paths execution (e.g., if the local path is a steep hill the robot should lower its speed).
- Behaviours at the reactive level are concerned with objects and conditions in the vicinity of the robot that are detectable primarily by on-board sensors. Reactive behaviours are changed and then executed when the new robot onboard sensors' data reading is provided.

In order for a robot to accomplish its goal, the robot control system must employ and perform different behaviours. The priorities of the behaviour may need to be changed with time according to the mission objective. Therefore, a controller (behaviour arbitrator), must select and actualize the suitable behaviour to fit the overall goal of different control objectives (Mai & Janschek 2012). Multi-behaviour robots may have to perform conflicting behaviours to accomplish their tasks, such as making a force closure on the object in order to make the robots able to move the object on the desired trajectory meanwhile avoiding any obstacle to prevent structural damage or damage the obstacle itself (e.g. if the object is a pedestrian). A behaviour selector equipped with high intelligence level may have to be employed to solve such conflicting behaviours.

1.4.1. Behaviours in the Sample Acquisition Problem

Designing a multi-behaviour search system for a mobile sample acquisition platform requires answering the following questions. Which is the suitable navigation mode for a specific water pollutant? How to compute the cost of the solution? How can the solution of the path planning problem deal with multiple patches of high concentration of the pollutant? (Châari et al. 2014).

Critical to the sample acquisition problem is an efficient path planning method, easily adaptable to different control strategies that ensure the collection of data of the greatest value. Acquisition of diverse types of samples may require appropriate behaviours that implement different collection strategies.

Based on the acquisition strategy, water pollutant samples are divided into many classes which are represented as water pollutant patches. The path passes sequentially through a set of patches complying with the mission objective. Some water pollutant patches have a special collections procedure; certain constraints should be considered regarding the sampling in such patches.

Depending on the time and the distance constraints, some pollutant patches can be neglected. Moreover, a decision on the other patches can be done using the following strategies:

- Uniform coverage of high-concentration areas;
- A certain number of samples have to be collected in a specific patch before heading to another patch;
- The sampling behaviour can be different in each patch to comply with the general and local mission goals.

The basic idea of the multi-behaviour sampling navigation is that the acquisition platform explores water pollutant patches using different behavioural characteristics depending on the sampling requirements in each patch.

1.5. Optimization in Path Planning

The path planning procedure designs a trajectory that visits a given set of points such that the optimization process minimizes the total travel distance. The trajectory is represented by a global path which consists of many local paths.

1.5.1. Path Planning Difficulties

Due to the large-scale and complex environment, this task is defined in terms of a combinatorial optimization problem with a globally optimal solution that satisfies all hard and soft constraints. The optimal solution or a set of globally optimal solutions minimizes or maximizes the travel cost (objective function).

First, some definitions are made regarding the navigation components:

- Trajectory: a segment connecting subsequent positions (waypoint) of a vehicle determined by a navigation system.
- Waypoint: a distinct point on the trajectory that usually corresponds to a change of direction.
- Local path: a route between two consecutive waypoints.

The pathfinding problem is typically defined in terms of the Travelling Salesman Problem (TSP) (Ergezer & Leblebicio 2014) or more generally, the Vehicle Routing Problem (VRP)(Cheng et al. 2012). Determining the optimal solution is an NP-hard problem, so the size of problems that can be solved optimally is limited (Châari et al. 2012). In the situation of environment monitoring systems, the problem is even more complex because exact positions of the sampling points are not known a priori.

NP-hard problems require time that is super polynomial in the input size. To solve these problems, many techniques can be applied:

- Approximation: a search process looks for a sub-optimal solution instead of the optimal solution.

- Parameterization: the algorithm gets faster when some input parameters are fixed.
- Heuristic: an algorithm that works "reasonably well" in many cases, but for which there is no proof that it is both always fast and always produces good results.

In practice, solutions to optimal path planning problems frequently incorporate heuristic methods.

1.5.2. Multi-Objective Optimization

The TSP is a special case of the VRP which have proved to be a NP-hard problem. TSP and VRP have different constraints imposed by the user that constitute different TSP/ VRP problems (Min & Dazhi 2014). The problem is to find a path planning that involves solving the sequential ordering problem with precedence constraints. (Yu & Cai 2009) consider a mobile robot exploration mission planning problem is a NP-hard problem where Small-scale mission planning may be solved by the traditional mathematical methods such as exhaustive method and linear programming. (Deng & Hu 2011) determine Route optimization model of multi-modal travel as a NP-hard problem which is difficult to get an optimal solution from by exact algorithms in an acceptable time. Most of the NP-hard problems needing efficient solutions call for the combinatorial optimization approach. Combinatorial optimization consists of finding an optimal object from a finite set of objects. Due to the problem of soft and hard constraints, it becomes Solving Combinatorial Optimization where the solution which calls for a globally optimal solution satisfies all constraints. The optimal solution or the set of globally optimal solutions minimize or maximize the objective function (Rajappa 2012). In general, each optimization problem to be solved requires a unique objective function that represents a performance criterion used in the evaluation of the performance of all individuals in a population. Many functions, such as travelling distance, time window and the sample values (weight) should be optimized simultaneously. This may involve a combination of maximization and minimization criteria (Coello Coello 2006). Individual objective functions are usually combined into a single composite function by weighting the objectives with a weight vector. The result of the optimization should reach a reasonable solution that satisfies the multiple objectives. For mission planning of an unmanned aerial vehicle (UAV), (Vachtsevanos et al. 2005) used the distance, the hazard, and the maneuvering of the route as components of their cost function. Each component had a weight factor which was assigned according to the objectives of the mission. The hazard is related to the existence of obstacles near the path, and the maneuvering refers to the maneuvers required to perform target tracking. To effectively determine and search the best flight (UAV) routes an objective function was created by (Sun et al. 2011) which involves the timeliness and the smoothness of the path. The work of (García et al. 2011) used an objective function that included several components: the cost of the motion from the start node to the current node, the heuristically estimated value of getting from the current node to the goal, the terrain traversability component, the direction change cost, and the cost of navigating in shadow areas. Each component has a corresponding coefficient factor used to weight it according to its importance in the mission.

Generally, the search and optimization problem involve many constraints which the optimal solution must respect and satisfy. Optimization is usually a nonlinear problem (Deb 2000). The approach that has proved effective in solving combinatorial optimization problems is the use of genetic algorithm or hybrid metaheuristic methods. (Braünl 2006) proposed a GA system for solving tasks that are difficult to solve such as NP-hard problems in a large solution search space.

1.5.3. Metaheuristics

Metaheuristics are high-level strategies designed to find, generate, or select a heuristic search algorithm that may provide a sufficiently good solution to an optimization problem. Efficient metaheuristics employ intensification and diversification in the search for good solutions. The term diversification generally refers to the exploration of the search space, whereas the term intensification refers to the exploitation of the accumulated search experience. Exploitation improves the local search by examining neighbours of elite solutions, while exploration uses the global search to examine unvisited regions and generates solutions that avoid getting stuck in a local optimum. Thus, efficient metaheuristics should employ a dynamic balance between diversification and intensification to control efficiently global and local searches. The goal is to quickly identify regions in the search space with high-quality solutions, but also not to waste too much time in regions of the search space which are either already explored or which do not provide high-quality solutions (Blum & Roli 2003)

Metaheuristics have been applied to a wide set of different problems using many techniques to conduct the search. Metaheuristics can be represented as a general algorithmic framework with few modifications and can be applied to different optimization problems.

Summarizing, we outline fundamental properties which characterize metaheuristics:

- Metaheuristics can adapt strategies which guide the search process.
- Metaheuristics can efficiently explore the search space in order to find nearly optimal solutions.
- Techniques which constitute metaheuristic algorithms range from simple local search procedures to complex learning processes.
- Metaheuristic algorithms are approximate and usually non-deterministic.
- Metaheuristics incorporate many mechanisms to avoid getting trapped in confined areas of the search space.
- The basic concepts of metaheuristics permit an abstract level of description.
- Metaheuristics are not problem-specific.
- Metaheuristics may make use of domain-specific knowledge in the form of heuristics that are controlled by an upper-level strategy.
- Metaheuristics employ the search experience to guide the search embodied in some form of memory

1.5.4. Classifications of Metaheuristics

There are a wide variety of metaheuristics, and there are a number of properties with which to classify them. Figure 1.4 illustrates the metaheuristic categories.

Metaheuristic methods are divided into two categories in terms of the search strategy:

1. Global search metaheuristic which includes
    - Genetic algorithms;
    - Ant colony optimization;
    - Evolutionary computation;
    - Particle swarm optimization.

2. Local search metaheuristic;
    - Simulated annealing;
    - Tabu search;
    - Iterated local search;
    - Variable neighbourhood search.

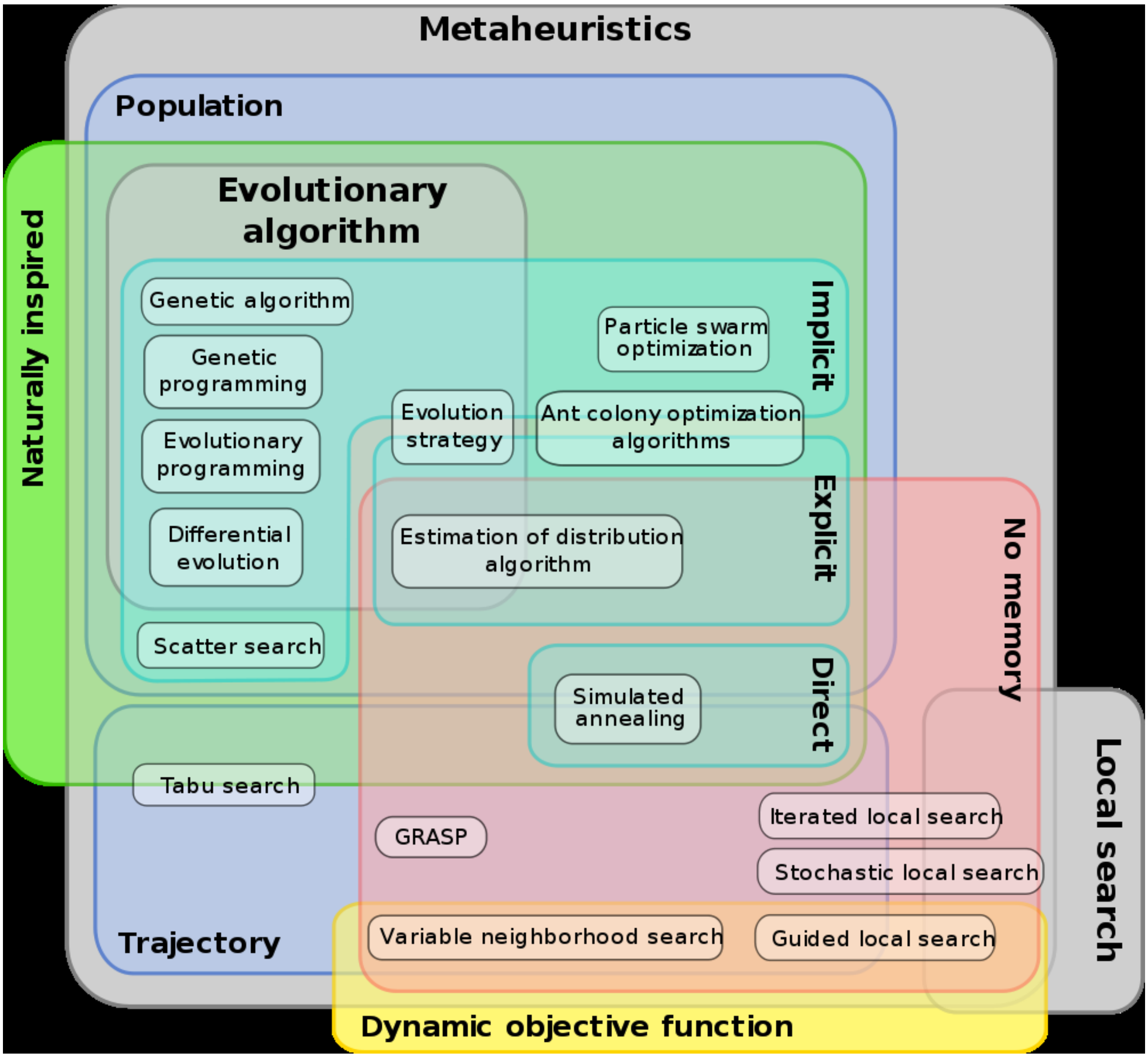


Figure 1. 4. Metaheuristics classification

Other metaheuristic classification concerns the evaluation of the solution. From this perspective, metaheuristic methods are divided into two categories as follows:

- Population-based methods
- Trajectory-based methods

Population-based approaches are based on multiple candidate solutions which are maintained and improved. Thus, the population characteristics guide the search; population-based approaches perform search processes which describe the evolution of a set of points in the search space. The following population-based metaheuristics are examples of this category:

- Evolutionary computation;
- Genetic algorithms;
- Particle swarm optimization;
- Swarm intelligence;
- Ant colony optimization;
- Particle swarm optimization;
- Social cognitive optimization;
- Artificial bee colony.

Trajectory-based approach describes a trajectory in the search space during the search process which is modified and improved. Trajectory metaheuristics include

- Tabu search;
- Simulated annealing;
- Iterated local search;
- Variable neighbourhood search;
- Guided local search.

Population-based methods are better in identifying promising areas in the search space. Meanwhile, trajectory-based methods are better in exploring promising areas in the search space.

1.6. Hybrid Intelligent Techniques for Navigation

Generally speaking, individual intelligent techniques have their limitations, which justify the development and use of intelligent hybrid systems. (Abraham 2003) introduces different generic architectures for integrating intelligent systems. Fig. 1.5 illustrates hybrid systems divided into four categories as below:

- Neuro-fuzzy system (ANN-FIS)
- Evolution –fuzzy system (EC-FIS)
- Evolution –neuro system (EC-ANN)
- Evolution -neuro-fuzzy system (EC-ANN-FIS)

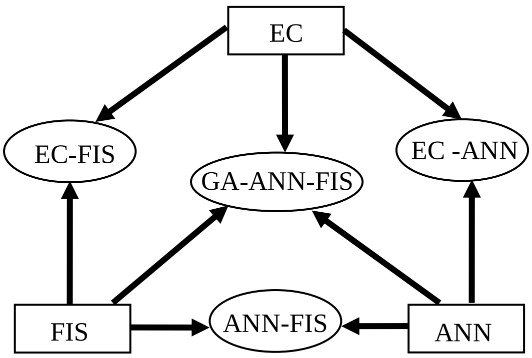


Figure 1. 5. General framework for hybrid soft computing architectures (Abraham 2003)

Geno-fuzzy system has been widely used to overcome the limitation in both systems given the rich knowledge base of fuzzy systems and the innovating power of genetic systems. The integration system requires a sophisticated design to optimally exploit the power of these techniques (Dongbing Gu et al. 2003) (Gu et al. 2003) (Karray & De Silva 2004) (Senthilkumar & Bharadwaj 2009) (Nolfi & Floreano 2000).

1.6.1. Genetic Algorithms for Deliberative Navigation

The concept of genetic algorithms was introduced by John Holland in the early 1970's (Holland, 1975). Genetic algorithms (GA) are based on a solid theoretical foundation of the Schema Theorem (Goldberg, 1989).

Genetic algorithms are adaptive methods that can be used to solve search and optimization problems involving large search spaces. As is the case in nature, solutions are processed by many operators, such as crossover and mutation. In each generation, the best solution survives and has more probability to reproduce new population in the next generation. Selection and Termination operator maintain and manipulate generations. A fitness function is used to simulate the evolution process.

Genetic algorithms (GA) have frequently been used in NP-hard problems, due to their flexibility and high quality of the search results (Samadi & Othman 2013). They can solve the problem without any advance knowledge about the environment, and are largely unconstrained by the limitations of the classical search methods (Rothlauf 2006). By mimicking natural evolution processes, they have the ability to adaptively search large spaces in near-optimal ways. In practical terms, GA methods are easy to interface with exciting simulation models. An important feature that should be considered in implementing GA techniques is that they are problem specific.

Genetic algorithm starts with an initial set of the population which can be generated either randomly or heuristically. This set presents an initial set of solutions for a given problem. A fitness function can then evaluate the generation in terms of solution quality. To obtain a new generation, many operators are applied, such as selection, crossover, and mutation. This procedure leads to solutions of increasing quality over the course of many generations. A termination criterion is applied to check the solutions' convergence. The search process is terminated, and the final solution is shown as output.

The power of the genetic algorithm can be summarized as follows:

- The chromosomes most successful in each generation will produce more offspring than the chromosomes that perform poorly.

- Two good parents will sometimes produce offspring that are better than either parent.
- Each successive generation will become more suited to their environment.

Some of the advantages of a GAs

- GAs can be used without a deep mathematical background;
- GAs can solve hard problems: a GA is one of the best ways to solve a problem about which little is known.
- GAs are easy to extend.
- GAs are easy to interface with exciting simulations and models.
- They simultaneously search from a wide sampling of the cost surface,
- GAs are able to deal with a large number of variables (Haupt & Haupt 2004).

GA disadvantages

Even though GAs can rapidly locate good solutions, they have some disadvantages for difficult search spaces (Binitha & Sathya 2012):

- GAs may tend to converge towards local optima rather than the global optimum of the problem if the fitness function is not defined properly.
- Operating on a dynamic data set is difficult.
- GAs are not directly suitable for solving constraint optimization problems.

Table 2 summarizes advantages and shortcoming of GAs

Table 2: Advantage and Shortcoming of GAs

Advantage	**shortcoming**
More general purpose than traditional optimization algorithms	Fitness function and search techniques often not obvious
Ability to solve "difficult" problems	Premature convergence
Solution availability	Computationally intensive
Robustness	Difficult parameter optimization
Inherent parallelism	

1.6.2. Ant Colony Optimization for Deliberative Search

Ant Colony Optimization is considered as a metaheuristic algorithm and is popularly used for the solution of complex optimization problems (Liu & You 2009) (Min & Dazhi 2014).

Generally speaking, in the standard ACO problem ants traverse arcs that connect nodes in a graph that comply with the objective of the path planning, such as

- Visiting all nodes (or most of them)
- Minimizing the total cost of the trip in terms of distance or time.

This system can be used to find a path for a mobile platform that visits several nodes in many patches to collect samples. During the mission, the ants have one of two roles, either as a collector or as an explorer.

While in the explorer role, an ant must select the next node in its route each time. It will check if the corresponding arc already exists and will use its pheromone and heuristic information for the calculations or if not, it will create a new node.

If an ant is in the collector role, it will use the pheromone and heuristic information stored in the grids to decide on its next move.

The collector ant will take small steps (limited by $\Delta coll$) in order to visit near points and collect samples; the collecting process will continue until the product of the collected samples (or visited nodes) and a "load" parameter is over a certain random number. When this goes over the threshold, the ant will change its role from the collector to the explorer and $\Delta expl$ will be used as the maximum step size. Table 3 shows ACO advantages and shortcomings.

Table 3: ACO Advantages and Shortcomings

Advantages	**Disadvantages**
Inherent parallelism	Theoretical analysis is difficult
Strong local search capabilities	Can sink in local optimum
Efficient for dynamic applications	Random behaviour
Convergence to solution is guaranteed	Time to convergence is uncertain

The main idea is that a set of robots, called *ants*, search in parallel for good solutions to TSPs/ VRPs. Each ant builds one solution in every loop depending on the information from the previous experience and on a greedy heuristic. All robots cooperate through pheromone-mediated indirect and global communication.

1.6.3. Geno-Fuzzy System for Mobile Robot Navigation

Genetic algorithms are applied for their learning abilities and for generation and optimization of fuzzy rules base which improve and modify the fuzzy system behaviour. Table 4 shows the advantages and the disadvantages for fuzzy systems and genetic algorithms (Karray & De Silva 2004).

Table 4: Advantages and Drawbacks of Fuzzy Logic and GAs

Properties	**Fuzzy systems**	**Genetic algorithm**
Store knowledge	Explicit	None
Learns	No	Ability to learn
Optimizes	None	Powerful
Handle nonlinearity	Yes	Yes

The geno-fuzzy system improves the performance of the fuzzy control system by controlling the fuzzy rules base structure (Halal & Dumitrache 2007). Geno-fuzzy system architecture consists of a mobile robot, a fuzzy control system, an evolution strategy unit that adapts the fuzzy membership function and the rules base, and a simulation or real environment to evaluate the quality of robot behaviours which are represented by the encoded chromosome. Fig. 1.6 shows geno-fuzzy system in a mobile robot. The evolution strategy is responsible for generating the fuzzy parameters that are needed to optimize the fuzzy control, which depends on a strategy defined by a human.

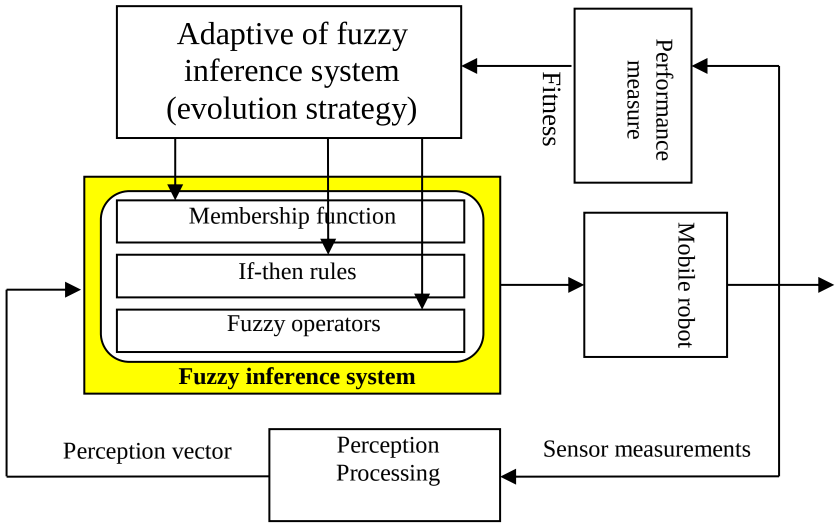


Figure 1. 6. General Geno-fuzzy system architecture for mobile robot

Fuzzy system membership functions and rules base are interdependent (Karray & De Silva 2004), and so by optimizing one of them, we may achieve a sub-optimal solution. The optimization of a fuzzy system is divided into two steps:

- Generating an optimal membership function set.
- Teaching fuzzy rules base that improves the robot's performance (Senthilkumar & Bharadwaj 2009)

1.7. Objectives

The principal objective of this thesis is the development of an integrated, heuristics-based intelligent methodology for designing hybrid deliberative-reactive navigation systems for complex and dynamic environments, where the mobile robot executes multi-task navigation subject to different behaviours.

We are dealing with large-scale and multiple entity environments. Both levels of navigation (the planning and the control system) should have the ability to successfully control the multi-behaviour strategies for a large range of tasks, including conflicting tasks. This system can navigate the robot in unstructured and dynamic environments by exhibiting high levels of intelligence in robot performance.

The aim is efficient path planning which applies different search strategies in complex conditions and a reactive control system that can comply with a global path sequential ordering of tasks in complex and critical conditions by applying multi-behaviour navigation.

One of the goals is to provide a formal architectural framework comprising principal components of the hybrid navigation system and the transitions between those components.

To reach our objectives, the following sub-objectives should be accomplished:

- Development of a comprehensive model of environmental representation which consists of multiple dimensional data and multiple scales using a variety of methods to properly interpret the global environment with respect to the mission objective.

- Employment of multi-layer maps to generate spatial and functional properties of the environment. These maps enable the planning system to perceive and interpret the global environment according to different environment features.

- Investigation of computational intelligence techniques to classify the multi-variable in order to extract proper entities (Regions of interest, obstacles, etc.) in the global context.

- Development of the deliberative level to produce efficient and flexible path planning dealing with the different objects in an adaptive way using different strategies. In order to deal with a dynamic and unstructured environment, the system should be able to change the search method as well as the search cost function complying with multi-behaviour navigation in the multi-dimension world model.

- Development of an integrated hybrid navigation control architecture, which seamlessly combines the deliberative path planning level with local reactive control to create an optimal global path and to also generate a trajectory dependent on context, selected global strategies and local multi-behaviours.

- Development of a hybrid metaheuristic method in order to deal with a dynamic environment as well as applying multi-behaviour navigation to get the optimal solutions that provides good performance.

- Development of an intelligent integrated system to deal with multi-reactive behaviour navigation in a dynamic and complex environment by performing different behaviours including conflicting behaviours in critical situations.

1.8. Structure of the thesis

This Ph.D. thesis is organized as follows:

- **Chapter 2. Multi Behaviour Navigation**.

In this chapter, the Global path section shows the definition, architecture and characteristic of the global path. Multi-objective optimization is discussed. Local path section studies 3 cases of local path: static local path, dynamic local path and unreachable local path. Navigation architectures are presented that show top down, bottom up and hybrid navigation architecture. Behaviour arbitrators are discussed. Behaviour and conflicting behaviour are explained showing the behaviour selection algorithm. In the last section, Multi-behaviour navigation state of the art is described including the progress done in this area. Many hybrid system architectures are explained illustrating general frameworks of multi-behaviour navigation.

- **Chapter 3. Thesis objectives & contribution**

Chapter 3 explains the thesis objectives focusing on multi-behaviour navigation. It also discusses thesis contributions, such as environmental representation, multi-behaviour deliberative search, multi-behaviour reactive navigation and a novel integrated navigation architecture.

- **Chapter 4. Integrated deliberative-reactive multi-behaviour navigation**

In this chapter, a novel architecture is proposed to integrate the hybrid multi-behaviour navigation. Environmental representation can be obtained from the global data which provides multi-layer maps depending on the mission strategies. A world model uses deferent sources of data to interpret the study area. A context module is responsible for providing all the navigation requirements to interpret the local and global context such as the region of interest from different maps regarding the multi-behaviour navigation goal. Navigation behaviour control module is responsible for regulating the two levels of navigation and enhancing the quality of the navigation in a complex and unstructured environment. Many models are developed using hybrids of metaheuristic methodology to improve the multi-behaviour navigation at the deliberative level. Three different local path cases: static, dynamic and unreachable local path and activating conflicting behaviour are discussed employing the integrated architecture.

- **Chapter 5. Deliberative multi behaviour navigation**

This chapter presents the development and testing of hybrid multi-behaviour navigation at the deliberative level. The study zone is obtained through the interpretation of multi-dimensional representations of the environment by different classification methods. A multi-behaviour system is presented that uses hybrid genetic algorithms to supervise the global search for an optimal global path. Multi-behaviour navigation is applied with regard to regions of interest (ROI) where each ROI is associated with a different local search behaviour related to a different sample collection strategy. A sample acquisition mission is optimized through the generation of a sequence of waypoints covering the required ROIs. Different cost functions are investigated to comply with different local and temporal constraints. This chapter presents four experiments showing the efficiency of our method.

- **Chapter 6. Hybrid deliberative-reactive navigation**

Deliberative reactive navigation is presented using a Khepera mobile robot in Kiks simulator environment. A design of a hybrid deliberative reactive navigation system is discussed that is able to perform rapid decision making in a complex and dynamic environment. Hierarchical fuzzy-based systems are employed to implement the multi-behaviour navigation using two strategies: avoidance behaviour and aggressive behaviour. Three tests are performed studying the dynamic and unreachable local path and as well as the effect of the aggressive behaviour in a specific situation.

- **Chapter 7. Conclusions**

In this chapter, we summarize the main contributions of the thesis, and present our accomplishments in the following problems: deliberative multi-behaviour navigation, hybrid multi-behaviour deliberative-reactive navigation and integrated deliberative-reactive multi-behaviour navigation architecture.

Chapter 2

Multi-Behaviour Navigation

2.1 Navigation Levels

The two main navigation approaches are the deliberative and reactive navigation. Each deals with different aspect of the environment to achieve the mission target. Deliberative and reactive navigation correspond, in general terms, to global and local navigation respectively. The deliberative navigation generally operates at a higher level and supervises the reactive level (Chen et al. 2008)(Chen & Cheng 2008)(Chunlin Chen et al. 2008). Deliberative navigation is responsible for planning routes and actions toward a given goal. The deliberative layer is designed as a discrete process providing regular input to the reactive layer (Powers & Balch 2009).

In the first step, the robot searches for a global path in the world model. Usually, global path plans a mission for long distances and time periods. The path planning complies with the mission objective to get to the target point. The path planning plans the mission into a sequence of local paths. In the second step, as soon as the robot determines the global path, the robot starts executing the local paths using its onboard sensors. The local path is concerned with objects and conditions in the vicinity of the robot preventing the robot from any structural damage (collision) and allowing the robot to travel safely.

2.1.1 Global Navigation

Path planning uses the world model and the ancillary information for establishing an action plan. At this level, the robot plans the necessary actions to meet its objective. This action plan is based on the robot's perception of the external world.

Path planning aims at determining a global path from a start to a goal position depending on a specific strategy. Many strategy types involve the computation of a collision-free path, and many other strategy types concern the mission time, the travelling distance allowing the robot to approach, move, and handle the nearby obstacles resulting in a reduction in the cost of the path. These two strategy types are called conflicting navigation strategies (Antonelo et al. 2008).

Path planning has two different classes of algorithms: Non-adaptive algorithms and Adaptive algorithms.

Non-adaptive algorithms plan the global paths depending on the world model, which consists of the predicated information about the global environment before any local observations are made;

Adaptive algorithms update and replan the sub-global paths whenever new information is provided refining the path planning process making it adaptable to changing environment conditions (Halal & Zaremba 2010) (Singh et al. 2009).

Velocity planning requires the consideration of the robot platform dynamics and actuator constraints (Jenne 2010). The new path might be longer and have new strategies to avoid the hazard locations.

2.1.2 Path Planning Components

The common path planning components and characteristics are as follows:

- Search Space: the robot environment where the robot has to perform its task which means the world model and the possible waypoints to visit (transit state). The robot is represented as (x,y) coordinates in the Euclidean space; many conditions can be added such as the fuel level.
- Actions (behaviour): A planner must generate the suitable behaviour which controls the robot when moving from state to state.
- Initial and Goal States: The planner should design the way for the robot in order to get from the initial to the goal state.
- Path planning (path): a path from the start configuration to the goal configuration.

- Time: the time needed for a robot to perform the path usually should be the least amount of time possible. The time can be expressed in this way: the robot should be at the point (x, y), at the time (t). Time may also be represented simply as a sequence of actions: “after Action A is completed the robot will do Action B”.
- Criteria: many criteria can be introduced to optimize the path such as time, distance, or safety which should be optimized simultaneously. This may involve a combination of maximization and minimization criteria (Coello Coello 2006).
- Constraints: optimal path planning should satisfy a number of constraints (Deb 2000), such as the maximum travel distance, the maximum mission time and maintaining the robot safety, on the other hand, the path planning should respect the physical limitations of the robot.
- Algorithm: This is the method by which the best plan is obtained given the criteria and constraints for planning. Section 4.10 will discuss the heuristic algorithms in depth.

2.1.3 Local Path

Local navigation strategy employs the reactive behaviours of the mobile robot, so that the latter can avoid unforeseen obstacles which appear on the global path specified by the planner, so that the mobile robot can safely move between its current position and the next intermediary position on the global path.

Local path navigation is divided into three categories which are as follows:

- Static Free local path
- Dynamic local path
- Unreachable local path

- Static Free obstacle Local path execution: the global path planning represents any local path as a straight line between two nodes (waypoints) which is the shortest local path (Sedighi et al. 2004). As a result, all of the locations on that path are free, and the robot navigates along this path applying destination seeking behaviour. Therefore, the trajectory of the local path will be very close to the local path as in the global path but not identical due to the robot kinematic and dynamic constraints.

- Dynamic local path: in a dynamic environment, any two nodes (waypoints) of a local path are reachable if there is a trajectory from the start waypoint to the target waypoint such that all of the locations on this trajectory are free. The dynamic local path changes its behaviour when an obstacle is detected. Obstacle Avoidance behaviour is applied to prevent the robot from any collision producing a new local path trajectory. This trajectory doesn't match with the predefined local path which has too many curves for the robot to reach its target safely (Belkhouche 2009).

- Unreachable local path: the local path target cannot be reached due to the unforeseen obstacles such as dead-end zone or u shape obstacle which prevents the robot from executing its local path. In this case, many scenarios can be applied depending on real-time information provided by reactive sensors and the robot multi-behaviour ability to perform. In the first scenario, the robot replans the sub-global path. The deliberative navigation should intervene to replan the sub-global path. A new sub-optimal path is generated giving a new task to the robot complying with the environment changes and the mission objective. During the replanning of the sub-global path, the robot continues its attempts to reach a previously recorded local-goal that is now unreachable. In the second scenario, the robot activates the conflicting behaviour to approach and move the nearby obstacle (see section 6.4.4.).

2.2 Navigation Architectures

Navigation control can be implemented by systems of different architecture. In broad terms, navigation architectures can be classified into two categories: top-down and bottom-up, corresponding in general to deliberative and reactive navigation. Figure 2.1 illustrates a typical top-down architecture approach using abstraction to decompose the perception, reasoning, and execution cycles (Xiao-Wen Terry Liu 2005).

This architecture has multiple levels of abstraction in the perception, reasoning, and execution stage; it has an explicit world model which focuses on generating one control strategy and carrying it through to the end.

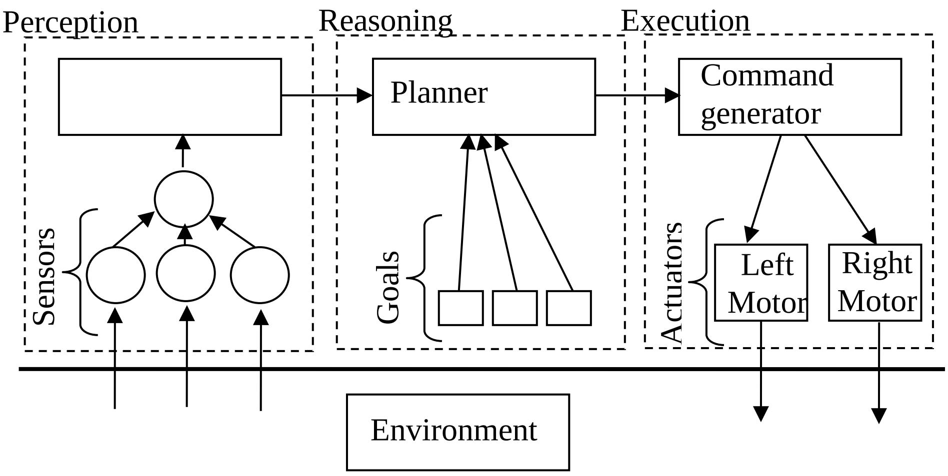


Figure 2. 1. Generic top-down architecture.

These systems are good at planning and higher-level reasoning, but are not reactive enough for dynamic environments.

The bottom-up architecture includes simple behaviours that map perceptions directly to actuator commands (Xiao-Wen Terry Liu 2005) as is shown in Fig. 2.2, where reflex is considered as a behaviour.

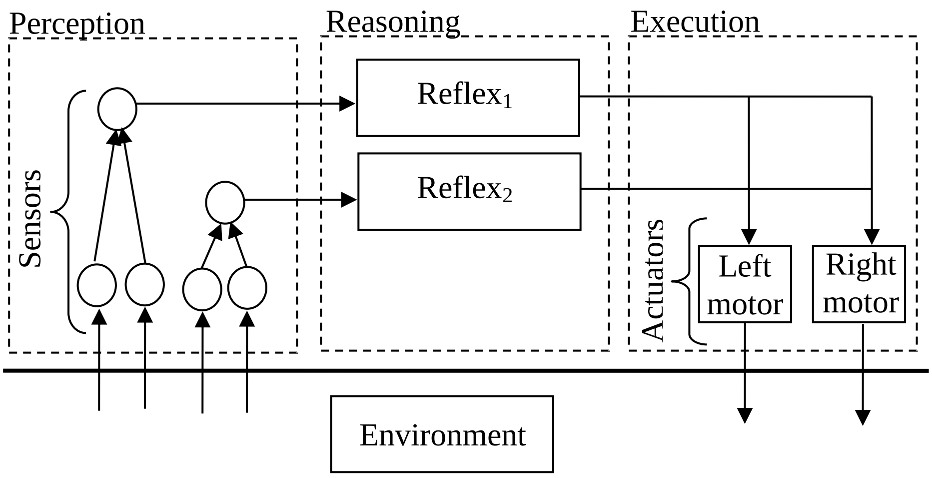


Figure 2. 2. Generic bottom-up architecture

This architecture has good reacting capabilities because of the direct links between the sensors and actuators. This architecture contains three modules: perception, reasoning and execution stage.

2.2.1. Hybrid Architectures

Hybrid architectures are intensively used because they are a mix between top-down and bottom-up architectures combining their advantages. These systems inherit the advantages and the weaknesses of both types of the systems. The hybrid architectures also have three stages: perception, reasoning and execution (Xiao-Wen Terry Liu 2005). In the perception module, there are many sensors which are directly connected to the actuators. So, there is no complete world model, and others sensors are processed more extensively as is shown in Fig. 2.3. Thus, the environment can be mapped to certain states in the system, and the reasoning module moves the system to the desired state. This hybrid architecture may be used in complex behaviour systems.

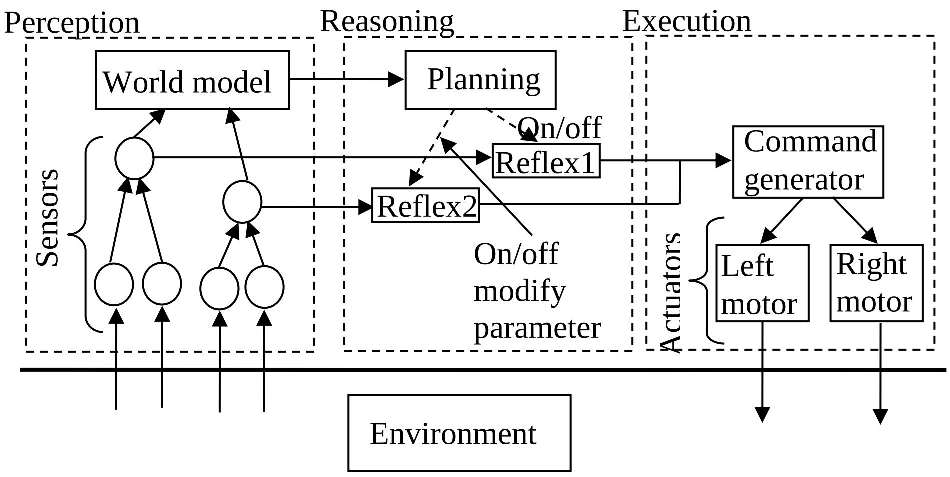


Figure 2. 3. Hybrid architectures

2.2.2. Hybrid Architecture Characteristic

Hybrid navigation can solve the planning problem in dynamic environments (Baltzakis 2004). The deliberative planning systems and purely reactive control systems have their limitations. For instance, deliberative control architecture is not useful in avoidance of unexpected and dynamic obstacles (due to its slowness). Furthermore, in reactive control architectures, the robot has the risk of falling into a dead-end zone or locking into a local minimum. As well as the time to perform the task can be very long (Hank & Haddad 2014). Hybrid deliberative-reactive navigation architecture, which combines reactive and deliberate navigation, has the advantages of both worlds (Rekik et al. 2009) (Lee-Johnson & Carnegie 2010). A pure reactive control system is responsive, flexible and robust while the deliberative planning system has slow responsiveness and abstract representational knowledge. Then hybrid deliberative-reactive robotic architecture can combine the above characteristics. Control system architectures are analyzed by (Nakhaeinia et al. 2011). Table 5 compares their characteristics.

Table 5: Analysis of the control system architectures.

Architecture Specification	Deliberative	Reactive	Hybrid
Goal oriented	Very good	Not good	Good
Flexibility	Very bad	Very good	Very good
Ease of application	Very bad	Very good	Good
Reactivity	Very bad	Very good	Good
Optimal operation	Very good	Very bad	Good
Task learning	Very good	Moderate	Moderate
Robustness	Not good	Good	Very good
Planning	Very good	Not good	Good
Efficiency	Not good	Very good	Very good

Figure 2.4 illustrates hybrid mobile robot architecture, combining the planning unit of a deliberative control and the behaviour strategies unit of a reactive control. The deliberative control has the highest level consisting of the planning block that creates the global path from the start point to the target point. The global path is divided into many local paths to send them to reactive control as sequential tasks. The deliberative level receives a feed-back signal from the reactive level to update the path when the robot couldn’t perform one of its local tasks, when the robot faces an obstacle in the dynamic environment and when the robot receives information about the surrounding changes. In these situations, the deliberative robot generates a sub-global path from the robot’s position where it is stuck to the target (unreachable local path).

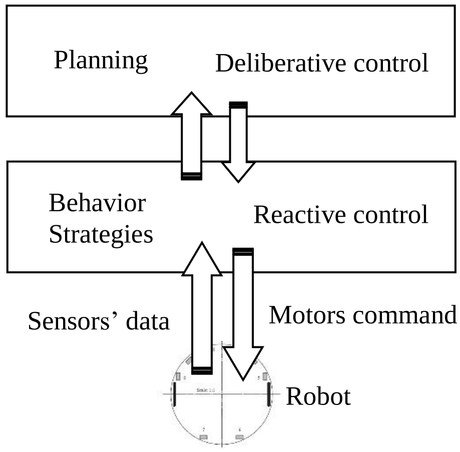


Figure 2. 4. Hybrid robot architecture

The reactive layer is at the lowest level executing the local paths. The robot performs these tasks depending on many behaviour strategies. This control has direct connections between the sensors and the actuators which provide the information about the surrounding environment. This information helps the robot to execute its tasks and to observe the changes in the surrounding area.

(Giesbrecht 2004) describes the relation between the two navigation processes as a complementary relation, meanwhile (Ruiz et al. 2016) proposes a fuzzy system serving as an arbitrator to resolve the conflicts between deliberative planning and reactive control. The fuzzy system selects which navigation should apply depending on two input variables, the validity of the plan and security of the robot; the first variable is provided by the deliberative layer and the second variable is obtained from the reactive layer.

Fuzzy rules which control the navigation layers are as follows:

- If Plan is Valid and Safety is High, Then Navigation is Deliberative.
- If Plan is Valid and Safety is Low, Then Navigation is Reactive.
- If Plan is Invalid, Then Navigation is Reactive.

Depending on these rules the fuzzy system generates a resulting command which will indicate the level of deliberativeness or reactivity (Ruiz et al. 2016).

2.3 Behaviour and Conflicting Behaviour

Stimulus-response is the most intuitive method of expressing behaviours, and any behaviour can be represented as a generated response to a given stimulus (Brooks 1986) (Mali 2002).

Generally, behaviour definition is based on IF-THEN rules. Each rule determines the conditions that relate to a specific behaviour. The behaviour is expressed in the form of IF condition THEN action. At the reactive level, the behaviour is executed once the stimulus is true, where preconditions are set up. In order to change the action regarding the mission strategy, an operator will activate the behaviour respecting the goal that the planner is trying to achieve.

The objective of mission navigation can be decomposed into simple objectives like obstacle avoidance, and goal seeking. These simple objectives are going to be the basic behaviours for the mobile robot (Amin et al. 2005). A behaviour-based system allows the robot to have wide application-specific behaviours where each behaviour is concerned with a sole objective. They have to be combined in intelligent ways to meet the mission objective (Abdellatif 2008).

2.3.1 Competitive and Cooperative Behaviour Control

Robot control usually handles diverse and variant behaviours to be able to perform complex tasks, in many intelligent techniques; more than one behaviour can be activated at a time. To solve such abnormal situations, a control system should be employed to determine the output of the overall system. Coordinator function which can be called behaviour arbitrator is used to solve these conflicts. There are mainly two types of coordination function: Competitive Methods and Cooperative Methods (Altuntaş 2003).

In competitive control methods, only one behaviour affects the actuator output of the robot in a particular moment. In cooperative control methods, different behaviours may contribute to a single actuator action although with different strength. Figure 2.5 illustrates behaviour coordinator (Floreano & Mattiussi 2008).

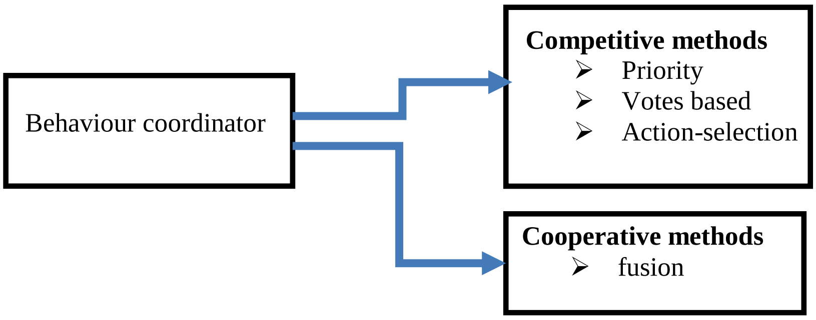


Figure 2. 5. Competitive and cooperative methods

2.3.2 Conflicting Behaviour

Robot should make a force closure on the object to make the robots able to move the object on a desired trajectory. Meanwhile the robot should avoid any obstacle to prevent its structure from any collision or damage.

(Antonelo et al. 2008) defined two conflicting behaviours which are: the first behaviour performs exploration of the environment ignoring the existent targets, whereas the second behaviour seeks and captures targets in the environment, avoiding collision with obstacles.

a) **Conflicting behaviour considering their effects**

(Mali 2002) detects conflicting behaviours from their effects on the world. The consequences of behaviours after they are performed allow identifying which behaviours can occur after the initial behaviour. Thus, behaviours which have the same effect are conflicting behaviours. For instance, if the consequence of behaviour B_1 *contains P and consequence of Behaviour* B_4 *contains P*, these behaviours conflict (Mali 2002).

b) **Conflicting behaviour considering stimulus conditions**

Conflicting behaviours have the same surrounding area condition (stimulus condition), but the consequences of these behaviours are different. To solve this situation, many strategies at the high-level control involve making a decision about the suitable behaviour to fulfill the mission goal.

For instance,
Behaviour 1
"if there is an object very close, then move away"
If stimulus condition represents an obstacle or an object, the consequence is to move away.
Behaviour 2
"if there is an object very close, then move forward towards the object"

Conflicting behaviours have the same stimulus conditions:
If the stimulus condition represents an obstacle or an object, the consequence is to move forward towards the detected obstacle.

2.3.3 Behaviour Selection

Autonomous robots usually tend to model human and/or animal behaviours which are quite complicated. Generally speaking, complex behaviour of an autonomous robot needs action selection to answer the question "*what to do next?*". Action selection should be able to deal with a variety of levels of abstraction.

Depending on the problem complexity, the behaviours can be divided into many categories to express multi-task and conflicting behaviours. A set of possible behaviours is usually predefined and fixed. The behaviour components can be decomposed into simple components on many layers, with the behaviour control coordinating the overall behaviour.

Action-selection algorithm is employed for its ability to solve the conflicts that arise from different behaviours (Brom & Bryson 2006). This algorithm is a mechanism designed to solve conflicts between rules when more than one of their conditions is held at a given instant.

Behaviour selection characteristics:

- The behaviour selection has to deal with dynamic and unpredictable environments and select the suitable behaviour;
- The behaviour selection should make decisions in a timely fashion respecting real-time task performance;
- The robot performs several different tasks. These tasks may conflict for resource allocation. Behaviour selection should coordinate the conflicting behaviours depending on the mission priority.

Action selection approaches rely mostly on condition–action rules. A condition-action rule is a rule in the form: IF Condition THEN Action. If the condition complies which a specific rule, then perform the action according to the respective rule.

The rules are organized in flat structures, as in the case of simplified subsumption architecture or in hierarchical structures. Flat structures can be applied to simple behaviours while the hierarchical structure can be employed to solve a very intricate set of conditions.

2.3.4 Behaviours Selection Algorithm for Multi-Behaviour Navigation at Reactive Level

Figure 2.6 shows a flowchart of a behaviour selection system which activates separate behaviour modules to fulfil the overall mission goal. The first priority is that a mobile robot moves to handle any object located in its local path; the second situation is that it applies aggressive behaviour to recover the robot from a hazard zone. When the robot faces an obstacle, avoidance obstacle behaviour is fired. Wall following and Seek the goal behaviours are employed allowing the robot to reach its destination.

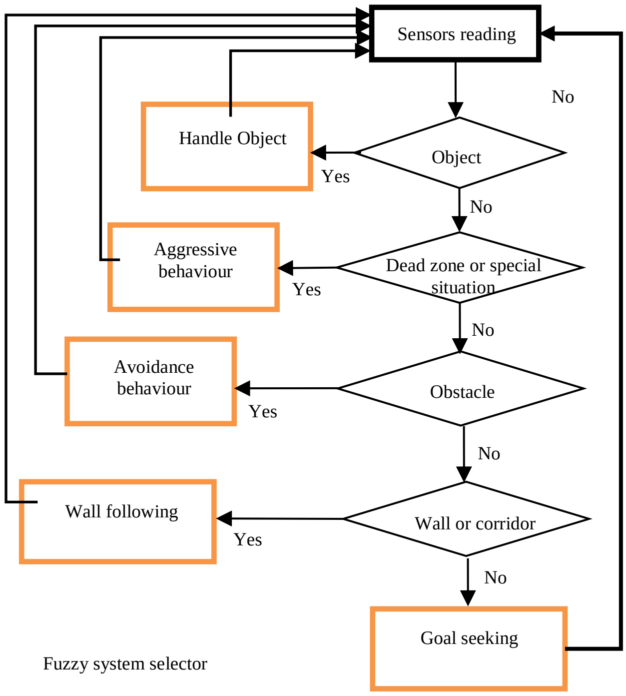


Figure 2. 6. Behaviour selection system at the reactive level.

2.4 Multi-behaviour Navigation: State of the Art

2.4.1 Multi-behaviour Navigation architectures

2.4.1.1. Traditional Sense-Plan-Act Architecture

A control system for an autonomous robot contains many subsystems responsible for perception, world modelling, planning, task execution and motor control (Murphy 2000). Figure 2.7 illustrates an approach building a control algorithm for a mobile robot, which is the “sense-plan-act” architecture. The robot problem was decomposed into sequential functions, which has the following functionalities:

- Monitor the surrounding environment
- Make an internal plan of the area
- Adapt the robot plan
- Execute the plan
- Create a new plan when environmental changes occur

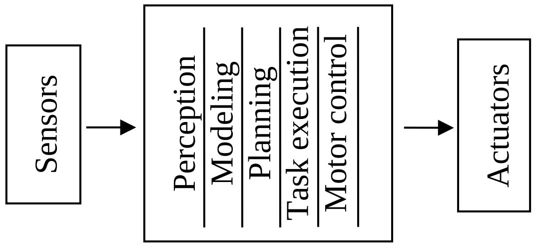


Figure 2. 7. Traditional decomposition of a mobile robot control system into functional modules

The advantages of this navigation architecture are task learning and planning abilities and its goal orientation.

2.4.1.2. Behaviour-Based Architectures

Behaviour-based architectures guide the designer in decomposing the control system into behaviour-related subsystems.

- **Brooks Architectures**

In 1986 Rodney Brooks came with a new approach, which decomposed the problem into behaviours instead of function components (Brooks 1986), and this is illustrated into a set of behaviours layers, as shown in Fig. 2.8. Behaviours could be obstacle avoidance, wall-following, exploration or target seeking. A certain number of behaviours run as parallel processes, while each behaviour can access all sensors, only one behaviour can have control over the robot actuators.

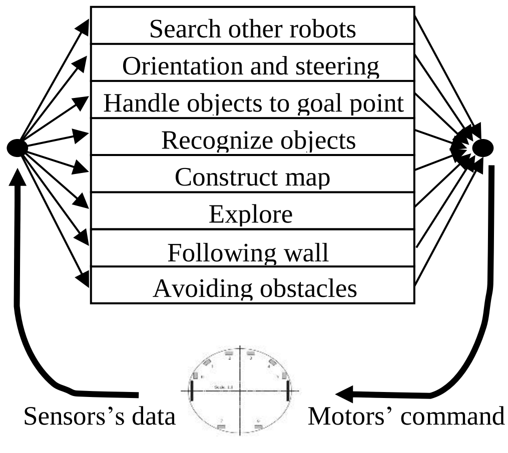


Figure 2. 8. A decomposition of a mobile robot control system based on task achieving behaviours

- **Subsumption Behaviour Architecture**

Usually, mobile robot control architecture is organized in layers which is called subsumption architecture as shown in Fig. 2.9. This architecture puts the whole system in a number of behaviours operating asynchronously and in parallel (Watanabe 2009). Each layer handles different behaviours defining purpose and responsibility. This allows the system to operate robustly and the robot architecture can be extended by simply adding new layers on top of the existing layers. The sensors and the actuators are connected directly by behaviours. There is a close relationship between each layer; the upper layer also can control the outputs of the lower layer.

Disadvantage

- The design of the higher layers is so complicated because it should have the ability to do everything that the lower layers can do (Yongjie et al. 2006);

- Each layer works without any information about what the other layers do (Brooks 1986) (Yongjie et al. 2006);

- This system uses the current sensory input without prior knowledge of the environment (Halal & Dumitrache 2007) which is not capable of performing complex tasks considering the planning ability (Yongjie et al. 2006).

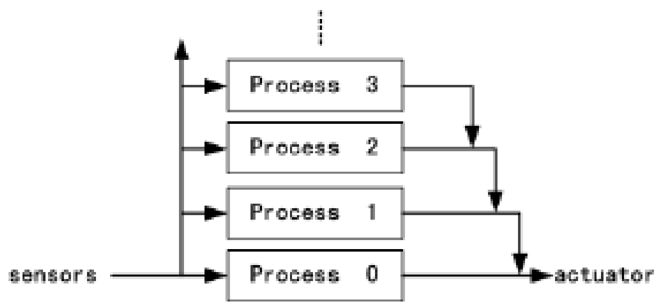


Figure 2. 9. Subsumption Architecture (Yongjie et al. 2006)

- **Modified Subsumption Architecture**

(Yongjie et al. 2006) brought out hybrid architecture based on subsumption architecture; this system is based on the following strategies:

- Each layer has a group of independent processes which is responsible for a function of a layer or a simple task.

- The behaviour-managing layer controls the behaviours changing the state of the processes (activated or not) depending on the sensors' reading. This layer also adjusts the behaviour parameters of the robot according to the current task.

- The scheduler decides which process will control the robot and the final output parameters to the actuator according to the output of every process.

The whole structure is illustrated in Fig. 2.10.

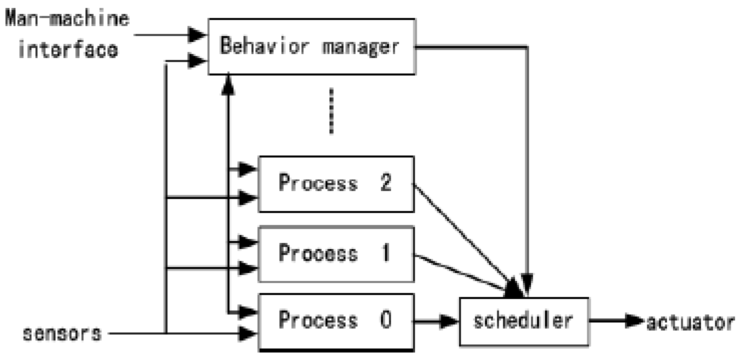


Figure 2. 10. Modified hybrid architecture (Yongjie et al. 2006)

A six layers control architecture was used by (Yongjie et al. 2006) with the following processes associated with the layers:
Process 0: wandering
Process 1: obstacles avoidance
Process 2: moving to the goal
Process 3: recovering from the deadlock
Process 4: planning routes in a known environment

Process 5: behaviour management scheduler: deciding the control layer and the final output.

Behaviour selection strategies are the key of all control architecture based on behaviours. thus, behaviour management is needed.

In this paper, an attention mechanism with different priorities is used to coordinate behaviours of all layers. The level of attention is different, and it is adjusted by the behaviour managing layer depending on environment, activity and task limiting conditions. **This system implements multi-behaviour navigation that performs well with coordinating behaviour at the reactive level.**

2.4.2 Intelligent Techniques for Multi-Behaviour Navigation

The classical approaches for solving the path planning problem, such as the cell decomposition approach, the artificial potential field approach, and graph search methods using grid-based map, suffer from many disadvantages, such as time consumption and getting trapped in a local optimum, especially when dealing with large environments with numerous solutions. Metaheuristic approaches were introduced to solve the path planning problem in a large-scale environment. A powerful method that can perform the path planning optimization is the ant colony approach.

2.4.2.1. Ant Colony for Vehicle Routing Solution

The family of Ant Colony Optimization (ACO) algorithms is used in several combinatorial optimization problems in both static and dynamic environments. The ACO optimizes the path planning to fit the goal of the multi-objective function.

An ant path starts from the start point which represents the nest or the first node and passes through nodes 2, ..., $n - 1$ for checking the neighbourhood of the node to improve the gain in the node vicinity. When an ant goes from node $j - 1$ to node j, a specific number of neighbourhoods are first investigated randomly from the *j* node surrounding area, regardless of the pheromone levels which are compared later. The neighbour that has the highest-level pheromone is selected to be the new node. A new local path is generated from node *j-1* to the new node *j*. The ACO improves the local search and updates the local paths to get the highest neighbourhood values. Figure 2.11 shows an ACO vehicle routing solution which is used to optimize the traffic problem (Kponyo et al. 2014).

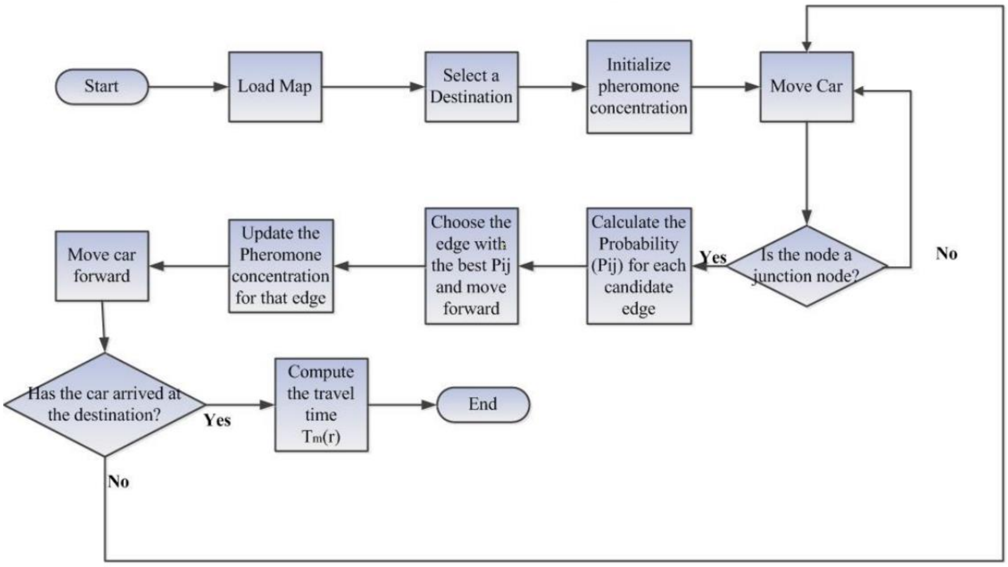


Figure 2. 11. ACO vehicle routing solution (Kponyo et al. 2014)

2.4.2.2. Multi-Behaviour Based Multi-Colony Ant Algorithm for TSP

(Liu & You 2009) proposed an ant colony system which consists of several sub-colonies; each sub-colony of ants navigates the environment and performs its task using its own search behaviour. The ant population evolves independently and in parallel. The system is divided into four different behaviour options (subsystems).

The behaviour characteristics are described as follows:

Behaviour 1: an ant selects the next city in a stochastic selection manner.

Behaviour 2: an ant selects the next city using a greedy selection technique.

Behaviour 3: an ant selects the next city applying a hybrid behaviour.
Behaviour 4: an ant selects the next city in a "follow the crowd" manner every time that ant's behaviour can be guided by the perceived behaviour of other individuals.

Each sub-colony of ants has different behavioural characteristics and evolves independently. An example of the parallel evolution of four sub-colonies is shown in Fig. 2.12.

This system can perform different deliberative searches using multi-behaviour to navigate the global path. But this system shows low performance at the reactive level because it doesn't handle local reactive behaviours.

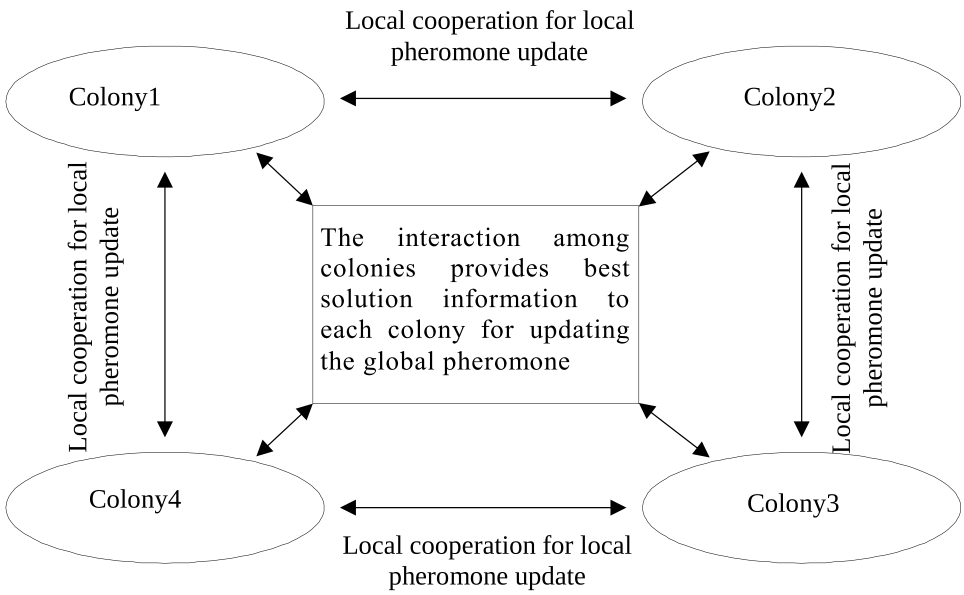


Figure 2. 12. A framework of multi-colony parallel evolution (Liu & You 2009)

This system offers a good method to handle multi-behaviour navigation at the deliberative level, but still shows disadvantages in terms of handling multiple behaviours because it doesn't consider multi-class environments. We employ this system idea in our multi-behaviour navigation scheme for large-scale missions.

2.4.2.3. Fuzzy Behaviour-Based Systems

Behaviour-based systems allow for robot application-specific behaviours where each behaviour is concerned with a sole objective. They have to be combined in an intelligent way to meet the mission objective (Abdellatif 2008).

Fuzzy systems (FS) have been widely used in reactive navigation. Fuzzy control systems have been developed in mobile robotics for exploration, localization and map building tasks (Pradhan et al. 2009) (Gu et al. 2003) (Yang et al. 2005). FS design consists of three stages that are as follows: definition of membership functions, determining the number of useful rules and rules consequent parameters.

Fuzzy systems mimic the human reasoning decisions which are based on IF-THEN rules. Each rule determines the conditions as they relate to a specific behaviour. More behaviours can be added to the system as needed. The fuzzy rules are defined based on the robot tasks. The robot task is determined by the user depending on the objective of the robot mission (Zein-Sabatto et al. 2003). The fuzzy system integrates and coordinates the tasks to form complex robotics system such as the fuzzy hierarchy system (Lee et al. 2003).

The inputs and outputs of the system are transformed into fuzzy variables which are called linguistic variables that are characterized by words rather than by numbers which are defined by different membership functions which vary from 0 to 1. In the rules base, the decisions are made according to the linguistic rules in the fuzzy systems using IF-THEN rules base (Narvydas et al. 2007).

Fuzzification is defined as the mapping from a real-valued point to a fuzzy set. In most fuzzy decision systems, non-fuzzy input data is mapped to fuzzy sets by treating them as Gaussian membership functions, triangular membership functions and trapezoidal membership functions.

The fuzzy inference engine is used to combine the fuzzy IF-THEN rules and to convert input information into output membership functions. An inference mechanism emulates the expert's decision making in interpreting and applying knowledge about how to perform a good control.

Examples of fuzzy systems that have been applied in behaviour-based control are reported in (Brooks 1986) (Kuo & Ou 2009) (Raguraman et al. 2009). Fuzzy behaviour-based system fuses different types of behaviour using fuzzy reasoning. Inference fuzzy system fires all types of behaviours depending on the rules base and the system inputs. The fuzzy rules base system can handle the behaviour-based system, where each rule or a group of rules can represent one behaviour.

- **Multi-Behaviour Fuzzy System**

A method for navigating in an unknown environment has proposed by (Bao et al. 2009) this system has four behaviours handled by fuzzy systems: goal seeking, obstacle avoidance, tracking, and deadlock disarming. A behaviour controller is designed to integrate these basic behaviours and to determine which behaviour has the control of the robot actuators. This system is illustrated in Fig. 2.15.

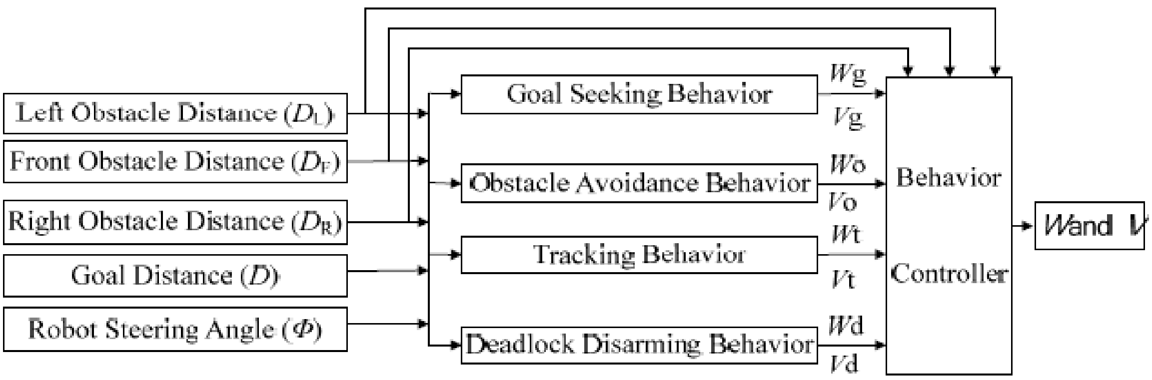


Figure 2. 13. Multi-behaviour fuzzy controller

The behaviour arbitrator resolves any conflicts arising from multiple behaviours attempting to control the same actuator simultaneously. High-Priority-Take-All strategy is adapted by most of the existing Behaviour Arbitrators determining control action of the entire system.

A behaviour controller concerning the obstacle distance DL, *D*F and *D*R is designed by (Bao et al. 2009). The obstacle distance is used to form many rules that decide which behaviour has high priority. The mobile robot is controlled by the highest priority behaviour as long as the mobile robot moves towards the goal. Fig. 2.16 illustrates this control.

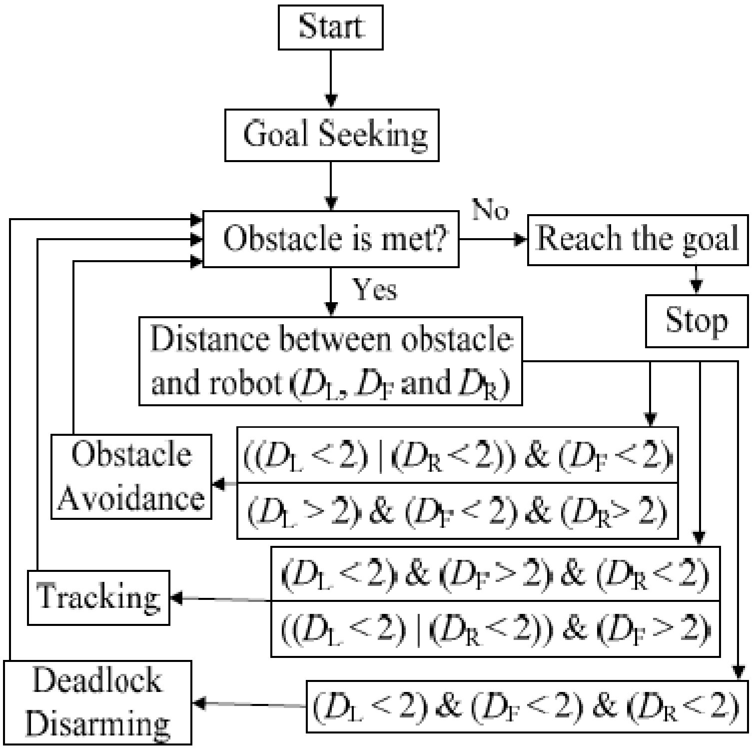


Figure 2. 14. Flowchart of multi-behaviour navigation (Bao et al. 2009)

This multi-behaviour system deals with the reactive level where the behaviour controller can disarm the deadlock behaviour when the robot gets stuck. **This system handles different behaviour models and applies a coordinator to arm and disarm these behaviours according to the robot situation and the mission objective.**

2.4.2.4. Genetic Algorithms in Behaviour-Based Systems

The controller task is to generate a good solution to resolve the robot problem. The fitness function evaluates the effectiveness of the chromosome by returning a numeric value that corresponds to the measured chromosome.

The genetic algorithm controller is based on an analogy with the genetic structure and behaviour of chromosomes within a population of individuals. The basic operation of a genetic algorithm can be summarized as follows, and can be seen in Fig. 2.13 (Halal & Dumitrache 2006)

Step 1: Generate an initial population of *N* solutions generated randomly or heuristically

Step 2: While the terminating criteria have not been satisfied

- Evaluate each solution of the population using a fitness function/objective function.
  - Construct the phenotype (e.g. simulated robot) corresponding to the encoded genotype (chromosome)
  - Evaluate the phenotype (e.g. measure the obstacle avoidance abilities), in order to determine its fitness

Step 3: Select solutions as parents for the new generation based on either elite solution selections or random solution selections;

Step 4: Use the parent solutions from Step 3 to produce the (offspring) population by applying the crossover operator;

Step 5: the mutation operator is randomly or heuristically applied to solutions using a mutation probability to generate the mutation population;

Step **6**: generate the new generation from the big search family using certain selection schemas and genetic operators.

Step 7: Repeat Steps 2 through 6 until a stopping criterion is met.

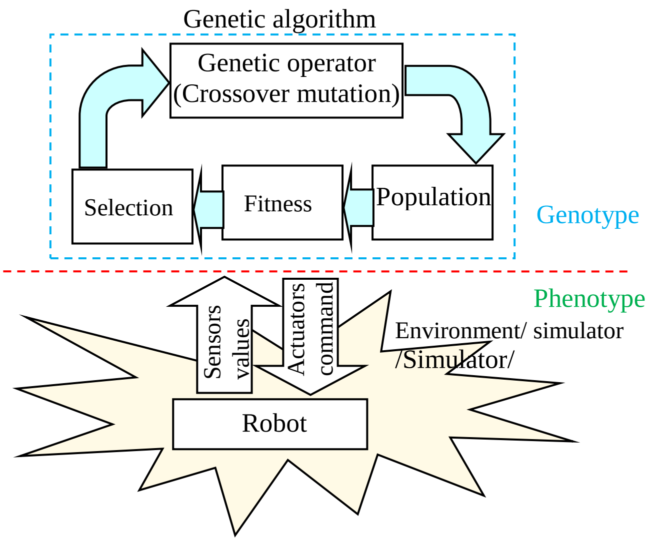


Figure 2. 15. Genetic algorithms in mobile robots control (Halal & Dumitrache 2006)

Genetic algorithm has been applied to handle the behaviour-based system (Halal & Dumitrache 2006). We are dealing with phenotypes, which are possible solutions to a given problem, for example, a simulated robot with a particular control structure as is shown in Fig. 2.14, and genotypes, which are encoded representations of phenotypes. The chromosome handles the behaviour-based system; the coding string has the most important role driving this system according to mission conditions. Robot behaviours are represented by a whole chromosome. Each behaviour should correspond to a gene or a group of genes in the coded chromosome, as shown in Fig. 2.14. Genetic operators work only on genotypes; phenotypes can represent a simulator or an environment where the control determines the values of the gene by applying a suitable fitness function to fit the mission goal.

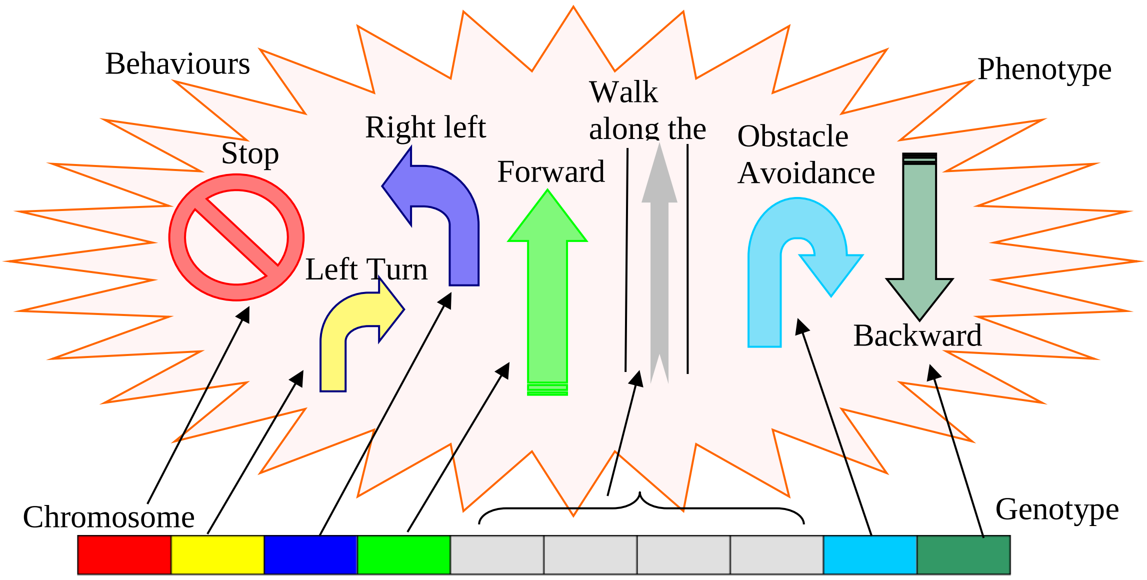


Figure 2. 16. Genetic algorithm handles behaviour based system

2.4.2.5. Multi-Behaviour Neural Network

In (Nojima 2009), Yusuke Nojima divided complicated tasks into a combination of behaviours based on apparent functional modularity. Each local fuzzy system handled a decomposed behaviour. A gating network is used for combining the outputs of local fuzzy systems. Behaviour coordination can be performed by the combination of outputs from fuzzy controllers.

The multi-objective behaviour coordination strategy work as follows:

When the robot faces an obstacle, the weight of the collision avoidance behaviour is updated. Otherwise, the weight of the target tracing behaviour is updated. After the update, behaviours weights are normalized in the range of [0 1]. This method can be considered as a mixture of fuzzy systems because the behaviour coordination mechanism is considered as a gating network. Fig. 2.17 shows the multi-objective behaviour coordination control.

This system can control the fuzzy behaviours by tuning the fuzzy membership function in order to select one behaviour over the others.

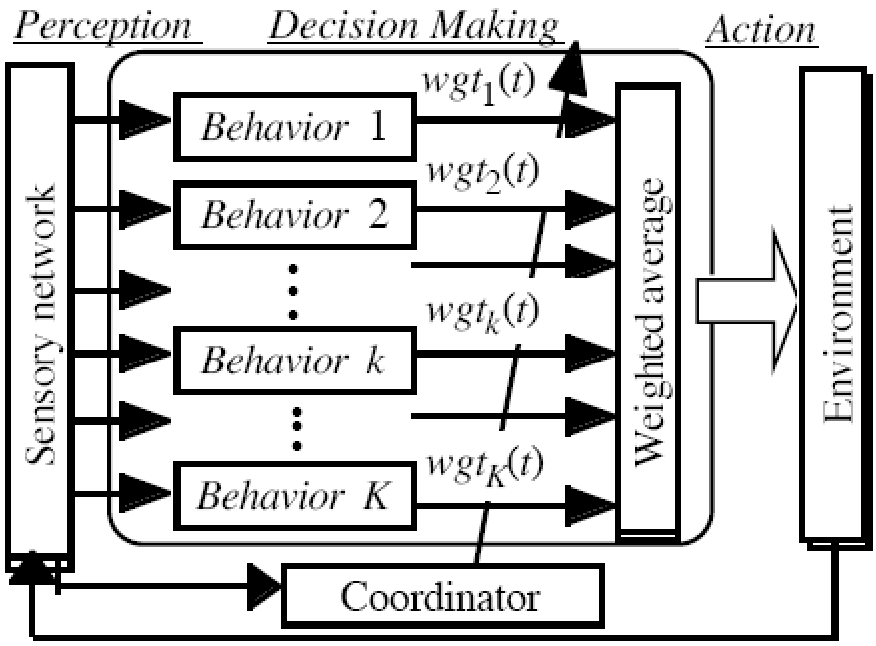


Figure 2. 17. Multi behaviour coordination

2.4.3 Hierarchical Multi-Behaviour Navigation

2.4.3.1 Hierarchical Multi-Behaviour Model

(Tunstel 1996) proposed a behaviour hierarchy model for indoor navigation organized as in Fig. 2.18. The system's main task is goal directed navigation which is decomposed as a behavioural function of goal seeking and route following which are considered as composite behaviours. These behaviours can be further decomposed into the primitive behaviours which can be combined synergistically to produce suitable behaviours for accomplishing goal-directed operations. Primitive behaviours usually associated with activation threshold which are at the root of the intelligent coordination of primitive behaviours. Fig. 2.18 **shows a conceptual model of a hierarchical intelligent behaviour system and its behavioural relationships.**

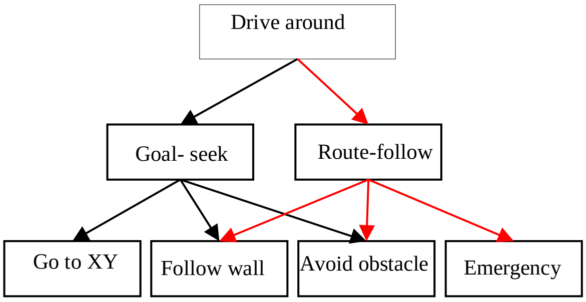


Figure 2. 18. Hierarchical scheme of intelligent behaviour system (Tunstel 1996)

The multi-task robot must be able to perform different behaviours including the conflicting behaviours. In order to accomplish its task, the priorities of the behaviour must change with time according to the mission objective. Therefore, a controller (behaviour arbitrator) must select and actualize the suitable behaviour that can be integrated to fit the overall goal of different control objectives (Mai & Janschek 2012).

A hierarchy based behaviour system is employed in (Mai & Janschek 2012) using a fuzzy system to control path planning and motion control modules. The behaviours were decomposed into bottom-up behaviours. A complex behaviour may have serval layers in which each activity at a given level is dependent upon behaviours at the levels below. Two behaviours were introduced into composite and primitive behaviours. The composite behaviour or the behaviour arbitrator has a high level and control the lowest layer which includes the primitive behaviours. A collection of primitive behaviours resides at the lowest level which is also called the primitive level. The primitive behaviours are simple and perform a single purpose. Fig. 2.19 shows the hierarchy of path planning behaviours.

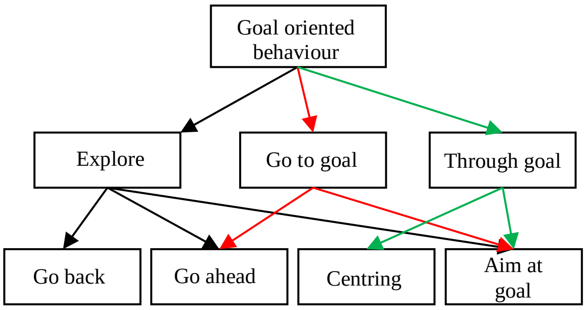


Figure 2. 19. Hierarchy of PAPFIS behaviours

Based on the information about obstacles, the motion control fuzzy system (MOFIS) computes primitive and composite behaviours to determine a collision-free behaviour. Figure 2.20 illustrates the behaviour based architecture for the motion control.

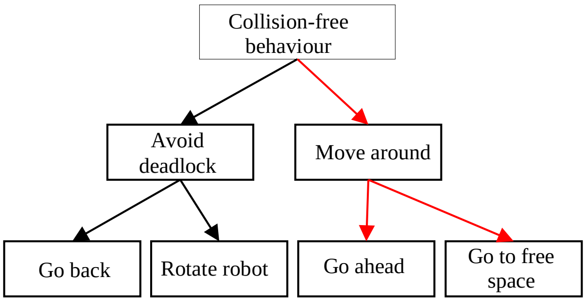


Figure 2. 20. Hierarchy of MOFIS behaviours

This system handles the multi-behaviour navigation at both levels, which represents the traditional hybrid multi-behaviour navigation. Since the system depends on sequential ordering tasks, it performs poorly when executing its mission in a complex and unstructured environment.

2.4.3.2 Cooperative Behaviour Hierarchy Model

(Vadakkepat et al. 2004) used behaviour-based architecture to decompose a complex multi-robotic system into three hierarchy layers which are robot roles, robot behaviours and robot actions. The robot strategy is divided into many robot roles such as attacker, midfielder, defender and goalie. The robots are able to execute the role by activating the suitable behaviour depending on the surrounding area conditions. Behaviours locate at the next lower layer. Also, the behaviours activate the suitable actions to accomplish the task. The action layer is the lowest layer. Depending on the task, the action layer could have many sub-layers. An extensive fuzzy behaviour-based architecture is proposed for the control of mobile robots where the fuzzy system is used to coordinate the various behaviours, to select roles for each robot and, for robot perception, decision-making, and speed control. The system was designed to show cooperative behaviour so that the robots team exhibiting good collective behaviour. Fig. 2.21 illustrates the proposed architecture which decomposes behaviours into a hierarchy based on complexity. The complex behaviours are decomposed into simpler and more manageable sub-behaviours from the top to bottom of the hierarchy.

The path planning in this context consists of sequential tasks such as chase, kick and avoid the wall. **This system implements a complex multi-behaviour strategy that is handled by many layers. This system is implemented in the reactive level showing the limitations of the planning ability.**

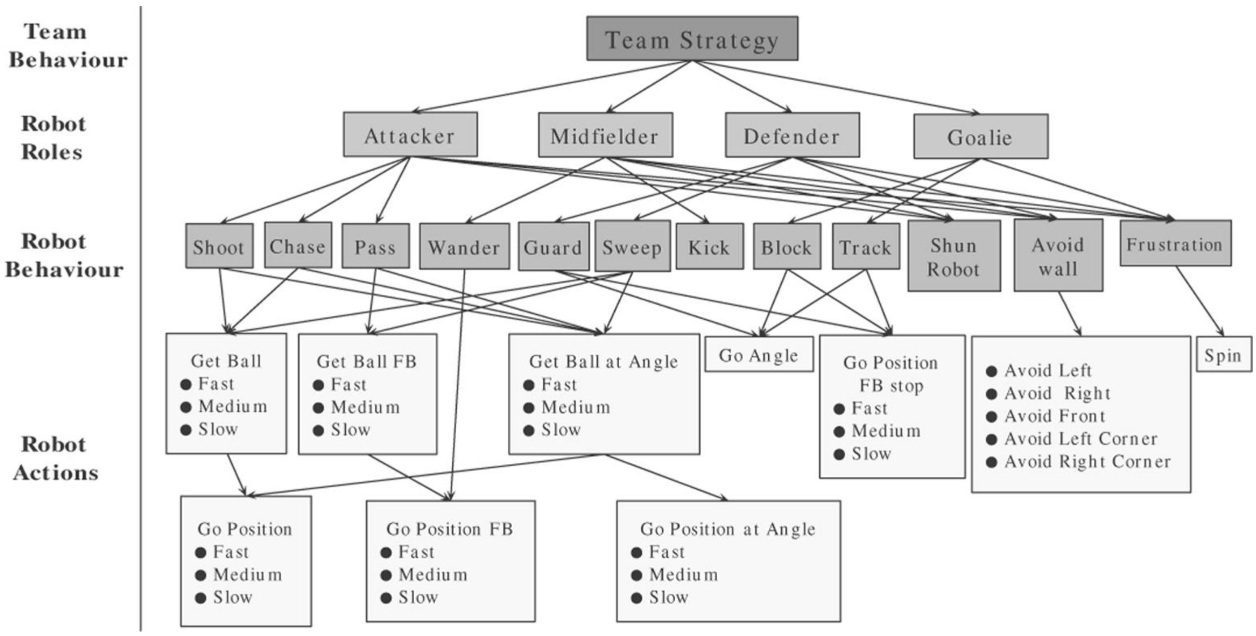


Figure 2. 21. **Behaviour architecture for a team of soccer robots** (Vadakkepat et al. 2004)**.**

2.4.4 Hybrid Intelligent Control for Multi-Behaviour Navigation

2.4.4.1 Hybrid Geno-Fuzzy Multi-Behaviour System

(Yan Yongjie & Zhang Yan 2009) designed a Geno-fuzzy multi-behaviour system to drive the robot in an unknown and static environment. They proposed a three behaviours control structure based on the fuzzy system which are: avoid obstacle, avoid robot, and move to the goal. The behaviour synthesis is always the key for designing behaviour-based control. The behaviour integration unit works depending on two factors: the priority and the weight value. The avoid obstacle and the avoid robot behaviours have the high priority whereas the priority of move to the goal is defined as the lowest one. The unit inhibits the output of the move to the goal behaviour while the robot is avoiding a robot or an obstacle. In this case, the control output which are the robot speed and the robot heading angle are determined by two factors: depending on the other robots' speed and heading angles and the controlled robot's speed and heading angle when avoiding an obstacle. Equations (1.1) and (1.2) explain these two factors:

$$v = \lambda_d \, v_{robot} + (1 - \lambda_d) v_{obstacle} \qquad (1.1)$$

$$\Delta\theta = \lambda_d \, \Delta\theta_{robot} + (1 - \lambda_d) \Delta\theta_{obstacle} \qquad (1.2)$$

where:

λ_d is the weight value of the most dangerous robot; λ_d is decided by robot's angle φ and distance d.

The outputs of avoid robot behaviour are v_{robot}, and θ_{robot}.

The outputs of avoid static obstacle behaviour are $v_{obstacle}$, and $\theta_{obstacle}$.

Fig. 2.22 illustrates the geno-fuzzy multi-behaviour system.

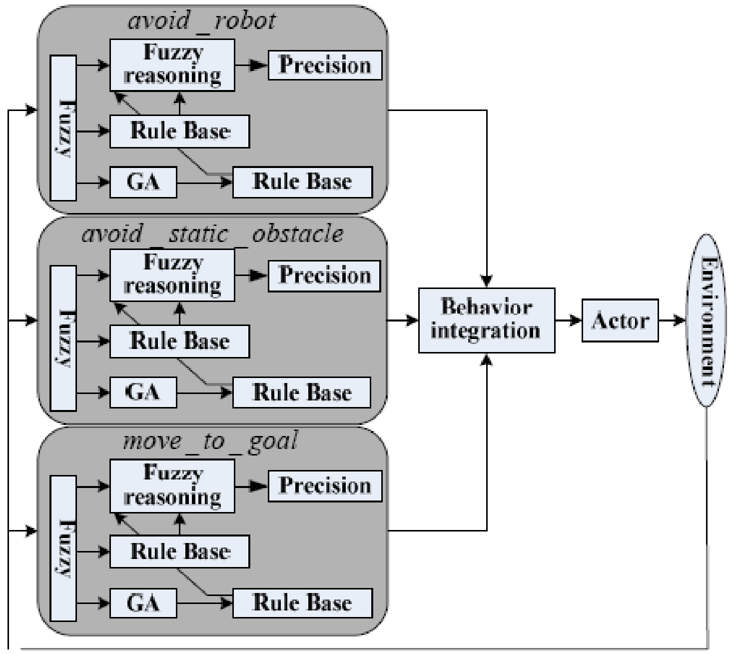


Figure 2. 22. Geno-fuzzy multi-behaviours system

This system controls multi-behaviour navigation using an integrated system. The Geno- fuzzy system changes the propriety of its behaviours depending on certain factors. This system deals with the reactive level only, which means it has no planning ability.

2.4.4.2 Hybrid Multi-Behaviour Navigation Architecture

(Zaremba et al. 2015) proposed a functional diagram depicting the deliberative-reactive control of the cruise ship as shown in Fig. 2.23. This model was the first hybrid multi behaviour navigation architecture for large-scale environments.

The reactive system has many sub-reactive systems where each system which handles a behaviour or a group of behaviours work on the same task. The proposed system performs complex tasks as well as conflicting behaviours. Behaviour selector is responsible for making a decision on the suitable behaviour depending on the robot status and the surrounding area conditions. In this model, the behaviour selector represents a decision maker at the reactive level where decisions are made regarding the local context conditions.

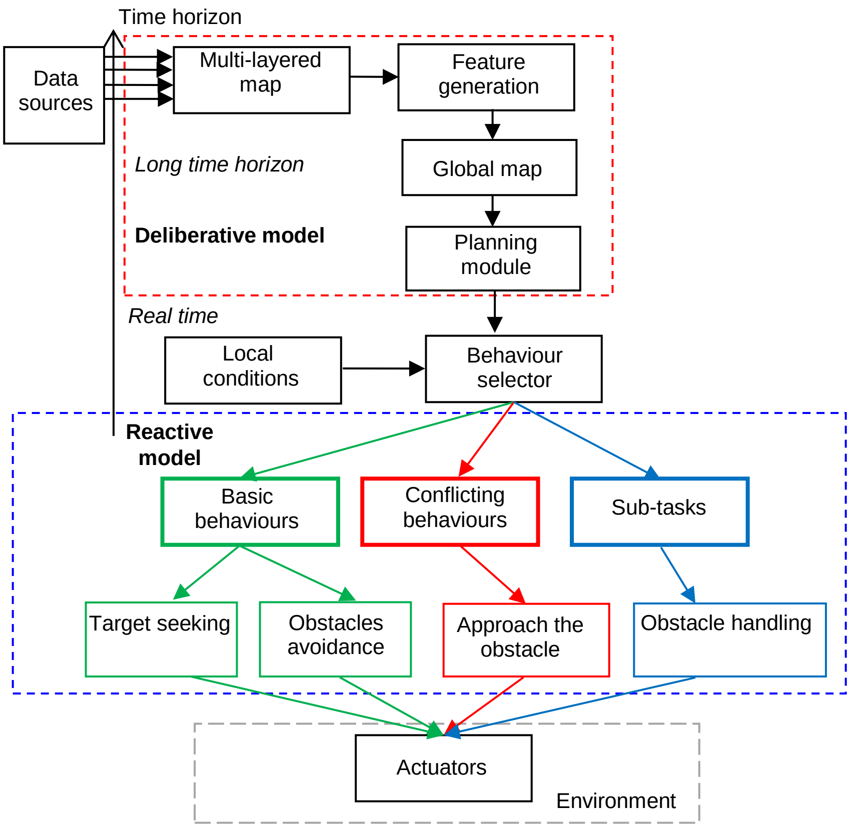


Figure 2. 23. Hybrid multi-behaviour navigation

The disadvantage of this model is that the behaviour selector makes his decision depending on the local condition only. Thus, the deliberative navigation does not support multi-behaviour operation in dynamic and unstructured environments.

2.5 Conclusions

The hybrid navigation provides a suitable degree of reactivity and deliberation. It offers a high level of artificial intelligence and has fast reactiveness, which allows the robot to perform well in a dynamic environment. Global navigation produces an optimal global plan for a given task, and then reactive navigation executes this global path into many local paths. An optimal sub-global plan is provided when the environmental changes prevent the robot from performing its tasks. Reactive control deals with onboard sensory data to make a decision preventing the robot from any collision. Reactive control discovers the surrounding area and observes the environment changes and updates the world model to reach real-time information.

The current models of navigation control architectures partially solve the multi-behaviour navigation dealing with one navigation aspect, but they don't fully consider complex and unstructured environments. They either consider a simple environment model at one navigation level or adopt the sequential ordering of the tasks. Thus, both navigation types are not integrated, which is the source of limitations in executing more complex tasks.

Large-scale complex and unstructured environments need a robust architecture which is able to deal with the environmental changes. The system should show flexibility and adaptability when performing its mission by employing both navigation levels.

Chapter 3

Thesis Objectives & Contributions

3.1. Objectives

The principal objective of this thesis is the development of an integrated, heuristics-based methodology for designing hybrid deliberative-reactive navigation systems for complex and dynamic environments, where the mobile robot executes multi-task navigation subject to different behaviours. Hybrid deliberative reactive navigation architecture will be investigated, which is to assure a seamless relationship between the deliberative and the reactive navigation and improve the overall quality of the navigation.

The aim is to obtain a control system that provides a suitable degree of reactivity and deliberation. This planning and control system has to have the ability to control successfully multi-behaviour strategies for a large range of tasks, including conflicting tasks. The system should exhibit a high level of intelligence in order to make the robot perform in unstructured and dynamic environments.

In this thesis, we are dealing primarily with large-scale navigation environments. Due to the complexity of the navigation context and the heterogeneity of data sources, one of the initial objectives is an appropriate environment representation. Multi-layer maps will be explored as the underlying mechanism to represent the environment data and to build the world model. A variety of methods will be employed to better interpret the global environment with respect to the mission objective. Computational intelligence techniques will be investigated to classify the environment in order to extract proper entities, such as regions of interest, obstacles, etc. in the global context.

One of the main goals is to provide a formal architectural framework comprising principal components of the hybrid navigation system and the transitions between those components.

3.2. Proposed Investigation Approaches

In order to achieve the thesis objectives, we propose and investigate the following approaches:

3.2.1 Hybrid Deliberative-Reactive Approach

Deliberative navigation control supervises reactive control, whereas the reactive navigation makes a local decision and provides feedback to the deliberative navigation about the surrounding area in the form of real-time sensory information. The reactive navigation is fast and prevents the robot from any structural damage, but his performance still shows lack of intelligence regarding the global goal. Usually, the reactive control dominates when onboard data doesn't comply with the global data (world model). The deliberative navigation system will be developed to get an optimal trajectory in multiple fields using hybrid metaheuristic search methods. In such conditions, the need to closely couple the two types of navigation is necessary for dealing with environmental changes in a cooperative way. Both navigations should work together in a parallel way while intelligent systems coordinate the role of each navigation type as well as their collaboration.

3.2.2 Metaheuristic Approach for Navigation Optimization

Metaheuristic approach which assures multi-behaviour navigation is employed to improve and to enhance the search process in path planning. It enhances the deliberative level by producing the optimal global path and sub-global paths. Due to the environment complexity conditions, it is considered a NP-hard problem which involves solving the sequential ordering problem with the precedence of constraints to determine the route optimization model of multi-modal travel. It is difficult to get an optimal global solution via exact algorithms within an acceptable time. Most of the NP-hard problem needing efficient solutions call for the combinatorial optimization approach. The approach that has proved effective in solving combinatorial optimization problems is the use of hybrid metaheuristic. Hybrid GA system (AGA) using waypoint navigation is developed to perform multi-behaviour navigation. Deliberative level is used to generate a global path planning which deals with multiple areas (e.g., different water pollutant patches) in a partially unknown and unstructured environment. Hybrid Metaheuristic is employed to improve the deliberative navigation allowing an efficient and flexible multi-behaviour path planning.

3.2.3 Multi-Behaviour Operation

Dealing with different classes of the entities creates the need to use a wide range of behaviours to comply with different types of environmental conditions. Metaheuristic methods can perform different behaviours during the search process where each method can apply its search strategy, such as the sub-colonies in ant colony optimization (Liu & You 2009). Genetic algorithm parallel search or hybrid Meta heuristic method can perform different search behaviours in a respective zone.

Traditional behaviour coordinator approach will be investigated and developed in order to deal with multi-behaviour navigation including the conflicting behaviour. Robot control should handle multiple behaviours to be able to perform complex tasks. Thus, Multi-behaviour navigation will be studied as a more robust and flexible solution for the complex and dynamic environment, in situations such as recovering the robot from dead-end zones and cycling modes. Many behaviour strategies will be adopted and designed providing a big range of tasks which help the robot perform different behaviours. Behaviour selector will be developed using hierarchy intelligent techniques to select the appropriate behaviour regarding the robot situation and respecting the global and local conditions. An integrated system combining both levels of navigation will be developed to control behaviours in both levels to assure that no interaction between the behaviours will occur.

3.3. Contributions

This thesis contributes to the development of intelligent navigation systems in the following areas:

- Solution of the problem of integrated, multi-behaviour navigation in large-scale, complex and dynamic environments.
- Employment of the multi-layer environment representation model for the integration of multi-source data (remote sensing, meteorological and ancillary data) and the navigation context identification.
- Optimization of multi-strategy path planning using hybrid metaheuristic approaches. Combination of the waypoint navigation with an optimization process to create an optimal path that fits the overall mission goal, respecting at the same time the temporal and spatial constraints.
- Development of an ACOGA algorithm (Genetic Algorithms and Ant Colony Optimization) integrated with the waypoint navigation approach to deal with the partially unknown environment.
- At the reactive level, development of a special architecture able to deal with dynamic environment conditions by controlling multi-behaviour reactive navigation by a hierarchical intelligent system.
- Development of a hierarchical geno-fuzzy system used to optimize reactive navigation within conflicting behaviours.
- One of the main contributions of the thesis will be a novel formal model of an integrated deliberative reactive navigation.

Chapter 4

Integrated deliberative-reactive Multi-Behaviour Navigation architecture

4.1. Introduction

Robot control architectures are designed to provide a framework for solving mobile robot problems. A survey of different architectures for navigation control was presented in Chapter 2. Reactive system architectures solve the navigation problem using different controls depending on the tasks' complexity. Typical architectures are the classical Brooks architecture, modifies subsumption architecture, and different architectures with behaviour coordinators to activate the suitable behaviour for the current robot situation. Deliberative navigation architectures are designed to solve many problems which vary in their complexity. The Sense-plan-act scheme is the fundamental architecture to solve the mission in an a priori known environment (world model). Deliberative architectures may solve the navigation problem using many sub systems to deal with different strategies selected to carry out the mission.

Hybrid control architectures are employed to work in prior-known and partially-known environments which combine both deliberative and reactive architecture.

Hybrid deliberative–reactive architectures are usually loosely-coupled and feature sequential ordering of tasks, which is not efficient in complex conditions, thus affecting the performance of the robot behaviour. Since there are practically no formal representations of these architectures, a rationale is, thus, provided for proposing an integrated architecture that can be represented by a formal transitional model. This model should be reliable, rigid and efficient hybrid navigation architecture that provides wide navigation capabilities to ensure the performance of the robot over identified areas of interest. In this thesis, we adopt and extend the robot architecture model proposed in (Lincoln et al. 2013) for autonomous asteroid exploration.

Main features of the new proposed architecture: transitional, general, close integration of the deliberative and reactive navigation.

A **navigation system** is defined as a tuple

$$NS = \{E, \alpha, C, \beta, B, \gamma, P, \tau\} \quad (4.1)$$

This system consists of four states, *E*, *C*, *B* and *P* and four functions, *α, β, γ* and *τ*, where:

E - environment model including the processing of all sensory data;

α- context generation,

C - global and local context,

β - navigation behaviour control,

B - set of behaviours,

γ - path planning,

P - set of executable plans,

τ - trajectory generation,

Figure 4.1 illustrates the architecture of the integrated hybrid navigation system for a mobile robot. The availability of the information to every module is provided in this model.

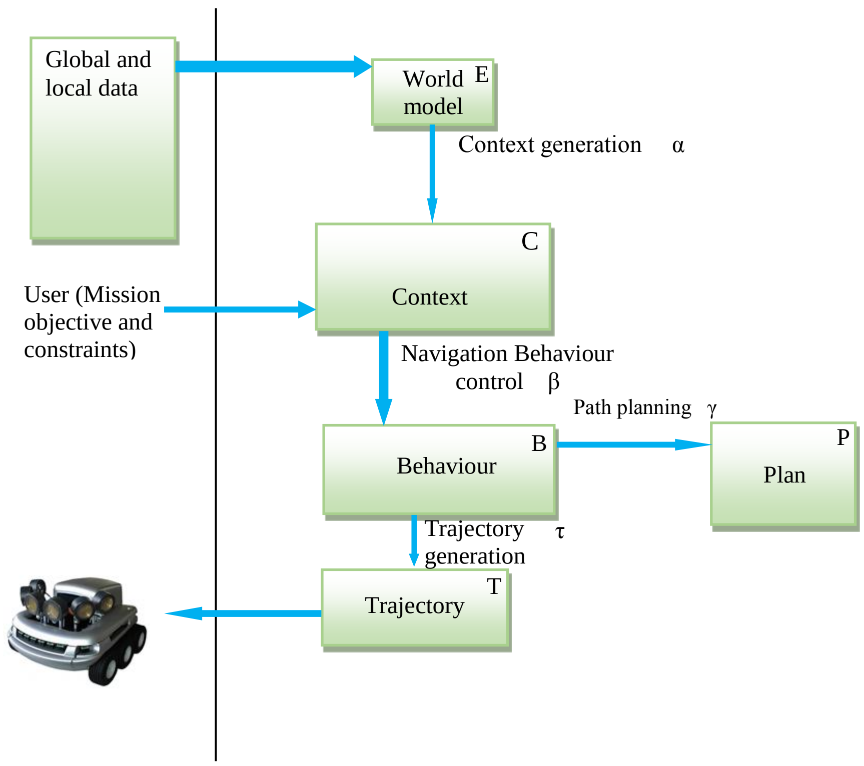


Figure 4. 1. Integrated deliberative-reactive navigation model

4.2. World model (*E*)

The robot environment (world model) is represented by a multi-layer map. Each layer represents an environment feature. Robot posture and additional robot states can be shown in separate layers. The world model is built up from the raw data - global and local data - and any ancillary information. Intelligent system techniques employing many approaches can be used to extract the necessary information needed for the mission. The world model in the deliberative control is enhanced by many information sources. Apart from remote sensing data, meteorological data and Global Information Systems (GIS) can be used to detect the environmental changes and produce multi-layer maps.

4.2.1. Data Resources

The robot must navigate from a known position to a desired new location and must orientate to execute its multi-tasks to gets to its target. For these tasks, the robot is equipped with the necessary sensors which acquire high-resolution data describing the robot's physical surroundings in a timely, yet practical fashion. The sensors are chosen depending on the robot tasks and the environment. These sensors provide the necessary data to the robot in order to interact with the physical objects and entities in the environment.

Environment sensing can be grouped into two categories, global and local:

- Global sensors are mounted outside the robot in its environment and transmit sensor data back to the robot. For instance, overhead satellite cameras and a GPS system provide the robot with its coordinates. This data is generally called long-term data.
- Local or onboard sensors: sensors mounted on the robot. Sonar or infrared sensors provide surrounding area information which is generally classified as short-term data.

Global information from satellite cameras (RS) has many advantages in environmental monitoring. These kinds of sensors have multi-spectral bands which have different applications, for instance MODIS (Moderate Resolution Imaging Spectroradiometer) has 36 spectral bands where each band or a group of bands has its application to detect and determine certain variables at the earth level (see Appendix 1.1).

Spectral bands and other information from different sensors in the robot environment are processed to form the multi-layer maps which have been employed to generate spatial and functional properties of the environment. These maps enable a robot to perceive and to interpret its environment in which each map represents one or many environment features. That is, the robot would first build a model of the world as closely as possible and then plan its actions from that model.

4.2.2. Multi-Layer Maps

The deliberative navigation control architecture is equipped with an environment model. A map generation grid-based model represents the environment by dividing it into square cell representations, which result in an N by M matrix. To simplify the world model each environment feature can be represented in one layer creating a multi-layer map.

Hybrid maps consist of topological maps and grid-based maps. Topological maps are used to interpret the global environment for path planning and decision making while grid-based or feature maps are used to describe the local environment to validate the information in topological maps (Chen & Cheng 2008). A set of multi-layer map is used to interpret the surrounding environment and to provide the capacity for dealing with multiple classes of environment objects. A multi-layer map used to interpret aquatic environments is shown in Fig. 4.2. The map consists of a bathymetric data layer, of the measured pollutants layers and meteorological data layers, and contains data related to wind speed and wave height.

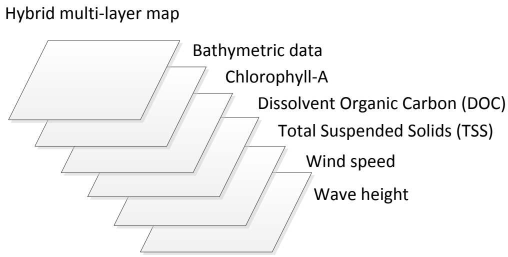


Figure 4. 2. Multi-layer map (Halal et al. 2014)

4.3. Context generation: (*α* function)

A method used for building the world model and the context (Schubert et al. 2014).

The development of an effective and efficient method is needed to extract unknown and unexpected information from datasets of unprecedentedly large size (e.g., millions of observations), high dimensionality (e.g., hundreds of variables), and complexity (e.g., heterogeneous data sources, space–time dynamics, multivariate connections, explicit and implicit spatial relations and interactions).

Context generation implies extraction of entities involved in the navigation process. In the environment monitoring context, those can be the type of pollutants and their concentration maps. In order to obtain this information from RS data, classification procedures have to be applied.

Remote sensing is commonly employing classification methods to classify image pixels into labeled categories. Classification will sort the data or information (world model) into many classes according to their properties. As an example, the water detection data may need to be classified into different pollutant types. Classification methods include, for instance, artificial neural networks (ANN), decision trees, support vector machines (SVM), and linear discriminant function (LDF),

The modelling on the input data produced in the form of multi-layer maps using different methods are listed below:

- Regression: Spatial regression or prediction models form a special group of regression analysis that considers the independent and/or dependent variable of nearby neighbours in predicting the dependent variable at a specific location, use of neural networks as model builders.

- Bands ratio: two-band, three-band, four-band or more bands ratio algorithms are constructed to retrieve the environment characteristics. This method is employed further in the thesis o extract the MCI and TSS patches

- Wavelength signature (shape): Remote sensing reflectances vary widely in their spectral shape and magnitude. Classes in the world model are determined by their wavelength reflectance features. (Li et al. 2011) (Vincent et al. 2004). These models are used to generate prior spatial distributions of environment characteristics which in turn are used by the context.

4.4. Context (*C*)

The context module consists of many components which are listed below:

- Set of available strategies to execute the mission
- Obstacle/ ROI
- Robot model (Point, kinematic and dynamic model)
- State of the navigation (Deliberative-reactive)

Most of the context components are extracted form the world model which comes in the multi-layer map form.

4.4.1. Set of Available Strategies

To deal with large-scale or complex environments, a set of mission strategies is defined by the user to comply with the mission objective, allowing the robot to perform its tasks in many different modes. Context generation and model interpretation should comply with the mission objective.

4.4.2. Region of Interest (ROI)

Region of interest (ROI) maps can be extracted from the multi-layer maps as additional layers which can be used to enhance the efficiency of the multi-task navigation. The ROI approach facilitates the planning system in directing the search towards desirable zones by paying additional attention to desired regions and assuring at the same time the generation of feasible solutions. Obstacles can be classified depending on their characteristics and then stored as in the multi-layer map.

4.4.3. Robot Types

In general, mobile robots are classified into two groups, holonomic and non-holonomic robots. A holonomic mobile robot can move freely in any direction. A non-holonomic robot is a mobile robot system with movement constraints.

The most important types of wheeled mobile robot depending on its wheels configuration (Siegwart & Nourbakhsh 2004) are as follows:

- Omnidirectional robot
- Differential robot
- Omni-steer robot
- Tricycle robot
- Two –steer robot

4.4.4. Robot model

Usually the robot has motion restrictions and has strict conditions for movement. In a two dimensional (2D global map), the robot configuration (x,y,θ) has a unique value in each node in the search space. (x,y) represents the robot coordinates and θ expresses the robot's heading angle. In the holonomic robot case, where the robot's heading angle is not a concern, the robot configuration can be represented in two dimensions (x,y). In a three dimensional global map (3D global map) the robot configuration uses up to 6 dimensions in the configuration space which are (x,y,z) positions as well as roll, pitch and yaw.

The robot model links the robot's variables through a set of equations. For example, the differential robot kinematics model is defined as:

$$\begin{aligned} \dot{x}(t) &= v_l(t)\cos\theta(t) + v_r(t)\cos\theta(t) \\ \dot{y}(t) &= v_r(t)\sin\theta(t) + v_l(t)\sin\theta(t) \\ \dot{\theta}(t) &= \frac{1}{2l}v_r(t) - \frac{1}{2l}v_l(t) \end{aligned} \qquad (4.2)$$

where $V_r(t)$ is the right wheel linear speed, $V_l(t)$ is the left wheel linear speed, $w(t)$ is the angular velocity and l is the distance between the two wheels. Thus, the robot position can be controlled by adjusting the average robot speed and the angular velocity.

4.5. Behaviour Module (*B*)

A set of mission strategies is defined allowing the robot to deal with different context conditions; thus, a multi-behaviours robot is needed to accomplish its task. Depending on the mission strategies, the robot must be able to perform different behaviours including conflicting behaviours. In the behaviour module, the issue is finding a suitable behaviour architecture at the behaviour-based level and a suitable behaviour algorithm at the search level to interact with the entire system. In order for a multi-behaviour robot to accomplish its goals, the priorities of the behaviours must change with time according to the robot's situation and surrounding area conditions. The behaviour design will then allow the behaviour selector to select and to actualize suitable behaviours that can be integrated to fit the overall goal of different control objectives (Mai & Janschek 2012).

The behaviour module consists of a set of deliberative and a set of reactive behaviours. The reactive control approach can be represented by a behaviour-based system which quickly responds to inputs sensor reading but does not plan for the future. Another approach such as “deliberative control” considers the problem more globally. This approach uses many heuristic search techniques.

- Deliberative behaviours deal with global entities and solve the global searches depending on a cost functions. Metaheuristic methods are employed to implement global multi-behaviour search (see chapter 5 for more detail).

- Reactive behaviours are represented by a behaviour-based system to deal with onboard sensors data and to perform the local paths. In our research, the behaviour-based system allows the robot wide application-specific behaviours where each behaviour is concerned with a sole objective.

4.6. Navigation Behaviour Control (*β* function)

To navigate in multiple classes’ environment, navigation goals should be specified for each class that may be guided by different patterns of behaviour for different purposes. In a large-scale world model and multi-water pollutant patches, multi-behaviour navigation is needed to comply with different collection strategies regarding the mission objective. A path planning should pass sequentially through a set of patches complying with the mission objective. Some water pollutant patches have a special collection procedure. Thus, a behaviour selector is needed to change the screech behaviour regarding the water pollutant patches.

Navigation Behaviour control has 3 basic duties:

- Regulate and supervise the deliberative-reactive navigation
- At the deliberative level, it takes responsibility for choosing and for changing the optimization method as well as the search method.
- At the reactive level, it is involved in the dynamic and unreachable local path.

This control should have 3 levels which are:

- behaviour navigation indicator

- Intelligent technique for deliberative level
- Intelligent technique for reactive level

Behaviour navigation indicator represents an arbitrator between the two navigation levels which can be represented by different hierarchy intelligent levels depending on the mission complexity. This navigation controller should be equipped with a high intelligence level to solve multi-behaviours which includes conflicting behaviours Intelligent technique for deliberative level involves controlling and activating, for example, a search method that would employ the power of global and local search to realize efficient path planning in a complex environment.

Intelligent technique for the reactive level can be represented by Geno fuzzy and fuzzy hierarchy systems which are efficient systems in deciding on the appropriate behaviour or on a suitable search strategy depending on the context extracted from the world model and the surrounding area's conditions which are detected by the onboard mobile robot sensors.

Figure 4.3 illustrates a navigation behaviour control that controls the global navigation by applying GA for global path planning while employing ACO for local path and sub-global path re-planning.

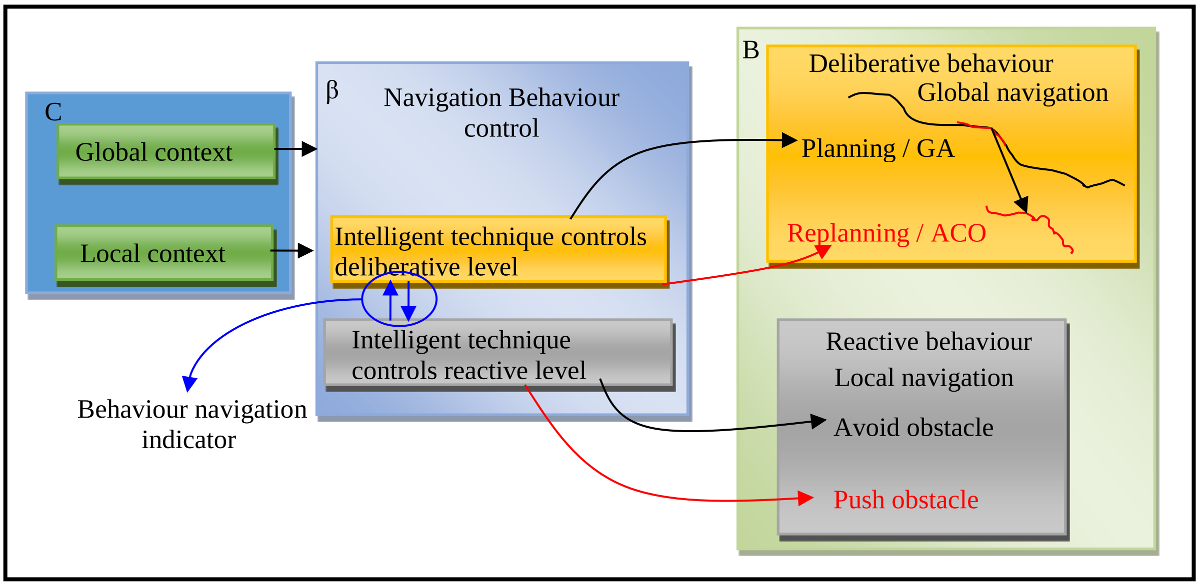


Figure 4. 3. Navigation Behaviour control for deliberative and reactive navigation

At the reactive level, an adaptive expert system or a geno-fuzzy system can be employed to generate new rules (new behaviours) allowing the robot to pass a critical situation.

This function is responsible for activating and deactivating the path planning and reactive behaviour in the critical situations such as a dead-end zone or cycling mode.

4.7. Path Planning (γ Function)

Deliberative control employs a multi-layer map to generate a path planning using many heuristic algorithms. Multi-strategy is needed to adaptively guide the trajectory of a mobile platform to deal sophisticatedly with its environment and perform multi-task missions. For path planning, see Chapter 5 for more details.

4.8. Plan (*P*)

The plan is an ordered set of waypoints. Each waypoint corresponds to an environment feature or landmark and connections represent paths or motion instructions between them. The route (path) goes through the nodes. Thus, the issue is that of finding an optimal route from one node to another (Busquets-Font 2003).

4.9. Trajectory Generation (τ)

Depending on the executed behaviour and the kinematic model of the mobile robot, a trajectory is produced which may comply with the local path or generate a different local path trajectory depending on the status of the local path.

4.9.1. Local and Global Frame

First some definitions are provided:

1) Global reference frame {W} of the plan represents the world model and it is defined by two axes (Yg, Xg) with an origin, O ν.

2) Any existing object in the global reference frame such as a robot or moving obstacle can be attached as a local reference frame. The local reference frame of the robot is defined by two axes (X_R, Y_R), where P is a point on the robot chassis as its position reference point and the origin of the local frame. The robot position in the global frame is three dimensional and is specified by the coordinates X, Y and θ (the robot heading angle). The initial position of the robot is given by $R_{(t0)} = (X_{R0}, Y_{R0})$.

3) The robot's final goal is a point G with coordinates (x_{tar}, y_{tar}) in {W}. The coordinates of this point are known to the robot as ∀t.

Figure 4.4 shows the robot's coordinates. That is the global reference frame to specify the position of the robot by choosing a point P on the robot chassis as its reference position. Thus, the robot's local frame P has target coordinates (x_{tar}, y_{tar}).

The angle α is the robot's target direction obtained from the following relation:

$\alpha = \beta - \theta$

Where: α is the heading direction, representing the correction angle to orient the robot toward the target

β – robot's target direction

θ - robot's heading angle.

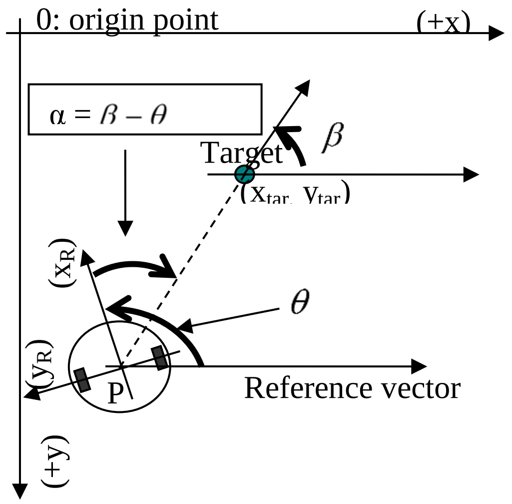


Figure 4. 4. Differential robot in its environment (Halal & Dumitrache 2006)

4.10. Multi-behaviour Deliberative Navigation

The model (Eq.4.3) for deliberative navigation takes the following form:

$$DC = \{E, \alpha, C, \beta, B, \gamma, P\} \qquad (4.3)$$

The deliberative navigation is employed to generate path planning which complies with the mission objective. Figure 4.5 illustrates the deliberative navigation.

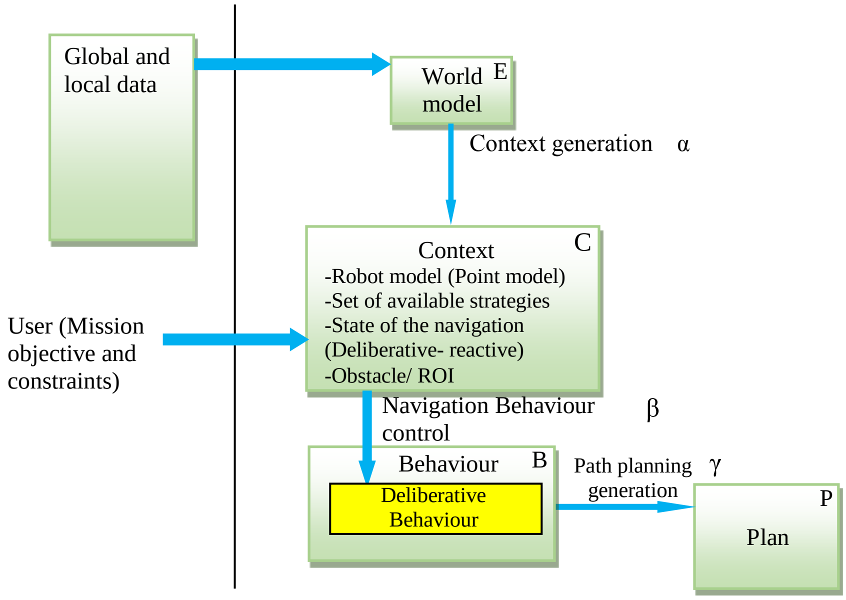


Figure 4. 5. Deliberative navigation

4.10.1. Hybrid Metaheuristic Global Search

A variety of heuristic methods have been investigated that can be considered for the implementation of the gamma function. Evolutionary algorithms have been employed in many variants, such as (Kponyo et al. 2014) who presented an ant colony optimization system to solve the problem of designing an optimal trajectory to improve the traffic situation in an urban environment. (Luo et al. 2013) proposed hybrid genetic algorithm with D* algorithm for real-time map building and navigation for multiple goals purpose.(Yoshikawa & Terai 2009) proposed car navigation system using hybrid genetic algorithms and D algorithm. This system enables the finding of a route which has several passing points before arriving at the final destination. (Li et al. 2014) solve the path planning problem for a submarine navigation application using the artificial bee colony algorithm. (Yu & Cai 2009) deal with a cultural hybrid algorithm to solve the mission planning meanwhile (Gao & Tian 2007) employed an improved simulated annealing artificial network to plan the path planning for a mobile robot. (Ciornei & Kyriakides 2012), developed a hybrid metaheuristic method called GAAPI which is a hybridization between ant colony optimization for continuous domains entitled API and a genetic algorithm (GA). This method has been used in solving different classes of complex global continuous optimization problems.

A Robot accomplishes a deliberative navigation in two steps: trajectory planning and tracking phases. The planning process should comply with the mission objective which can be evaluated by performance criteria such as distance, travel-time, energy consumption, sensing time etc. and should respect a certain number of constraints (geometric, kinematic and/or dynamic) (Haddad, 2007). This approach provides a safe path planning which divides the mission into a set of sequential tasks. Path planning deals with the waypoints optimization using global and local search to apply multi-behaviour navigation.

4.10.2. Hybrid GA Procedure for Path Planning

In hybrid procedures employing a combination of GA and other metaheuristic local search methods, usually GA conducts the global search by supervising the other search methods. GA was employed to perform path planning optimisation in (Halal & Zaremba 2015b). Ant colony optimization was used to conduct the global search. (Colmenares et al. 2014). The results of these two experiments gave reason to propose a novel hybrid search system which displays a better performance than the classical search approach.

A novel system is proposed employing GA and many other metaheuristic local search methods where GA conduct the global search having a supervisory role on the different search methods.

A search technique for hybrid GA is developed to navigate in a large-scale environment. This technique consists of generating an initial population of adaptive solutions which can be done either randomly or heuristically. In each generation, the fitness function evaluates the generation in terms of solution quality and provides a numeric value according to its performance. Mutation and crossover generations are generated in a parallel way into two generations which are generated by the elite individuals for the first population and random individuals have been selected to generate the other population.

Hybrid GA has multiple, independent populations which are generated in a parallel way. The new populations are created by different metaheuristic methods which are integrated with the GA system. Each population evolves using different search strategies. Figure 4.6 illustrates the general framework of the proposed hybrid search architecture.

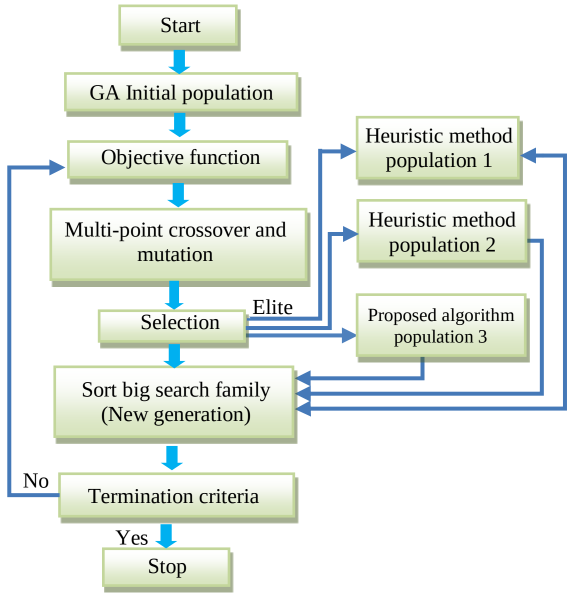


Figure 4. 6. General framework of the proposed hybrid search architecture

The next generation is formed by a random selection process using a higher probability for chromosomes with higher fitness values. This process prevents low-fitness individuals from the current generation to move on to the next generation.

Next generations can be produced either synchronously, so that the old generation is completely replaced, or asynchronously, where the generations overlap (Abu-Dakka et al. 2012).

4.10.3. Ant Colony Local Path Navigation

The ant colony system has been proven to be an efficient method for the local search. Ant colony local search procedures have used best neighbour and random neighbour to improve the vicinity. The objective of this search is to find the best replacement for each neighbour in the path planning node. The replacement node can be chosen randomly or heuristically. After selecting a new neighbour, one of following procedures is applied:

- best_neighbour uses (4.4) to select the best option among the neighbours.

$$j = \arg\max\{(\tau_{iu})(\eta_{iu})\}^{\beta},\ for\ u \notin M_k \qquad (4.4)$$

- random_neighbour computes the probability values of each neighbour according to (4.5), and the selection is made with a Roulette Wheel procedure.

$$p_{ij} = \frac{(\tau_{ij})(\eta_{ij})^{\beta}}{\sum_{u \notin M_k}(\tau_{iu})(\eta_{iu})^{\beta}}, if \;\; j \notin M_k, otherwise \;\; 0 \qquad (4.5)$$

where τ_{ij} is the current pheromone trace in the arc *ij*; η_{ij} is the heuristic value of the arc *ij*. To avoid the repetition of a location in the route, each ant stores the location of the visited nodes in a temporal memory M_k. The pheromone update process is done in two phases; first, each ant updates its own path and later a global process updates the arcs of the best route according to (4.6) and (4.7), respectively.

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \rho\tau_0 \qquad (4.6)$$

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot\Delta\tau_{ij}(t) \qquad (4.7)$$

4.10.4. Hybrid GAACO for Deliberative Navigation

The main idea is to build a hybrid system of two meta-heuristics that are genetic algorithm and ant colony optimization to profit from the advantages of both systems and to overcome their shortcomings.

The novel GAACO proposed system has the ability to deal with a dynamic environment by applying efficient local and global search procedures. Ant Colony Optimization (ACO), local search algorithm and many other metaheuristic systems improve the local search by examining the neighbours of elite solutions. The GA system examines unvisited regions and generates solutions that avoid being stuck in a local optimum. The big family pool which consists of all old-generation solutions and current-generation offspring obtained after mutations and crossover operations combined with ACO solutions give the system the ability to save the elite searching experience from one population to the next. Figure 4.7 shows the GAACO system proposed for global search.

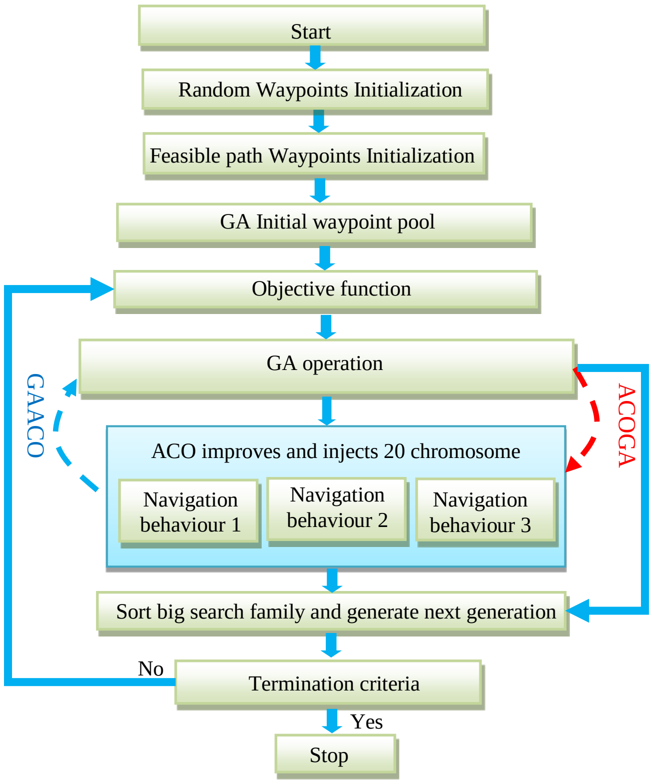


Figure 4. 7. GAACO proposed system for global search (Halal & Zaremba 2015a)

Figure 4.8 shows a hybrid GAACO architecture-based mission planning system. The mission objective is defined and accompanied with a strategy definition to achieve the mission goal. A multi-layer maps is generated to interpret the global environment and to weigh the importance of different environmental features to the acquisition strategies. Depending on the mission objective, the context components are modelled. A set of ROIs is generated to guide the search toward specific patches associated with their acquisition strategies. Metaheuristic methods perform the search for optimal solutions in parallel, or each method performs a specific search behaviour in a receptive patch.

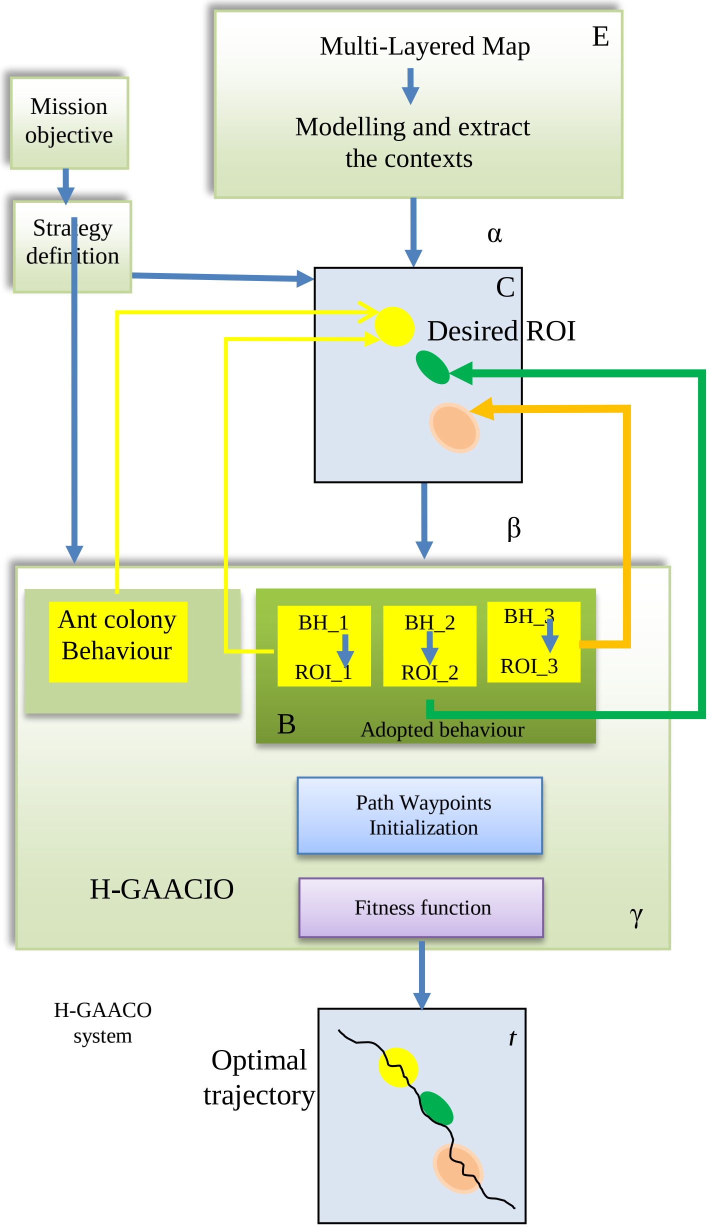


Figure 4. 8. Hybrid genetic algorithm for multi behaviour deliberative navigation

4.11. Integrated Architecture for Local Path Problems

Navigation in dynamic and unreachable local paths is investigated. During the execution of the reactive level, Navigation behaviour control involves controlling 3 robot statuses which are as follows:

4.11.1. Static Local Path

When the robot detects no obstacle, the task will be executed exactly as the deliberative plan. Figure 4.9 shows traditional hybrid deliberative navigation where reactive navigation executes the plan of deliberative navigation. The navigation procedure is as follows:

- Sense the environment
- Interpret the surrounding area
- Plan the mission
- Execute the local paths
- Get to the destination

Figure 4.9 illustrates the flow chart of the navigation process.

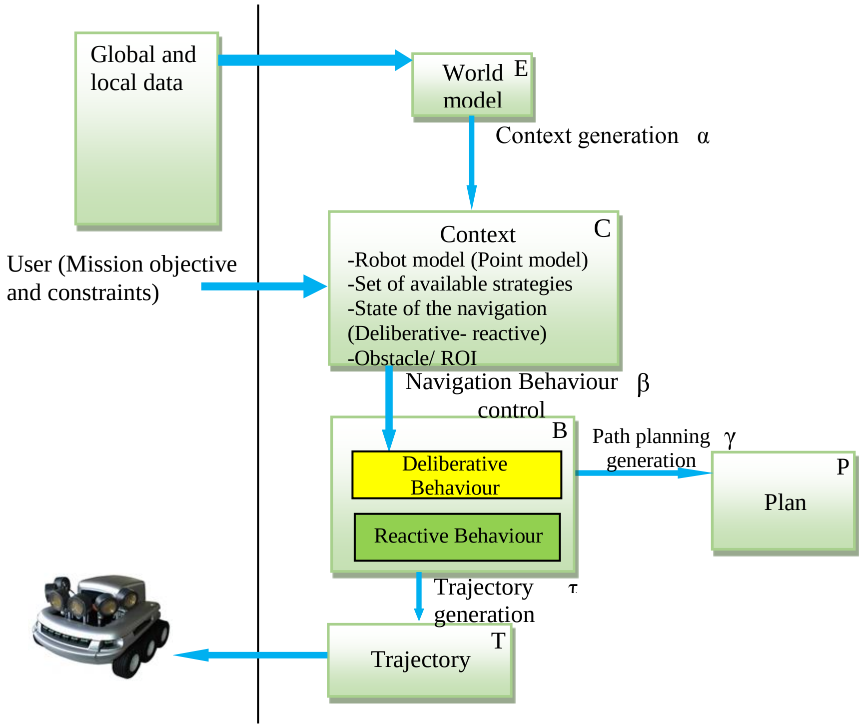


Figure 4. 9. Traditional deliberative reactive navigation

4.11.2. Dynamic Local Path

A dynamic local path occurs when new environmental changes start producing new surrounding area conditions such as fog or haze which impact the local mission. In the sequential hybrid navigation, the robot shows bad performance regarding the deliberative task. Applying different strategy behaviours at the reactive level allows it to reach its target. Reactive navigation solves the issues depending on the robot's vicinity information which is captured by the robot's onboard sensors.

In the novel architecture, Navigation behaviour control will be involved in the appropriate reactive behaviour decisions as quick solutions to prevent the robot from facing any collisions. The decision will be made depending on the local and wider range information from the global data.

When the changes don't require a quick decision and local path replanning is a suitable solution for this situation, the navigation behaviour control exchanges the responsibility between the reactive and deliberative in an "integrated" synergetic way to execute the sub-mission safely and intelligently. Both navigation levels cooperate to execute the local mission. The reactive controller changes its behaviour while the path planning generation searches for the optimal local path.

The navigation procedure in this situation is as follows:

- Sense the environment
- Interpret the surrounding area
- Plan the mission
- Execute the local paths
    - If the local information doesn't match the information in the global data
        - The reactive level changes its behaviour depending on the local and global information
        - While the reactive level executes the local path by changing its behaviours, the deliberative level searches for an optimal local path if the replanning is suitable for the respective situation
- Get to the destination

A flowchart of the process is illustrated in Fig. 4.10

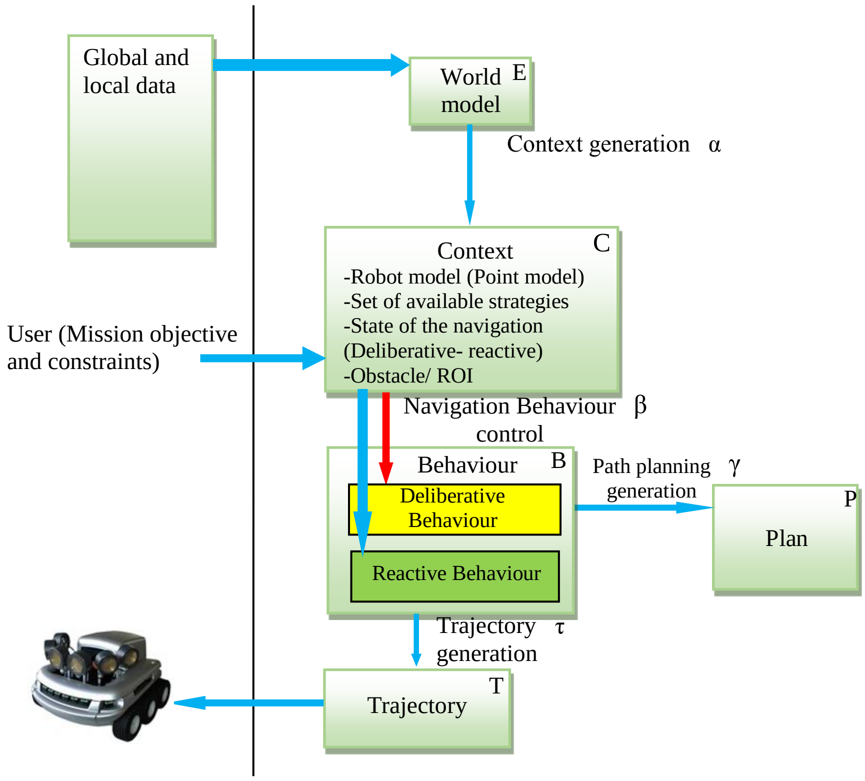


Figure 4. 10. Dynamic local path navigation

4.11.3. Unreachable Sub-Local Task

Navigation behaviour control is introduced to change the execution plan when the sub-task can't be executed in a specific time due to the surrounding area information where local data does not match the information in the world model.

To solve this situation one of the following procedures will be executed:

- Sub-global path replanning

A new planning strategy can be used to solve the robot situation and deal with environmental changes. The result is that a new sub-global path deals with the new conditions. Reactive navigation will keep trying to execute the local mission while the replanning process develops the new sub-global path.

- Activating the conflicting behaviour

In many critical cases, the robot is unable to reach its local target and so the robot cannot reach its destination; replanning is unfeasible and so the behaviour selector is involved in activating the conflicting behaviour (aggressive behaviour).

The navigation procedure in this situation is as follows:

- Sense the environment
- Interpret the surrounding area
- Plan the mission
- Execute local paths
    - If the local path is unreachable
        - While the reactive level tries to get to the local destination, the path generator searches for the optimal sub-global path
        - The behaviour selector activates the conflicting reactive behaviour due to a robot critical situation, and so replanning is not feasible
- Get to the destination

A flowchart of the process is illustrated in Fig. 4.11

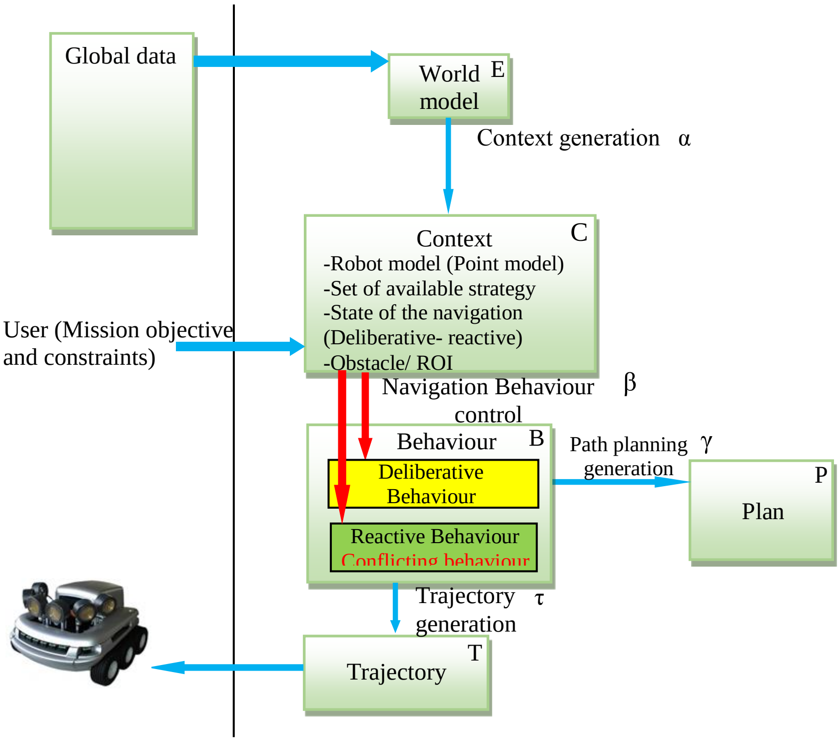


Figure 4. 11. Unreachable local path navigation

4.12. Conclusions

A general architecture was presented, flexible and complete enough to guide the design of a multi-behaviour deliberative-reactive navigation system in a complex and unstructured environment. The main feature of the proposed architecture is a close integration of the deliberative and reactive navigation, and the transitional type of relationships between its components. A comprehensive model of environmental representation is provided, which consists of multi-dimensional data and multiple scales. Contexts are extracted using a variety of methods to properly interpret the global environment with respect to the mission objective. The navigation behaviour controller involves deciding which navigation level should be applied level. Path planning generation based on the hybridization of the different metaheuristic methods has the ability to optimize a path planning for large scale and partly unknown environments. This system applies different patterns of behaviour for different purposes regarding the mission objective as predefined by the user. A range of designs involving computational intelligence techniques can perform extracting the context, generating the path planning, and selecting the behaviour.

The flexibility of the architecture was demonstrated through the examples of hybrid path planning, and multi-behaviour operation in the situation of dynamic and unreachable local paths. The integrated deliberative-reactive navigation control improves the quality of both navigation levels for complex and dynamic environments, where the mobile robot performs multi-task navigation subject to different behaviours. This integrated system provides an efficient framework for designing solutions for navigation in dynamic environments with the ability of solving critical situations of unreachable local paths.

Chapter 5

Deliberative Multi-Behaviours Navigation for Environment Monitoring

5.1. Problem Statement

The algal blooming in inland lakes and coastal waters has become a critically important issue for its impacts not only on local natural and social environments but also on the global human community. Authorities responsible for water quality environmental protection, economic development and public health must develop and implement plans and strategies for prediction and mitigation of the effects of algal blooms (Harmful Algal Blooms - HAB). This requires means to detect and monitor the occurrence of the blooms. Modelling of the underlying phenomena that lead to HAB is complex and is a subject of ongoing research (Duan et al. 2009). Detection of the concentration of algae in waters basins is based on the assessment of the concentration of chlorophyll-a (chl-a). Aside from the detection of high levels of chl-a concentration, water quality monitoring includes the detection of other water pollutants, such as Total Suspended Sediment (TSS) and Dissolvent Organic Carbon (DOC). Remote detection techniques provide significant advantages in the detection of water pollutants over ground-based monitoring in terms of spatial and temporal coverage and cost-efficiency. In (Vincent et al. 2004) a set of algorithms was developed to derive phycocyanin, Chlorophyll-a, and sediment for the detecting of blue-green algal blooms in Lake Erie based on Landsat ETM images; A model to quantify chlorophyll-a in Lake Balaton using Landsat ETM imagery was discussed in (Tyler et al. 2006). The existing algorithms depend on water quality and RS sensors. The core approach to the detection has been automatic analysis of multi-spectral image sequences, mostly from MODIS and MERIS sensors.

Acquisition of the reference data is usually a costly and time-consuming process. In the application area addressed by this project, it implies the carrying out of data collection by a specially equipped mobile platform, such as a cruise ship, a glider or a floating robot. Our study uses information obtained from in situ measurements performed for Lake Winnipeg in Canada by a ship equipped with scientific instrumentation.

Critical to this research are reliable, efficient, and adaptive control strategies that ensure mobile sensor platforms collect data of the greatest value. In addition, the large size of the lake, the tenth largest lake in the world, and the use of a large ship for data acquisition missions make the development of optimal navigation control important.

The problem addressed in this thesis consists in the trajectory planning for precise acquisition of water pollutants by a mobile platform, when the planning process is guided by prior rudimentary information about the distribution of pollutants obtained from remote sensing data, and should incorporate different acquisition strategies.

The sample acquisition mission is performed within a more general procedure consisting of the following phases:

1) Determination of the type of water regions and types, sample location zones, and water pollutants to be sampled;
2) Identification of the pollutant detection indices (e.g., maximum chlorophyll index (MCI), Fluorescent Line Height (FLH)), coverage methods (e.g., uniform coverage, maximum concentration gradient) and the number of samples;
3) Selection of the sources of remote sensing data and their calibration methods;
4) Selection of the ancillary data from in situ sensors needed to determine the factors affecting the pollutant distribution dynamics (e.g. wind, temperature);
5) Determination of the acquisition mission parameters (e.g., total mission time, sampling methodology).

Most of the above factors and conditions affect the strategies that have to be incorporated in the planning procedure. Mission strategies can be classified into two categories:

(1) Water pollutant concentration strategies

In these classes of strategies, the aquatic acquisition platform collects the most valuable samples from different pollutant classes and their combinations, such as

- Chl-a
- Chl-a & TSS,
- Chl-a & DOC
- Chl-a & TSS & DOC.

In this class of strategies, specific samples should be collected while neglecting the other samples within a certain time window.

(2) Local coverage strategies:

In this mode, the ship executes a specific navigation and collection behaviour depending on the shape of the sample spatial distribution. We distinguish here such sampling strategies as the uniform coverage of high-concentration areas, sampling at local concentration maxima, and sampling along maximum gradient lines which is of interest in many environmental monitoring applications (Zhang & Leonard 2005). The sampling process can be different in each patch to comply with the general and local mission goal.

Both types of strategies are executed under some specific constraints, such as time and distance constraints. Time window constraints can be imposed on certain pollutant patches which can minimize the travel distance constraint on other patches. Also, a certain number of samples have to be collected in a specific patch before heading to another one.

5.2. World Model

Satellite images have successfully been used to navigate and observe natural phenomena, such as lakes and oceans. The satellite systems used for inland water monitoring are mostly medium resolution imaging instruments, such as the NASA MODIS (Moderate Resolution Imaging Spectroradiometer) sensor (see Appendix 1.1) or the MERIS (Medium Resolution Imaging Spectrometer) sensor carried aboard the ESA's Envisat satellite. (See Appendix 1.2). The multi-spectral data can subsequently be used to obtain models of water pollutants, such as the concentration of chlorophyll or suspended sediments (Koponen et al. 2005), by applying such measures as the maximum chlorophyll index (MCI) (Gower et al. 2008) or the ocean chlorophyll 4 algorithm

(OC4v4), (O’Reilly et al. 1998) . The remote sensing data often have to be augmented and updated by *in situ* measurements due to the need for precise local measurements, for the calibration of satellite imagery in varying water conditions, and for the purpose of precise local decision making.

5.2.1 Parametric Model

The modality issue has been successfully resolved in this thesis by introducing an additional stage of processing of remote sensing data, which is the classification of water characteristics using the methods of band combinations and the regression analysis.

Estimation of the chlorophyll concentration is typically obtained by using indices that exploit chlorophyll absorption/reflectance wavelengths (Topliss & Piatt 1986).

MODIS and MERIS sensors have spectral bands in the range of 665-750 nm. These bands have been used to determine the radiance baseline for comparison with florescence measurements near 680 nm, the interpolated radiance near 680 nm is subtracted from the observed radiance at this wavelength to give a measure of florescence line height (FLH). The FLH algorithm was developed for the MODIS satellite. This algorithm returns normalized fluorescence line height in, calculated as the difference between the observed radiance at (676 nm) and a linearly baseline defined by radiance at (665 nm) and (746 nm), as illustrated in Fig. 5.1.

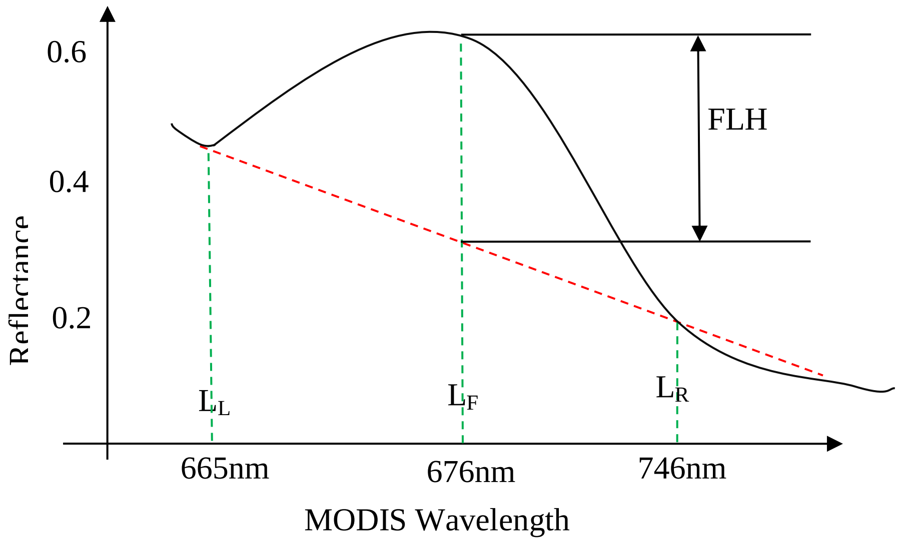


Figure 5. 1. FLH index for MODIS

The subsequent calculation of chlorophyll concentration can be performed using the FLH index or a similar Maximum Chlorophyll Index (MCI) index developed for MERIS. These indices are based on the following equation:

$$L_{MCI} = \left(\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}\right)(L_1 - L_3) + L_3 \qquad (5.1)$$

where λ_1, λ_2 and λ_3 denote three subsequent wavelengths, and L_1, L_2 and L_3 are the corresponding radiance values. The pertinent wavelengths are given in Table 6.

Table 6: MODIS and MIRES (MCI and FLH)

	MERIS		**MODIS**
	MCI	FLH	FLH
λ_1	685	665	667
λ_2	709	685	678
λ_3	753	709	748

5.2.2 Pollution Indices

- **Detection of Total Suspended Solids (TSS)**

There is currently no uniform remote sensing model to estimate TSS, since in practice clear and turbid waters are often combined, and TSS size variations affect the choice of the most appropriate wavelength. Many models have been proposed based on the combination of MERIS red and near-infrared bands. For each water class, a separate model was obtained and used for the assessment of the water pollutant. Equation (5.2) has been used to measure TSS (Koponen et al. 2005):

$$TSS = 53.7\left[\frac{L_{709}}{L_{560} - L_{665}}\right] - 17.0 \qquad (5.2)$$

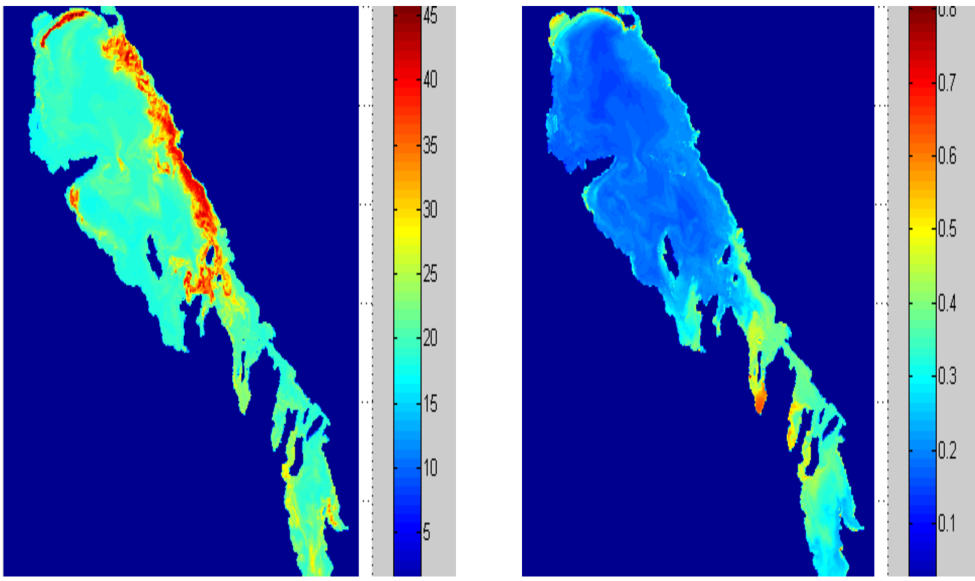
where L_{709}, L_{560} *and* L_{665} denote the wavelength of 709 nm, 560 nm and 665 nm respectively.

- **Detection of maximum chlorophyll index (MCI)**

The MCI index offers good performance especially for concentrations between 10 and 50 μg/l. Equation (5.3) represents a MCI calculation based on (Gower et al. 2008).

$$\mathrm{MCI} = \mathrm{L}_{709} - \mathrm{L}_{681} - 0.389\,(\mathrm{L}_{753} - \mathrm{L}_{681}) \qquad (5.3)$$

The factor 0.389 is calculated as the wavelength ratio (709–681) / (753–681). An example of the distribution of chlorophyll-a and TSS in Lake Winnipeg as shown in Fig. 5.2.



A) B)

Figure 5. 2. A) MCI map and B) TSS map for Lake Winnipeg

5.2.3 Multi-Layer Maps

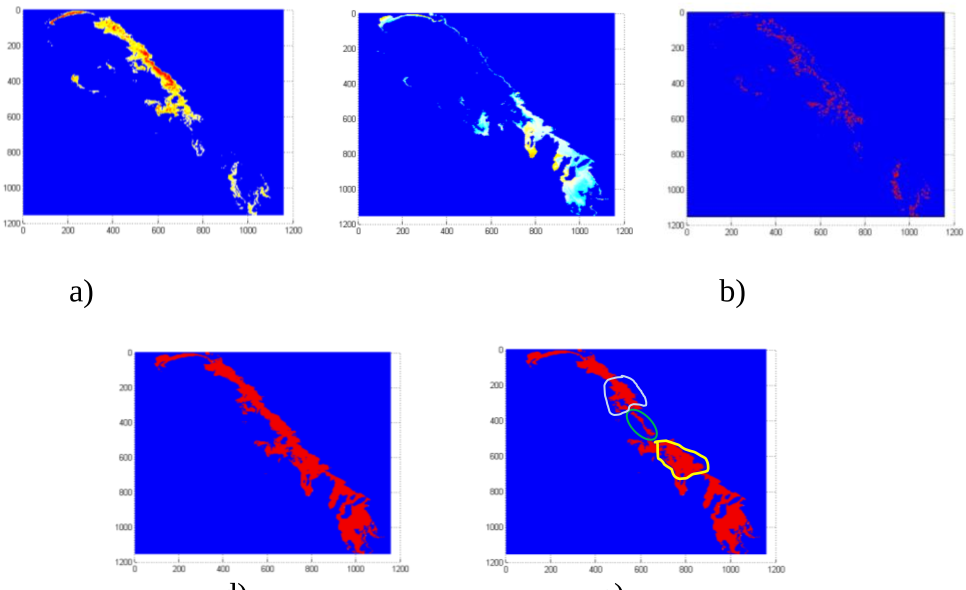
The maps consist of the measured pollutants layers (Chlorophyll-a, Total Suspended Sediment (TSS), a bathymetric data layer, and meteorological data layers. The issue of different spatial scales arises in reference to the resolution of measurement sensors and sampling fields as well as to the sampling strategy. In terms of sampling strategies, feature-tracking strategies, such as gradient climbing strategies, are particularly useful for sampling at relatively small spatial scales. Strategies that provide synoptic coverage are best suited for larger spatial scales.

5.3. Context

With respect to the types of pollutants, the RS data have to be pre-classified. The path planning maximizes the value of the collected samples along its trajectory where it traverses regions of different distributions of the pollutant concentration. As a result, the planning algorithm works on many maps created to represent different concentration levels for different water pollutant classes. The optimal strategy directs the ship to the best Region of Interest (ROI) zone. The samples values (weights) vary depending on the mission objective.

5.3.1. ROI Approach

The ROI approach was used to identify the study zones and their boundaries (Park & Cho 2013). ROI maps guide the multi-strategies sampling to orient the acquisition platform toward the valuable samples in the ROI and selects the suitable samples using the penalty/award mechanism. Figures 5.3 a) b) and c) show regions of interest for MCI, TSS and the maximum gradient of the chlorophyll concentration. The regions are defined as the concentration of TSS bigger than 0.3 from the normalized TSS model, and the concentration of chlorophyll-a bigger than 0.5 from the MCI normalized model. Fig. 5.3 d) represents the overall ROI formed from the MCI and TSS zones. Fig. 5.3 e) illustrates three ROI zones, which are MCI, TSS and maximum gradient chlorophyll concentration, used in the experiments.



a) b) c)

d) e)

Figure 5. 3. a) Chl-a ROI (MCI > 0.5); b) TSS ROI (TSS > 0.3); c) Chl-a Max Gradient ROI; d) Combined Chl-a & TSS regions of interest, and e) Combined Chl-a & TSS & MG regions of interest,

5.3.2. Multi-Behaviour Operation

The basic idea of the multi-strategy GA-based path planning is that the acquisition platform explores water pollutant patches using different behavioural characteristics depending on the sampling requirements in each patch. The behaviours affect the local search optimization where the bes-evaluated neighbour is selected according to the adopted behaviour. The following behaviours represent different sampling strategies.

Behaviour 1- Short local path and high sample values.

The sampling process selects the best sample values defined as in (5.4)

$$SV = \left[1 - \frac{local\ path_{i\,j}}{\mathrm{Max local\,path}}\right] * V_{chlo\,j} \qquad (5.4)$$

where i is the departing waypoint, j is the destination waypoint, and $V_{chlo\ j}$= $Chl_{x,y}$ is the chlorophyll concentrations in the cell (x,y) of the MCI layer

Behaviour 2 - Maximum gradient (MG) sampling.

Valuable samples (bigger than a given threshold number) are selected along a short local path according to the following equation:

$$SV = \left[1 - \frac{local\ path_{i\,j}}{\text{Maxlocal path}}\right] * V_{MG} \qquad (5.5)$$

The sampling behaviour for other samples maximizes the local path according to equation (5.6):

$$SV = \left[\frac{local\ path_{i\,j}}{\text{Maxlocal path}}\right] * V_{MG} \qquad (5.6)$$

Behaviour 3 – Multiple pollutant patches.

The sampling procedure selects the best sample value respecting equation (5.7), Where the value sample has the maximum local path range distance and the highest sample weight;

$$SV = \beta\left[\frac{local\ path_{i\,j}}{\text{Maxlocal path}}\right] * V_{chlo\,j} * V_{TSS\,j} \qquad (5.7)$$

Where $V_{chlo\ j}$ and $V_{TSS\ j}$= $Chl_{x,y}$ and $TSS_{\ x,y}$ are the chlorophyll and TSS concentrations in cell *x,y*, taken from the MCI and TSS maps.

Behaviour 4:

The sampling procedure selects the best sample value respecting equation (5.8), where the sample value has the maximum local path range distance and the highest sample weight;

$$SV = \left[\frac{local\ path_{i\,j}}{\text{Maxlocal path}}\right] * V_{TSS\,j} \qquad (5.8)$$

$V_{TSS\ j}$= $TSS_{\ x,y}$ TSS concentrations in respective cells were taken from the TSS map.

An example of water pollutant patches obtained for different behaviours from a 3-layer map (MCI, TSS and MG) map is shown in Fig. 5.4.

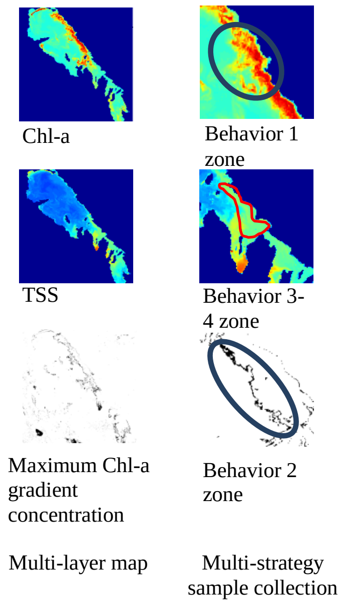


Figure 5. 4. Multi-behaviour navigation in water pollutant patches

5.4. Path Planning Generation

5.4.1 Genetic Algorithm Architecture

The basic operation of the proposed GA-based path planning procedure can be summarized as follows: The sampling points correspond to the waypoints of the global path of the mobile platform. Thus, the global path consists of several local paths, which are the arcs between two waypoints with a directed connection between them. The random waypoint approach is applied using many search navigation strategies to generate a set of global paths. The initial population of waypoints is pruned to generate collision-free path and subsequently stored in the initial chromosome pool population. Unfeasible solutions are deleted. Fig. 5.5 illustrates the developed genetic algorithm based path planning procedure.

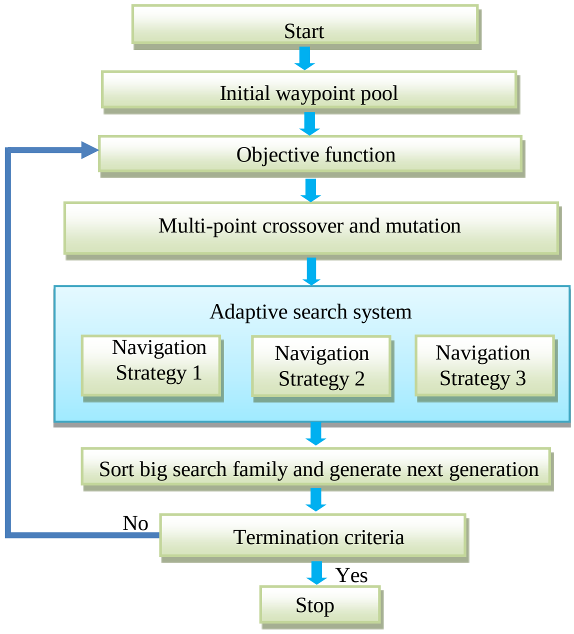


Figure 5. 5. Genetic Algorithm based path planning.

The adaptive search (AS) system improves the elite path (the best ten solutions) and returns efficient paths adapted to the local navigation behaviour. The big family pool consists of all old-generation solutions and current-generation offsprings obtained after the mutation and crossover operations combined with AS solutions. It gives the system the ability to save the elite searching experience from one population to the next one (Hsu & Liu 2014). The big family search results are sorted and pruned to form the next generation. Fig. 5.6 shows the diagram of the big family search. A more detailed description of individual steps of the algorithm follows below.

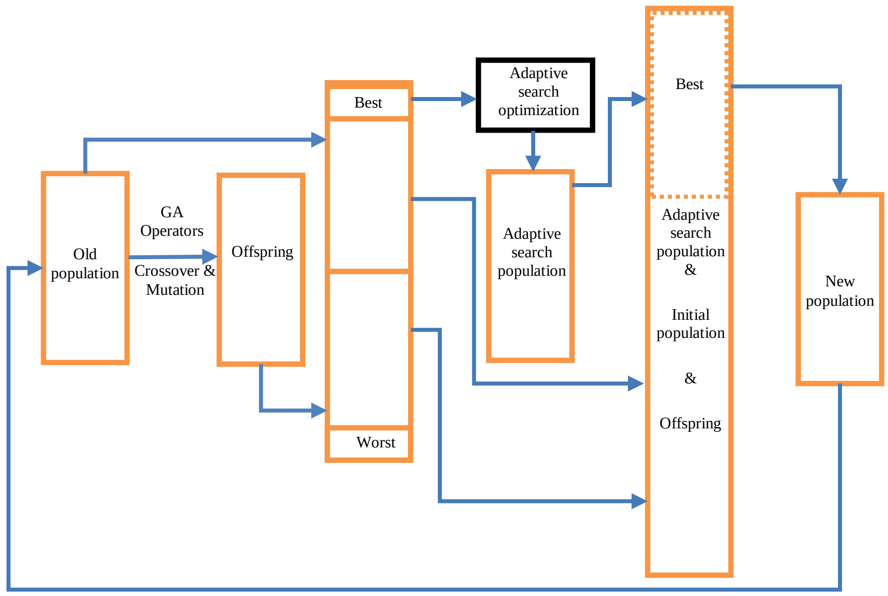


Figure 5. 6. Big family search (Hsu & Liu 2014)

Path planning algorithm is illustrated as follows:

Path-Planning Algorithm:

1. Call multi-layered maps(Environment)
2. Call the initial population
3. Determine the GA operators and their operation rate,
4. Set the Best Objective Value= 0
5. While loop for GA Generation (Termination Condition)
6. Evaluate the initial population using the fitness function (Objective function)
7. Call simple point crossover; the parents are chosen randomly
8. Call multi-point crossover; the parents are the elite paths
9. Call multi-point mutation; the parents are chosen randomly
10. Generate the Crossover and Mutation population randomly and heuristically
11. Generate new population from the old population and the crossover and mutation population
12. Sort the new population and send a copy of the best 20 paths to adaptive search algorithm
13. Call adaptive search algorithm, delete the waypoint outside the ROI and generate new waypoint in the ROI randomly and heuristic
14. Generate the adaptive search population,
15. Generate the big family search population
16. Compute Objective Values for each path in the population
17. Sort Objective Values
18. Set the best Objective Values= Objective Values (1)
19. Set new population size
20. If Objective Values (1) greater than Best Objective Value, Then

21. Set Best Objective Values (1) to Best Objective Value
22. If the Objective Values (1) equal to Best Objective Value for 40 consequent times stop the evaluation
23. Next generation

Figure 5.7 a) represents the flowchart of the proposed systems for multi-behaviour global search and 5.7 b) shows the flowchart of a modified search system combined with ant colony optimization to enhance the local search.

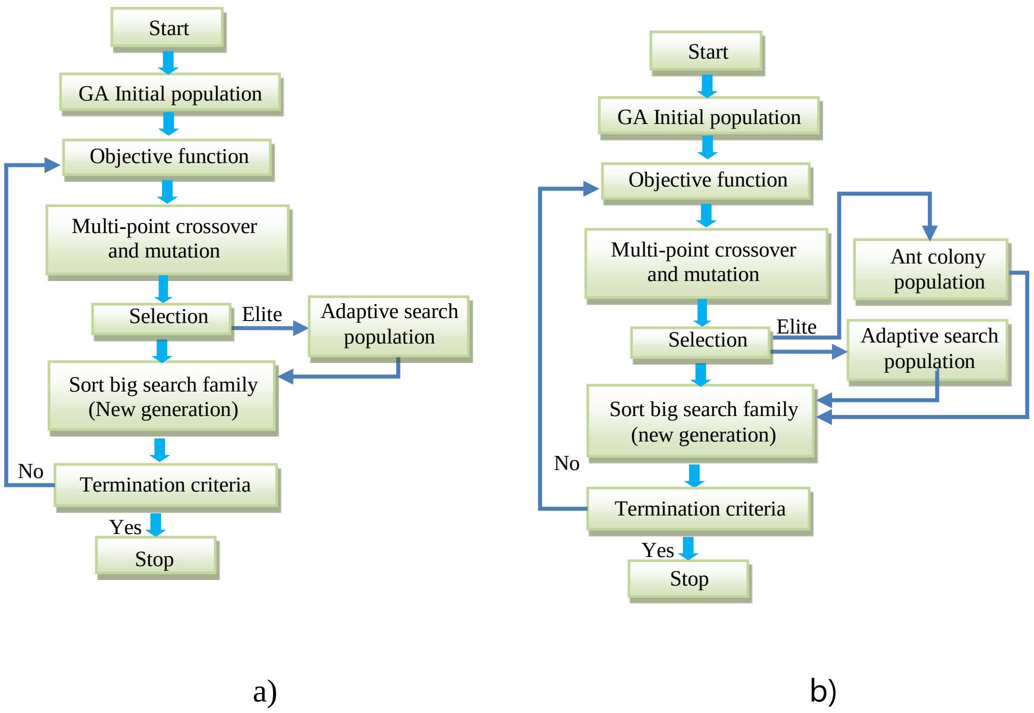


Figure 5. 7. Multi-behaviour navigation for global search

5.4.2 Waypoints

The waypoint technique was used in the GA-based path planning process as a technique appropriate for the large monitoring environment (Veera Ragavan et al. 2011) . Waypoints are usually abstract points (Ibrahim et al. 2009) used to help to define local paths through which a mobile platform can navigate, reach its region-of-interest destination, and collect the water pollutant samples (Park & Cho 2013). In the application discussed, waypoints correspond to sampling points. In order to deal with multiple sampling areas, multi-point crossover (MPC) was implemented. The MPC operator works to build the final solution which consists of valuable segments of local paths from many search strategies. The mutation operator improves the local search and helps the population to avoid local minima. The evolution process optimizes the path planning by designing new chromosomes which consist of best value samples from many global paths.

5.4.3 Path Planning and Initial Waypoint Population

In the GA-based path planning procedure, the population is represented, as in the Vehicle Routing Problem, by ordered sets of waypoints. Each feasible set is considered to be an individual in the population. Each waypoint, which is a sample candidate, represents a location in the environment and is characterized by an identifier in the form of (x,y) coordinates. The initial genotype can be represented by a cell array, where each pair of cells represents the local path length and the heading angle towards sequential waypoints.

The path planning generator works as follows:

1) Determine the first waypoint in the path, i.e., the starting point, with the initial angle equal to zero.

2) While the path planning doesn't reach the desired target, generate a random number of L, the path length, between $L_{\min}$ and $L_{\max}$, and a random heading angle β between $\beta_{\min}$ and $\beta_{\max}$ obtaining the next waypoints (Xiao-Ting et al. 2013). A maximum number of waypoints is given for each search strategy.

3) Different strategies are applied to water pollutant patches by adjusting L and β. Each path planning strategy handles a different number of samples depending on the search path.

4) Continue with another patch or return to the starting point, depending on constraints, such as the maximum travel distance or the maximum number of water samples.

Fig. 5.8 illustrates the path planning generator.

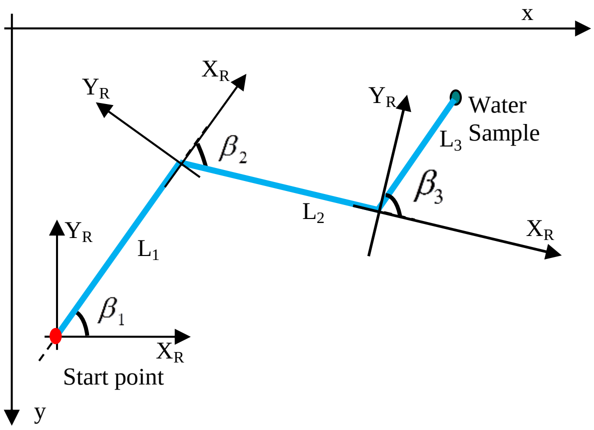


Figure 5. 8. Waypoint generation scheme

The chromosomes are encoded as an integer string. Each gene consists of two variables, the local path length and the heading angle as seen in Fig. 5.9 a). Depending on the start point and the chromosome, the waypoint generation is done as seen in Fig. 5.9 b). The path planning waypoints are represented in the form of a long array as shown in Fig. 5.9 c). The GA search consists in determining the waypoints between the starting point of the mission and the destination point.

a)

Heading angle	travel distance	Heading angle	travel distance	…………….	Heading angle	travel distance

b)

heading	travel distance	previous waypoint (start point)	waypoint$_1$ x coordinate (Latitude)	Waypoint$_1$ y coordinate (Longitude)

c)

Starting point	Node 1	Node 2	..	..	..	…..	Node n	Destination point

Figure 5. 9. Chromosome and waypoint array. a) GA chromosome; b) Waypoint representation; c) Waypoint array.

An obstacle-free path planning algorithm (Zeng et al. 2011) was adopted to deal with the experiment surrounding conditions. It produces a feasible path that satisfies the following conditions:

- Waypoints should be located outside the obstacles.
- Waypoints should be located in the sampling space.
- The local path should not intersect with the obstacles.

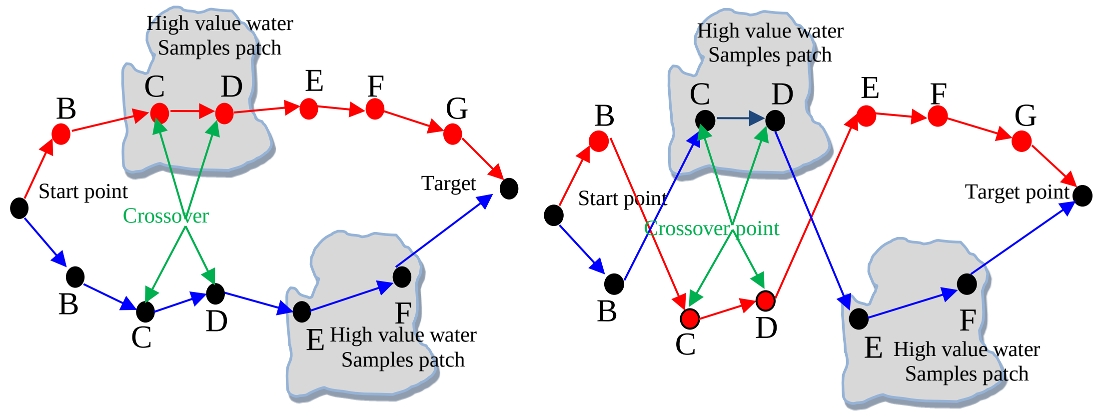
5.4.4 Waypoint Modification

In order to comply with the feasibility constraints and to enhance the efficiency of the path, a certain number of the waypoints in the elite solutions can be modified for each generation by applying three possible operations: waypoint deletion, insertion, or replacement (Châari et al. 2014). Waypoint deletion eliminates all waypoints in the clear water body. The waypoint insertion operation explores the neighbourhood and inserts a new waypoint, according to a predefined behaviour for each water pollutant type, which fits the goal of the sampling strategy. After deleting and inserting the waypoints, the algorithm evaluates the path and, depending on the numeric value which should be under a certain threshold, conducts a neighbourhood search to replace the lowest waypoint value with a new one and builds another feasible path P_n. The path planning should satisfy constraints such as the maximum travel distance and the maximum mission time.

5.4.5 Multi-Point Crossover

Various crossover techniques such as one-point crossover, two-point crossover, multi-point crossover have different advantages. One or more crossover can be applied to generate the offspring generation employing the advantage from each operator. Once the crossover operations are performed, one or more mutation operation are also done to prevent the genetic algorithm from being trapped in local optima. The mutation rate probability should be kept low to avoid delay in convergence with global optima.

Multi-point crossover operates in the global path planning phase and is used to enhance the process of selecting valuable samples located in different zones. The crossover procedure is explained in Fig. 5. 10. Parent chromosomes, P_1 and P_2, are cut at multiple random locations, and the portions of the chromosomes between the cuts are swapped. The result is a pair of offsprings I1 and I2. The crossover is applied on the best-fitness chromosomes chosen from the pool. Due to the difference in the chromosome length, the crossover points should be applied to the shorter chromosome.



a) b)

Figure 5. 10. Multi-point crossover. a) A two-chromosome and two-point crossover. b) Two offsprings.

Figure 5.11 represents the overall architecture of the developed adaptive GA-based mission planning system. The mission objective is defined and accompanied by a strategy definition to achieve the mission goal. A multi-layer map is generated to interpret the global environment and to evaluate the importance (weight) of different water pollutants in the sampling strategy. A set of ROIs is generated to guide the search toward specific patches associated with their acquisition strategies.

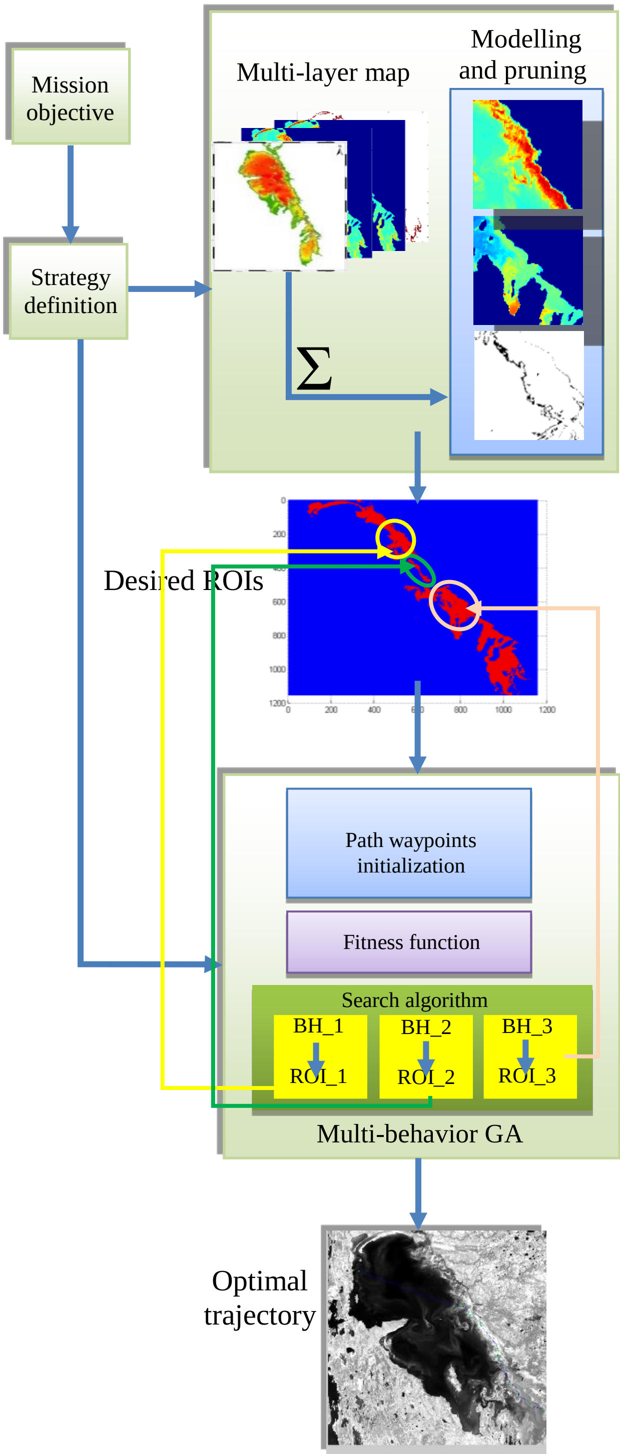


Figure 5. 11. Adaptive GA-based navigation system

5.4.6 Adaptive Global Path Planning

In order to improve the path planning process and make it more adaptable to changing environment conditions, adaptive global path planning is employed to update and to re-plan the sub-global paths whenever new information is provided.

The adaptive path planning follows the concentration of the chlorophyll-a biomass as shown in Fig. 5.12 which is, to a large degree, subject to the wind impact. Table 7 summarizes the wind speed and direction at the time of the simulation. The data is taken from the C45144 buoy which is located in Lake Winnipeg's northern basin. This data is used to simulate the water pollutant patch. The path planning avoids the unforeseen obstacle in (local path 4) to prevent the robot from any collisions. In this local path, the robot doesn't follow the concentration of the chlorophyll-a biomass.

Table 7: Wind Data

Date and Time	Wind Speed	Wind Direction (°N)
17/07/12 08:08	2.3	130
17/07/12 09:08	1.8	144
17/07/12 10:08	1.1	141
17/07/12 11:08	1.4	176
17/07/12 12:08	1.3	195
17/07/12 13:08	1.3	240
17/07/12 14:08	2.5	228
17/07/12 15:08	2.3	220
17/07/12 16:08	2.5	229

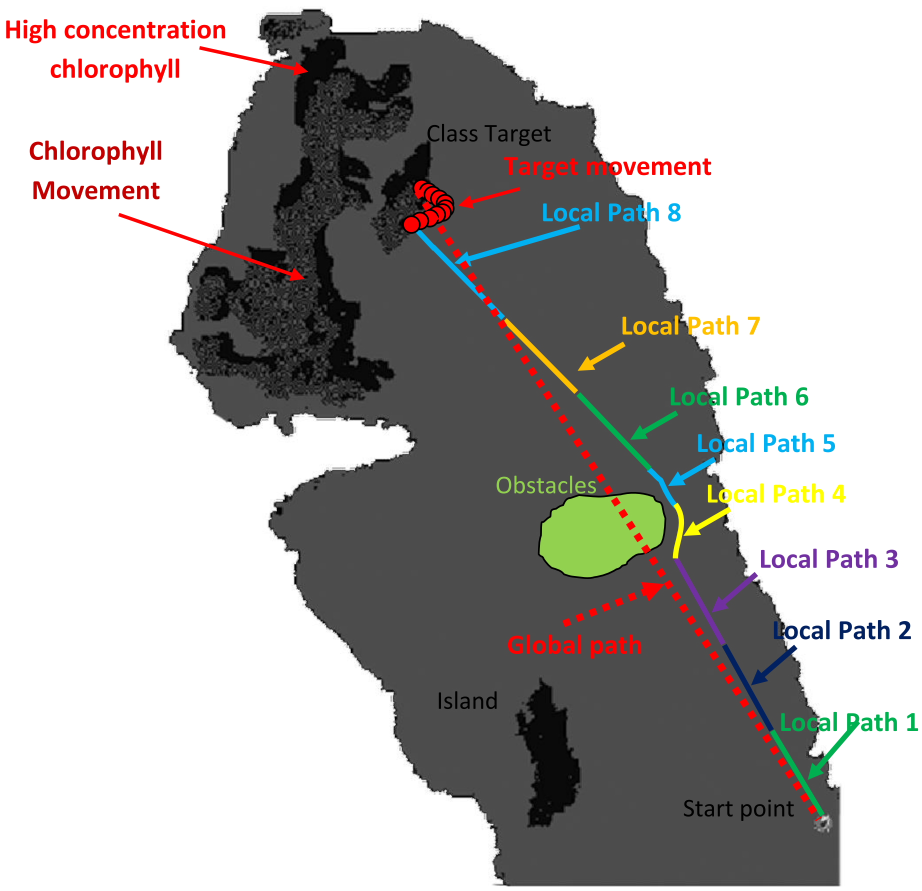


Figure 5. 12. Adaptive global path planning

5.5. Experiment

5.5.1 Experimental Framework

The experiments were carried out using satellite data from the northern basin of Lake Winnipeg for a path starting at the point located at longitude (99˚ 02' 08") W and latitude (55˚ 35' 18") N and the destination point at longitude (96˚ 50' 24") W and latitude (51˚ 55' 51") N. The direct distance between the start point and the target is around 236 km. The maps used in the experiments were in the form of a raster grid, where the dimensions of cells corresponded to the resolution of the MERIS satellite sensor, i.e., 260 m x 300 m. Each cell has an associated value $V_{x,y}$ obtained from the multi-layer map.

Genetic Algorithm Optimization Toolbox (GAOT) for Matlab was modified and used to program the proposed hybrid system. Table 8 shows the Genetic Algorithm parameters chosen for the optimization process.

Table 8: Parameters of the Genetic Algorithm

Genetic Parameters	Magnitude
Number of generations	150
Population size	120
Crossover rate	60% randomly and the elite
Mutation rate	5% randomly and the elite
Type of crossover	Single-point and multi-point crossover
Type of mutation	4 points random & 4 maximum points
Selection type	Roulette Wheel

a) Fitness function

The fitness function is a particular type of the objective function that quantifies the optimality of a solution and evaluates the suitability of a solution with respect to the overall goal. In our navigation problem, it maximizes the collected information, directs the robot towards the ROI, and incorporates distance and time penalties.

Many factors are involved in determining the weight of the sample in certain patches. For example, some types of samples can be kept in the dark on ice only for a limited time without any degradation. The time ranges from hours to days depending on the sample's sensitivity. Holding time is an important consideration because time-sensitive samples may need to be filtered in the field and placed on dry ice. Then time windows can be imposed for some types of water pollutant samplings considering the deterioration of the quality of samples of a specific pollutant. As an example, time requirements for chlorophyll concentration sampling are discussed in (Hambrook Berkman & Canova 2007).

In this thesis, a general objective function proposed to deal with the experiment conditions comprises the components: the samples value, the ROI award, the distance and the sampling time.

The proposed fitness function *F* consists of 4 components, calculated with each candidate sample (Halal & Zaremba 2017).

$$F = SV + ROI + DIS + ST \tag{5.9}$$

where:

SV - data set value, which determines the value of acquired samples according to Eq. 5.10;

$$SV = \left[\frac{\sum_{i=1}^{t} samples\,values}{\text{Maximum number of collected samples}} \right] \tag{5.10}$$

where sample values are calculated as the sum of all pollutant water values according to (5.11)

$$\max V = \max \sum_{j=1}^{M} \left[\sum_{i=1}^{N_J} (V_i^j(x, y)) \right] \tag{5.11}$$

where V is the value of the sample, Nj is the number of the samples for each pollutant, and M is the number of water pollutant classes.

ROI - the region of interest award was introduced in order to optimize the convergence of the search for quality samples. ROI numeric value is obtained by applying Eq.5.12;

$$ROI = \left[\frac{\sum_{i=1}^{t} ROI\_samples}{\text{Maximum number of collected samples}} \right] \quad (5.12)$$

DIS - distance factor;

ST - sampling time factor;

Two objective functions with different forms of DIS and ST factors were tested to assess their impact on the effectiveness of the sample acquisition mission:

- Objective function 1 linearly maximizes the sample value and the ROI award and exponentially minimizes the sampling time and the mission travel distance. The distance and the time become, as the sample acquisition mission progresses, quadratically more expensive.

Objective function 2 maximizes the sample value as well as the sampling time and the ROI award, and linearly minimizes the mission travel distance. As shown in Table 9

Table 9: Objective Function 2 and 1

Objective function 1	**Objective function 2**
$DIS = 1 - \left[\frac{\sum_{i=1}^{t} \sqrt{(y_t - y_{t-1})^2 + (x_t - x_{t-1})^2}}{\text{maximum allowable distance}} \right]^2$	$DIS = 1 - \left[\frac{\sum_{i=1}^{t} \sqrt{(y_t - y_{t-1})^2 + (x_t - x_{t-1})^2}}{\text{maximum allowable distance}} \right]$
$ST = 1 - \left[\frac{\sum_{i=1}^{t} sampling\ time}{\text{Maximum allowable sampling time}} \right]^2$	$ST = \left[\frac{\sum_{i=1}^{t} sampling\ time}{\text{Maximum allowable sampling time}} \right]$

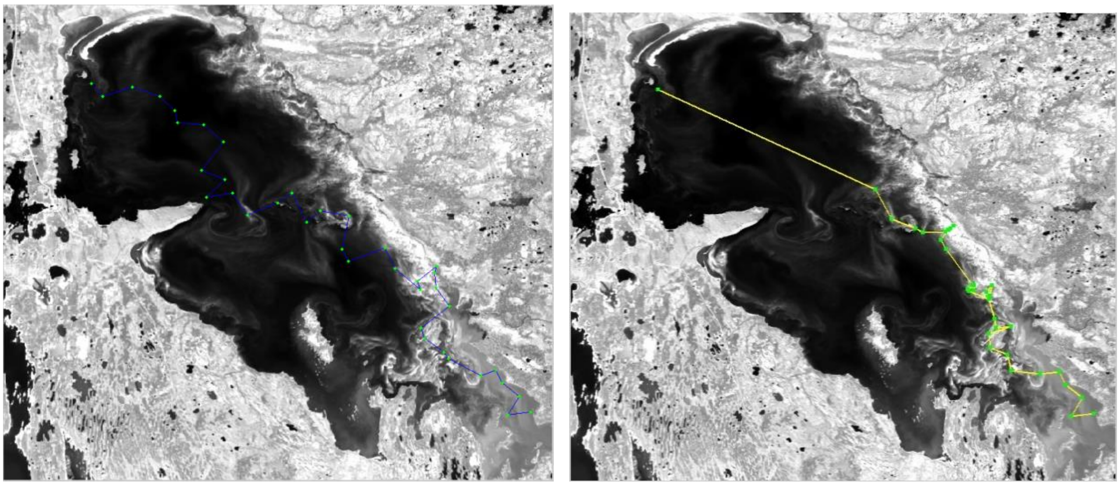
5.5.2 The Results

Four experiments were conducted with two objective functions tested. Objective function 2 (linear optimization) was incorporated in the fitness function used in experiments 1 and 2, and objective function 1 (exponential optimization) in experiments

3 and 4. Hard distance and time constraints were implemented in the first two experiments. The mission time was bounded by the value of 12 hours, and the travel distance was limited to 400 km. In experiments 3 and 4, the mission time had to be less than 9 hours, and the travel distance was limited to 330 km.

5.5.2.1 Path Planning Experiments

In the first experiment, the sample value (SV) was the sum of the TSS and Chl-a sample values. The results show that the path includes 10 samples from the clear water zone (outside the ROI zone), as shown in Fig. 5. 13a. The obtained results provide the rationale for hybridizing the GA-based search for optimal samples.



a) b)

Figure 5. 13. Sample acquisition paths: a) Experiment 1, b) Experiment 2.

A simple adaptive search, consisting of limiting the search to ROIs, as explained in waypoint modification section (5.5.3), was introduced in the second experiment. However, no specific behaviour guides the waypoint generation. Fig. 5. 13b) presents the path generated by the modified system. The sampling area is located entirely in the ROI. Table 10 compares the performance of the two experiments.

Table 10: Results of Experiments 1 and 2.

	Experiment 1 (GA)	**Experiment 2 (ROI-optimized GA)**
Sampling time	0.475 @ 38 samples	0.475 @ 38 samples
Real path length (m)	3.9989e+005	3.4364e+005
Samples value	0.7004	0.8304

ROI award	0.3675	0.5550

The path in the second experiment was approximately 56 km shorter, and the value of the samples increased by about 13 percent while keeping the number of samples at the same level.

The result obtained provides the reason to introduce the local search optimization behaviour to improve the path planning performance.

5.5.2.2 Multi-Behaviour Navigation

In order to assess the multi-behaviour performance of the system and to further improve the path quality - in the context of the GA methodology – different behaviours were introduced to the local adaptive search in the next two experiments. The third experiment explores the local behaviour optimization which performs two collection strategies depending on the types of samples. Therefore, the ROI set consists of two zones, Chl-a and TSS. The search minimizes the local path in the MCI patch according to Eq. 5.4, and maximizes the local path in the TSS patch according to Eq. 5.8. The neighbourhood of a solution is explored, and the best-evaluated neighbour is selected according to the adopted behaviour in each patch. Objective function 1 has been used to optimize this experiment. The multi-behaviour navigation shows good sampling performance in the two different patches, as shown in Fig. 5. 14.

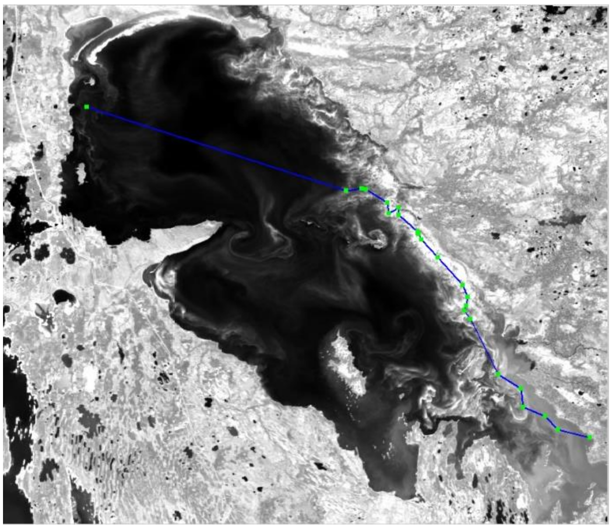


Figure 5. 14. Sample acquisition path from experiment 3.

The mission collects 22 pure chlorophyll-a samples and 6 TSS samples along a 282 km long path. The samples value is 0.645, and ROI award equals to 0.6125. The distances between the chlorophyll samples are shorter than between the TSS samples, which are a consequence of applying the behaviour equation (Eq. 5.4) and high award for the Chl-a ROI. The longer local path between the six TSS samples results from the behaviour equation (Eq. 5.8). The total mission time is 8 hours and 54 minutes. The travel time is 7 hours and 14 minutes.

In the fourth experiment, the zone of the maximum gradient of chlorophyll concentration was introduced, which produced three separate patches with three different local search behaviours. Due to the behaviour conflict between the maximum gradient and the maximum value of the chlorophyll concentration a new ROI zone was created. Thus, the three separate ROIs were generated as follows: the Chl-a zone, the maximum gradient of chlorophyll concentration, and the chlorophyll and TSS concentration zone. Fig. 5.15 depicts the ROI map which was used in this experiment. A modified fitness function was employed to comply with the goal of the experiment. The Chl-a samples were treated as the highest value samples with the shortest local path in the search algorithm that follows Eq. 5.4. In the maximum gradient zone, the search made the acquisition platform navigate in an adaptive way to follow the maximum gradient curve, using Eq. 5.5 and Eq. 5.6, and to maintain a proper distance between the samples. The chlorophyll and TSS zone adopted the behaviour model as in Eq. 5.7. All behaviour optimization algorithms explored the neighbourhood and selected new waypoints in order to enhance the quality of the solution. Fig. 5.16 shows an example of the planned path.

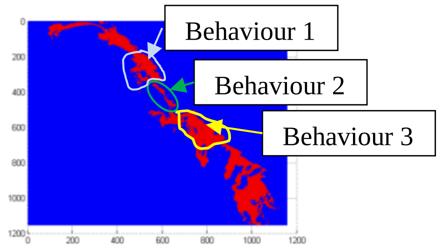


Figure 5. 15. Multi-behaviour sampling for different patches.

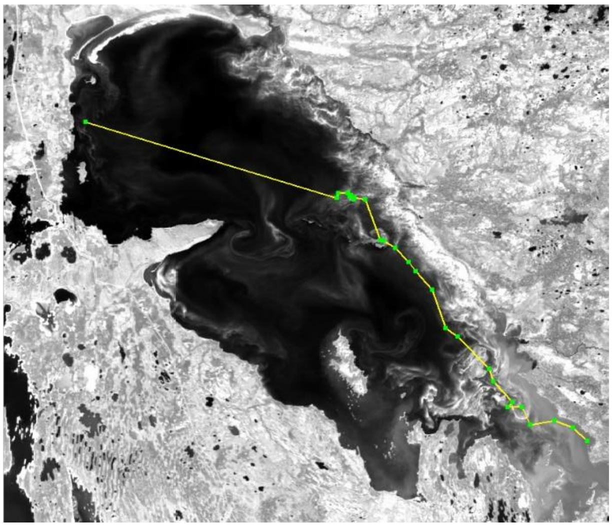


Figure 5. 16. Sample acquisition path from experiment 4

The path planning algorithm produced 28 samples as follows: 9 samples from the TSS & Chl-a zone; 5 samples from the MG zone; 14 samples from Chl-a zone including the start waypoint samples. The samples were collected along the path 285 km long. The normalized sample value was 0.5040 with the ROI award equal to 0.5650.

The experiments show that the adaptive GA-based path planning method offers robust search capabilities, and supports different sample acquisition strategies, ensuring the collection of meaningful data over pre-identified areas of interest.

5.5.2.3 Convergence Analysis

To improve the convergence of the GA-based search, two crossover and two mutation operations were employed. The solutions to these operations were divided into two categories as follows: the first one consists of the elite solutions, and randomly selected solutions represent the second category.

The simulation results show that:

(1) The new procedure effectively enhanced the global search ability and improved the local searching ability;

(2) High convergence rate was obtained.

The results without the enhancement are shown in Fig. 5,17a. Both the quality of the solution and the speed of the optimization are enhanced by an order of magnitude by applying the improved operations (Fig. 5. 17b).

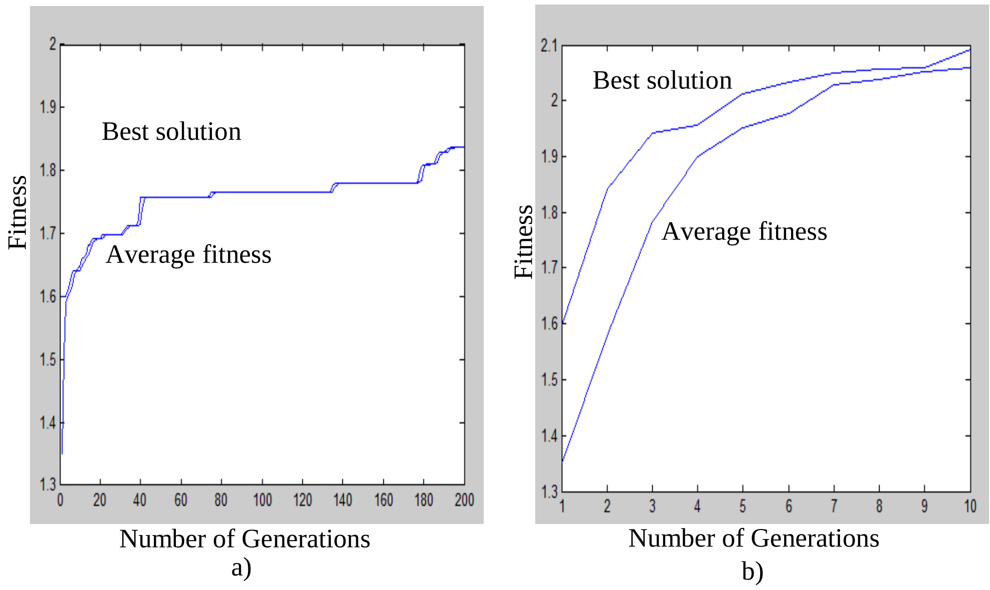


Figure 5. 17. Convergence in experiment 1 & 2

The repeatability of the results is depicted, for experiments 3 and 4, in Figures 5.18a and 5.18b respectively. The convergence of both the best solution and the average solution is high.

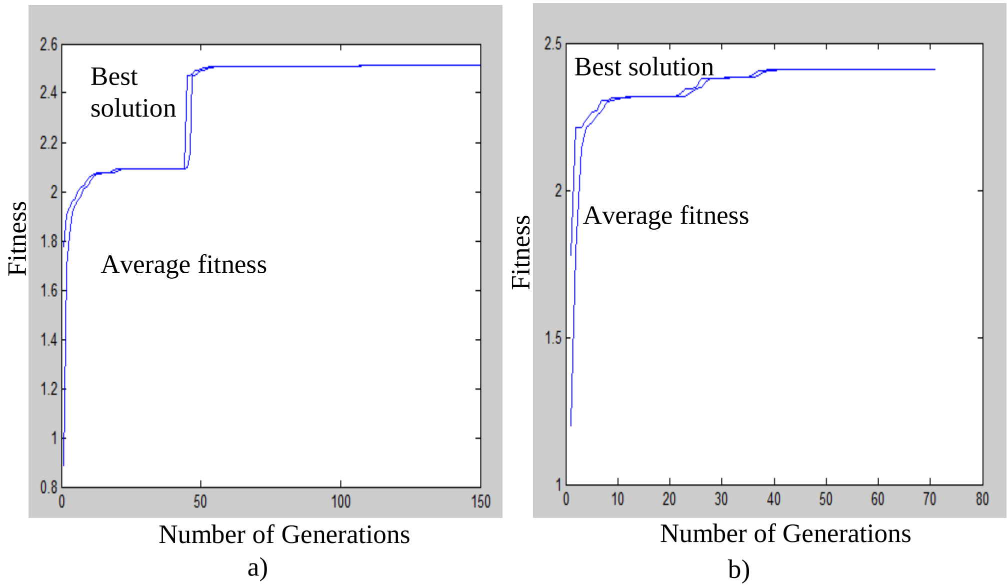


Figure 5. 18. Convergence in experiment 3 & 4

5.6. Conclusions

In this chapter, hybrid genetic algorithms were proposed for navigation in a partly unknown environment, where the objective of the planning task is to find the optimal path for a mobile sample acquisition platform. The total quantity and quality of water samples were maximized according to the navigation goals specified for each acquisition zone. Sampling in each patch may be guided by different patterns of behaviour for different purposes. Thus, the path planning has to be able to execute different behaviours along the global path. A hybrid genetic search was developed to deal with these requirements. The adaptive search algorithm (local search optimization) models behaviours in different surrounding area conditions and executes them in each generation at the level of local path navigation. The locality of the navigation was defined in terms of regions of interest (ROI). In the process of generating the waypoints, the adaptive search deletes and inserts new waypoints in each solution depending on the ROI behaviour. This enhances the flexibility and the efficiency of path planning. The ROI component was introduced also in the fitness function, greatly speeding up the convergence of the planning process. Tests were conducted using medium-resolution satellite imagery which represents the simulated environment. Multi-layer maps provided a rich context to the adaptive search system to perform flexible local search behaviours. The proposed hybrid genetic algorithms have demonstrated their usefulness in solving multi-objective path planning problems, where multi-objective functions are used to find the suitable solution which fits the overall mission goal.

Chapter 6

Multi-Behaviour Deliberative-Reactive Navigation

6.1 Problem Statement

In this chapter, we address the issue of hybrid deliberative-reactive navigation focusing on the local path execution in a dynamic environment. A multi-task robot is designed to comply with a complex surroundings' context. The flexibility of this design allows the robot to run both navigation levels in parallel while solving a critical or complex situation. The navigation controller makes the decision on which navigation level should be applied and keeps the other level working on solving the problem at its level.

This chapter deals with hybrid navigation and focuses mainly on the dynamic local path and unreachable local path subject to replanning (see 6.4.2 and 6.4.3) and conflicting behaviours (see 6.4.4). We present a hybrid geno-fuzzy control system for a small mobile robot which is capable of handling multi-behaviour operation. A hierarchical intelligent system using a geno-fuzzy algorithm has the ability to change, modify and teach a rules base to comply with the mission strategy. Depending on the mission objective and the robot's ability, our formal architecture is adopted and modified to deal with behaviours and control the navigation in both levels.

6.2 Control Architecture

Chapter 2 reviewed many behaviour controller approaches. These controllers make a decision depending on a predefined strategy and the surrounding area's conditions. The more criteria are considered, the more complex control is needed.

Based on the on integrated deliberative reactive architecture introduced in Section 4.1, a hybrid deliberative-reactive system is proposed to perform multi-behaviour navigation as shown in Fig. 6.1. Intelligent behaviour selector decides on the appropriate behaviour complying with the mission objective. It makes a suitable decision depending on the information received from onboard sensors (local context) and the mission strategies to execute its task.

Intelligent behaviour selector is responsible for either activating the replanning in a specific case such as when the robot can't execute its local tasks or deciding on the appropriate behaviour in the same situation by executing the conflicting behaviour.
Both deliberative and reactive navigation can be involved to solve the robot situation by employing different strategies such as avoidance strategy and aggressive strategy. Intelligent behaviour selector coordinates the behaviours at both levels. The behaviours can be divided into two basic groups as follows:

- Avoidance behaviours

  Avoidance behaviours are employed at the reactive level when the robot deals with a dynamic local path. In the case of an unreachable local path, an obstacle avoidance at behaviour is employed by activating the local or sub-global path replanning as determined by the robot situation.

- Aggressive behaviours

  Depending on the mission objective and the efficiency of the robot tasks, aggressive behaviours are used to solve a critical robot situation such as the recovery for a dead-end zone. The robot is allowed to approach the obstacle and to manage to move it. The robot creates a new local path or cleans the local path reducing the cost of the mission by saving time and energy.

The reactive module, which is shown in Fig. 6.1 in more detail, consists of fuzzy behaviour-based systems which handle individual behaviour strategies. Most fuzzy behaviour-based systems include the following behaviours which a mobile robot generally needs to apply in order to perform its task.

- Target reaching

If there is no obstacle, then go to the target

- Avoiding obstacles
  If the robot detects an obstacle, then avoid it

- Aggressive behaviour
  If the robot detects an obstacle, then seek the obstacle and move it

Geno fuzzy system is a satisfactory solution in multi-behaviours navigation where the robot needs to improve its performance by learning or optimizing these systems in the presence of uncertainty.

In the presented solution, fuzzy controllers (behaviour-based system) are employed to handle the reactive navigation system controls. The behaviour based system is used to handle a different group of behaviours which can be varied depending on the robot's abilities and the mission objective. Operating in a dynamic environment, the intelligent behaviour selector (decision maker) makes a decision to switch between behaviour-based fuzzy systems. At the reactive level, genetic algorithms are proven to be an efficient tool for designing an optimal fuzzy control system. The hybrid system optimizes the rules base of the fuzzy controller. The optimization improves the robot's performance and the robot's behaviours (see Section 6.3.2). Since the systems handle many rules bases, multi-objective optimization can be applied to optimize the behaviour-based systems obtaining high performance for each system.

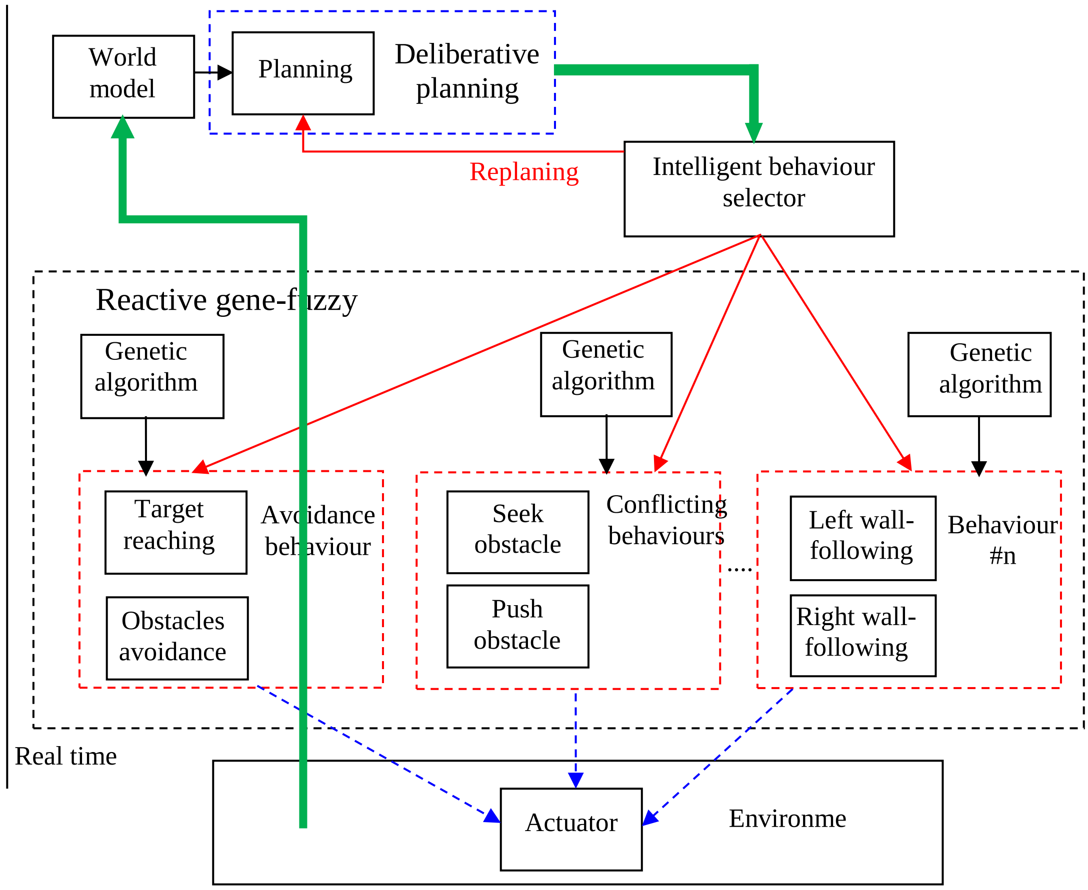


Figure 6. 1: Hybrid navigation control for multi-behaviour robot architecture

6.3 Experiment Framework

6.3.1 Multi-Behaviour Fuzzy System Controller

In this experiment, we used a fuzzy control to drive a Khepera robot in a simulation environment named Kiks. Khepera is a miniature mobile robot with a diameter of 55 mm and a weight of 70 g. The robot is supported by two lateral wheels that can rotate in both directions and two rigid points in the front and the back. By spinning the wheels in opposite directions at the same speed, the robot can rotate without lateral displacement. The robot has eight sensors distributed around the body, six on the front side and two on the back side. The control input variables are the six sensor inputs (S0 …S5), the robot coordinates and the heading direction. We ignored the two back sensor inputs (S6 and S7) that do no effect on the fuzzy control. The output variables are the left motor speed and right motor speed (LMS and RMS). Fig. 6.2 illustrates the Khepera robot with its fuzzy variables.

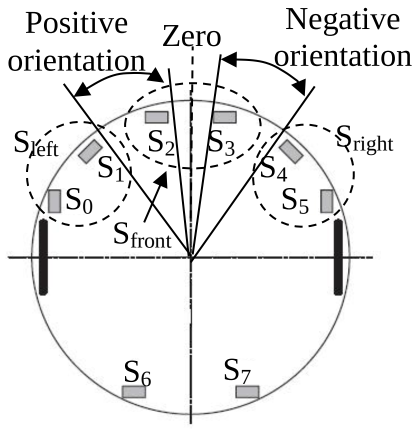


Figure 6. 2. Simulation model of the Khepera robot

The number of fuzzy control rules is determined by the number of the fuzzy membership functions of the controller inputs and the number of its inputs. The desired behaviour for our Khepera simulated robot is to move from the start point to the target point, avoid obstacles, follow walls and stop within the target zone.

The robot sensor readings are grouped into many groups to represent fuzzy input variables which divide up the rules base in smaller ones. Thus, Sensor simplification was used as follows to reduce the number of the sensor inputs:

$S_{left} = ((S0 + S1)/2)$

$S_{front} = ((S2 + S3)/2)$

$S_{right} = ((S4+S5)/2)$

This simplifies the fuzzy rules base structure, without affecting the robot performance.

In most fuzzy decision systems, non-fuzzy input data are mapped to fuzzy sets by treating them with Gaussian, triangular or other membership functions. Piecewise linear functions are evaluated faster and more efficiently by computers in embedded applications, hence the membership functions used in a fuzzy-logic navigation system of a mobile robot take triangular and trapezoidal forms (Yang et al. 2005). The input variables are S_{left}, S_{front}, and S_{right} as shown in Fig. 6.3. Each input has three trapezoidal linguistic membership functions, called *near*, *med*, and *far* (Figure 6.3 A and 6.3 B), denoting the distance from obstacles. All inputs have their membership functions of the same shape. The *orientation* input has five trapezoidal membership functions, called *hneg*, *neg*, *forward*, *pos*, and *hpos* representing the robot target direction (heading direction), as shown in Fig. 6.3 C. The output variables are *LMS* and *RMS*, which are the left motor speed and right motor speed. Each output has five triangular linguistic membership functions, called *hneg*, *neg*, *slow*, *norm* and *fast* (Fig. 6.3 D).

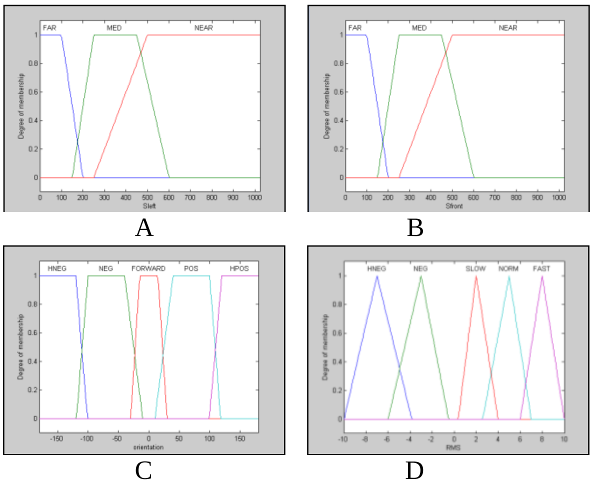


Figure 6. 3. A. *Sleft* input membership function, B. *Sfront* input membership function, C. *Orientation* input membership function, D. *RMS* output membership function.

The robot heading direction (the orientation input variable) is always taken as the 0° direction, with the negative direction, counter-clockwise direction, to its left and the positive direction, clockwise direction, to its right. The range of this variable, discrete universe of discourse, is [-π π] as shown in fig. 6.4.C

The output control variables are LMS and RMS, which are left motor speed and right motor speed. Each variable has five memberships function (HREV, REV, SLOW, NORM, FAST) denoting the absolute speed of the robot's left and right wheel. The distance between the robot and the target and the target direction are provided to the fuzzy controller as two additional inputs.

Each rule can be considered as a robot behaviour and the rules making the same behaviour are grouped together under one category.

Examples of the individual rules for avoidance obstacle behaviour:

Steering and tracking behaviour

Description

Target steering rules are used for orienting the robot towards his goal point. These rules are always active when the robot doesn't detect any obstacle. Dynamic tracking can be achieved by setting the destination to a moving goal.

Rules

The rules for steering towards the goal are as follows:

Steering right back behaviour - this behaviour occurs when the robot's target is into the right-left zone, and the following rules present this behaviour:

If (S_{left} is FAR) and (S_{front} is FAR) and (S_{right} is FAR) and (S_{back} is FAR) and (orientation is HNEG) then (LMS is FAST) (RMS is REV)

Steering right
If (S_{left} is FAR) and (S_{front} is FAR) and (S_{right} is FAR) and (S_{back} is FAR) and (orientation is NEG) then (LMS is FAST) (RMS is SLOW)

Steer straight
If (S_{left} is FAR) and (S_{front} is FAR) and (S_{right} is FAR) and (S_{back} is FAR) and (orientation is FORWARD) then (LMS is FAST) (RMS is FAST)

Steer left
If (S_{left} is FAR) and (S_{front} is FAR) and (S_{right} is FAR) and (S_{back} is FAR) and (orientation is POS) then (LMS is SLOW) (RMS is FAST)

Obstacle avoiding with steering behaviour

Description

The robot has the ability to avoid obstacles with respect to the target direction. When the robot encounters an obstacle in front, and the target direction is at the right side then he avoids the obstacle by moving to the right. The rules for obstacle avoiding and steering are as follows:

Rules

The obstacle at the front of the robot and the goal point at the right side:
If (Sleft is FAR) and (S_{front} is NEAR) and (S_{right} is FAR) and (S_{back} is FAR) and (orientation is NEG) then (LMS is FAST) (RMS is SLOW) (1)

Obstacle at the left side of the robot at a medium distance and the goal point at the right side:
If (S_{left} is MED) and (S_{front} is FAR) and (S_{right} is FAR) and (S_{back} is FAR) and (orientation is NEG) then (LMS is FAST) (RMS is SLOW) (1)

There are two obstacles at the left and front side of the robot at a medium distance and the goal point at the right-back side:
If (S_{left} is MED) and (S_{front} is MED) and (S_{right} is FAR) and (S_{back} is FAR) and (orientation is HNEG) then (LMS is FAST) (RMS is REV) (1)

The obstacle is at the right side of the robot at a medium distance and the goal point at the left right side:
If (S_{left} is FAR) and (S_{front} is FAR) and (S_{right} is MED) and (S_{back} is FAR) and (orientation is POS) then (LMS is SLOW) (RMS is FAST) (1)

There are two obstacles at the right and front side of the robot, at a medium distance and the goal point at the right-back side:
If (S_{left} is FAR) and (S_{front} is MED) and (S_{right} is MED) and (S_{back} is FAR) and (orientation is HNEG) then (LMS is REV) (RMS is FAST) (1)

Obstacle avoidance obstacle behaviour

Description

The obstacle avoidance behaviour prevents the robot from any collision by making many maneuvers such as avoid the left or right corner. The rules for obstacle avoidance are as follows:

Rules

Avoiding left obstacle:
If (S_{left} is NEAR) then (LMS is FAST) (RMS is REV) (1)

Avoiding right obstacle:
If (S_{right} is NEAR) then (LMS is REV) (RMS is FAST) (1)

Avoiding back obstacle:
If (S_{left} is FAR) and (Sfront is FAR) and (S_{right} is FAR) and (S_{back} is NEAR) then (LMS is FAST) (RMS is FAST) (1)

Avoiding front obstacle:
If (S_{left} is FAR) and (S_{front} is MED) and (S_{right} is FAR) and (orientation is NEG) then (LMS is FAST) (RMS is SLOW) (1)

If (S_{left} is FAR) and (S_{front} is MED) and (S_{right} is FAR) and (orientation is HNEG) then (LMS is FAST) (RMS is REV) (1)

If (S_{left} is FAR) and (S_{front} is MED) and (S_{right} is FAR) and (orientation is POS) then (LMS is SLOW) (RMS is FAST) (1)

If (S_{left} is FAR) and (S_{front} is MED) and (S_{right} is FAR) and (orientation is HPOS) then (LMS is REV) (RMS is FAST) (1)

If (S_{left} is MED) and (S_{front} is FAR) and (S_{right} is FAR) and (orientation is HPOS) then (LMS is NORM) (RMS is NORM) (1)

If (S_{left} is FAR) and (S_{front} is FAR) and (S_{right} is MED) and (orientation is HPOS) then (LMS is NORM) (RMS is NORM) (1)
If (Sleft is FAR) and (S_{front} is NEAR) and (S_{right} is FAR) and (Sback is FAR) and (orientation is HNEG) then (LMS is FAST) (RMS is REV) (1)

If (S_{left} is FAR) and (S_{front} is NEAR) and (S_{right} is FAR) and (S_{back} is FAR) and (orientation is POS) then (LMS is REV) (RMS is FAST) (1)

If (S_{left} is FAR) and (S_{front} is NEAR) and (S_{right} is FAR) and (S_{back} is FAR) and (orientation is HPOS) then (LMS is REV) (RMS is FAST) (1)

If (S_{left} is FAR) and (S_{front} is NEAR) and (S_{right} is FAR) and (S_{back} is FAR) and (orientation is FORWARD) then (LMS is REV) (RMS is FAST) (1)

If (S_{left} is FAR) and (S_{front} is MED) and (S_{right} is FAR) and (S_{back} is FAR) and (Orientation is FORWARD) then (LMS is REV) (RMS is FAST) (1)

Wall following behaviour

Description

The wall following behaviour is a tracking control mechanism which is able to walk along any continuous surface keeping a fixed distance, wall following behaviour is useful when exploring an unknown environment. The rules for following walls are as follows:

Rules

If (S_{left} is MED) and (S_{front} is FAR) and (S_{right} is MED) then (LMS is SLOW) (RMS is SLOW) (1)

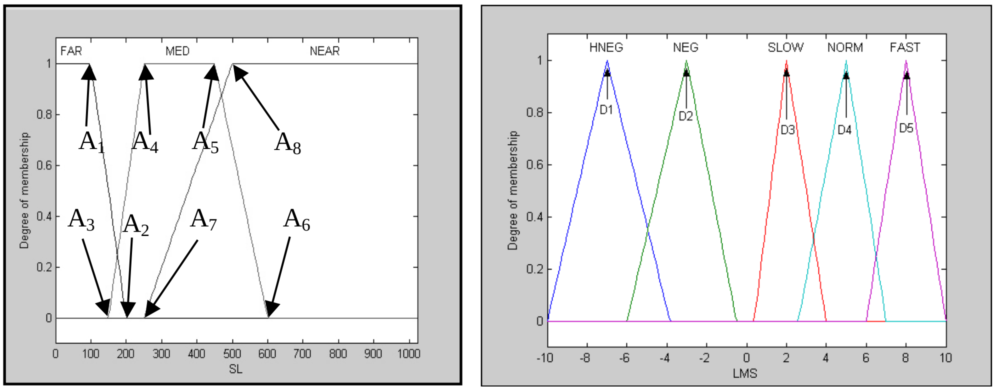
If (S_{left} is NEAR) and (S_{front} is FAR) and (S_{right} is NEAR) then (LMS is SLOW) (RMS is SLOW) (1)

6.3.2 Optimization and Adaptation of the Fuzzy System

6.3.2.1. Fuzzy System Optimization

Recently numerous researchers explored the integration of genetic algorithm with fuzzy logic systems. Researchers are concerned with the optimization of fuzzy systems either by automatically designing the membership function or by learning the fuzzy if-then rules. The role of the genetic algorithm in the geno-fuzzy system is to optimize the fuzzy system. Thus, a genetic algorithm generates the fuzzy parameters, and it evaluates these parameters by using a fitness function. The optimization can be done in many ways by generating fuzzy membership functions, rules base, output rules base or by changing the fuzzy operators and defuzzification strategies. These tasks can be done together or each of them alone. A chromosome encoding algorithm used to optimize the membership functions, and the output rules base was presented in (Halal & Dumitrache 2007).

An optimal fuzzy control system is obtained which drove the Khepera mobile robot to achieve its target with good performance and optimal behaviours. Thirty-four parameters from the input and output membership function were encoded to form a chromosome segment in order to optimize the shape of these functions. Figure 6.4 shows the eight genes that were encoded from S_{left} input function; these genes are called respectively A_1, A_2, A_3, A_4, A_5, A_6, A_7 and A_8. From the S_{front} and S_{right} input, we encoded the two other chromosomes, segment B and segment C, where each of them contains eight genes. They are named respectively B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, C_1, C_2, C_3, C_4, C_5, C_6, C_7 C_8, and Fig. 6.4 shows the membership functions chromosome segment. The genes D_1, D_2, D_3, D_4, D_5, E_1, E_2, E_3, E_4, and E_5. These input variables are encoded for tuning the LMS and RMS output membership functions.



A_1	….	A_6	A_7	A_8	B_1	B_2	..	B_8	C_1	C_2	…	C_8	D_1	...	D_5	E_1	…	E_5

Figure 6. 4. Chromosome's segment for encoding membership functions

In this fuzzy system, 18 antecedence rules were proposed, and the genetic algorithm optimizes the output of the rules base. We don't need to generate the whole rules base because the robot operates in a simulation environment and thus, the inputs values are predictable. The chromosome output rules segment contained 36 genes that present the supposed LMS linguistic term and RMS linguistic term, as shown in Fig. 6.5.

F_1	F_2	F_3	F_4	…	…	F_{33}	F_{34}	F_{35}	F_{36}
Output rule 1		Output rule 2				Output rule 17		Output rule 18	

Figure 6. 5. Chromosome Segment F

Figure 6.6 shows the whole chromosome and his genes.

A_1	--	**A_8**	**B_1**	--	**B_8**	**C_1**	--	**C_8**	**D_1**	--	**D_5**	**E_1**	--	**E_5**	**F_1**	**F_2**	--	**F_{35}**	**F_{36}**

Figure 6. 6.The whole chromosome

- **Multipoint crossover**

The multi-point crossover is the suitable genetic operator method that can be used in this problem in order to increase the number of string segments exchanged. The parent chromosomes, P_1 and P_2, are cut virtually at multiple random locations, and the portions of the chromosome between the cuts were exchanged. The result is two offspring I_1 and I_2, as is shown in Fig. 6.7. We used the multi-point crossover because the genes have integer values, the genes values of the output rules base are between 1 and 5, whereas the genes values of membership function chromosome segment are between 100 and 800, depending on the membership function itself. On the other hand, the genes had bounded to keep the overlaps between the membership functions.

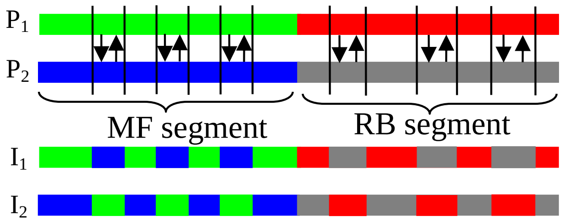


Figure 6. 7. Multi-point crossover

6.3.2.2. Encoding of the Rule-Based System

In order to optimize or adapt the behaviour-based systems, rules bases are encoded into corresponding chromosomes for a genetic algorithm. We encoded the membership functions in each sensor input (Far, Med and Near), coded as 1, 2 and 3. The input *orientation* membership functions called *hneg*, *neg*, *forward*, *pos*, and *hpos* were encoded as 1 2 3 4 5. For each output membership function (Hneg, Neg, Slow, Norm, and Fast), they were coded as 1, 2, 3, 4 and 5. Figure 6.9 illustrates the encoded input membership function and the encoded output membership function. So, the rule *If* S_{left} *=far and* S_{front} *= far and* S_{right} *= far and orientation forward Then LMS = fast and RMS = fast*, can be encoded as a string vector 1 1 1 3 5 5, and the chromosome can be represented as a string vector (Fig. 6.8).

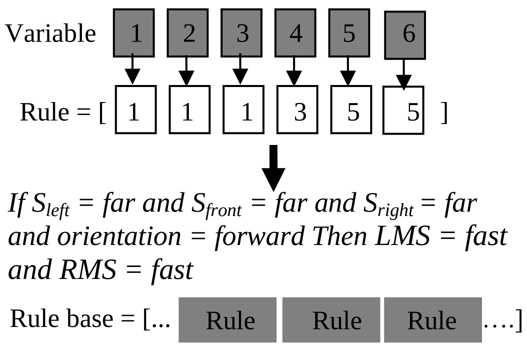


Figure 6. 8. The encoded rules base chromosome encoded

6.3.3 Multi-Behaviour Navigation Controller

This experiment employs multi-behaviour-based fuzzy systems using chromosomes to handle the rules bases as seen in 6.3.2.2. A modified deliberative-reactive multi-behaviour navigation architecture is proposed to attain the objective of the experiment. As shown in Fig. 6.9. The control architecture consists of two fuzzy systems which handle conflicting behaviours.

Intelligent behaviour selector is employed to control the multi-behaviour strategies depending on the local context conditions. This control unit prioritizes one system over the other, so the two fuzzy systems do not work simultaneously.

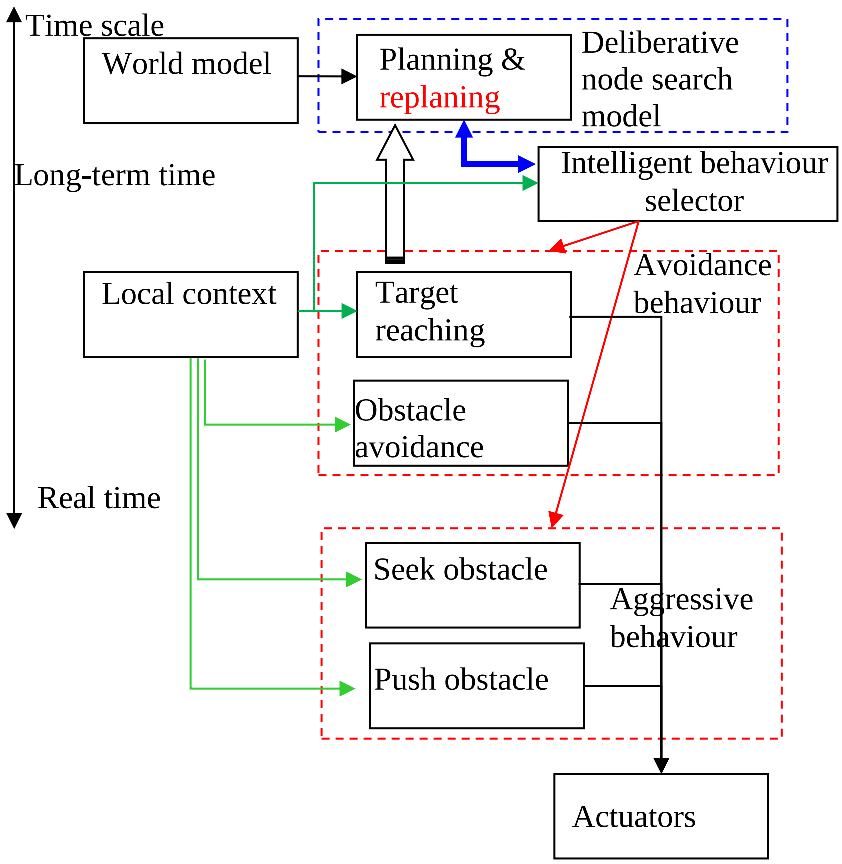


Figure 6. 9. Hybrid deliberative-reactive multi-behaviour navigation architecture

6.4 Experiment Results

All the obstacles in the global path are unforeseen obstacles. Thus, the robot executed its task in a partially unknown environment. The global navigation generated the optimal global path depending on the provided information, represented by the word model as in Fig. 6.10.

In this experiment, we used hierarchy and hybrid fuzzy control to drive a Khepera robot on a given simulation environment named Kiks. Kiks is a Khepera Simulator which uses MatLab software to drive the simulation. The simulator models a single Khepera that operates in many enclosures area with reconfigurable walls and obstacles as seen in Fig. 6.10. The robot is provided with a 2D-map, consisting of line features representing walls and other obstacles, as well as with its own location on this map. 12 given nodes with edges between them represent the area in which the robot navigates as shown in Fig. 6.10. The paths directions are always towards the target, but when the robot faces a blocked path or an unreachable local path, the motion on edge can move in both directions allowing the robot to recover from the dead-end zone.

A heuristic search was applied considering one direction for local paths which approached towards the target. A chromosome pool consisted of 100 individuals who were evaluated by a fitness function to choose the optimal global path from the start point to the target point taking the shortest global path. The pool and the fitness function were also applied in the sub-global path.

If two nodes n_1 and n_2 are connected, then qualitative actions $Action_n$ exist to move the robot from n_1 to n_2. Reactive navigation is employed to perform the local paths, where fuzzy reactive strategies handle the task to perform the local paths.

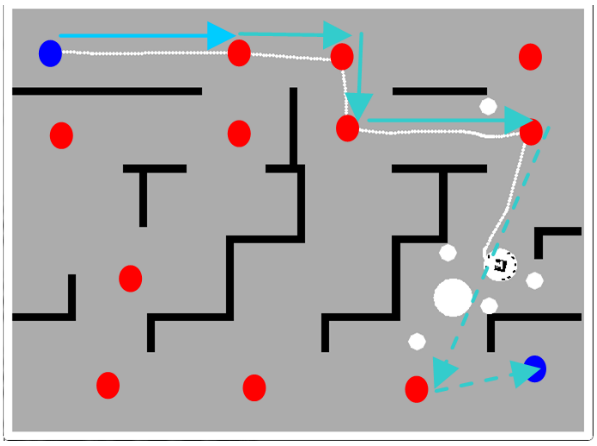


Figure 6. 10. Robot environment map and dead-end zone problem

6.4.1 Avoidance Behaviour Strategy

A simple strategy is investigated, where rules that cause the same effect in the robot movement and low probability rules to occur are not considered. This strategy is called **avoidance behaviour**. A fuzzy system is used to handle eight behaviours which were divided into two groups: the first group helps the robot to get its target. It consists of four behaviours which are: walk straight to target, walk along the corridor, left wall following, and right wall following; The second group prevents the robot from any collision by handling four behaviours which are: avoiding right obstacle, avoiding left obstacle, avoiding front obstacle, and avoiding blocked zone. Table 11 presents this fuzzy rule base structure.

Table 11: Avoidance Strategy Fuzzy Rules Base

S_{left}	S_{front}	S_{right}	ORT	LMS	RMS	Robot's behaviours
1	1	1	1	5	1	straight movement
1	1	1	2	5	2	
1	1	1	3	5	5	
1	1	1	4	1	5	
1	1	1	5	2	5	
2	1	2	0	4	4	walk along the corridor
3	1	3	0	3	3	
2	1	3	0	3	4	
3	1	2	0	4	3	
2	1	1	0	4	2	left wall following
3	1	1	0	5	1	
1	1	2	0	2	4	right wall following
1	1	3	0	1	5	
2	2	1	0	4	2	avoiding left front obstacles
3	3	1	0	5	1	

3	2	1	0	5	2	
1	2	2	0	2	4	avoiding right front obstacles
1	3	3	0	1	5	
1	2	3	0	2	5	
1	2	1	0	1	5	avoiding front obstacle
2	2	2	0	2	2	avoiding blocked zone
3	3	3	0	2	2	

- **Fuzzy inference diagram**

Figure 6.11 depicts the fuzzy inference diagram for the basic strategy fuzzy inference system, which consists of 23 rules. The diagram shows all parts of the fuzzy inference process from inputs to outputs. Each row in Fig. 6.11 corresponds to one rule and each column in Fig. 6.11 corresponds to either an input variable (yellow, on the left) or an output variable (blue, on the right).

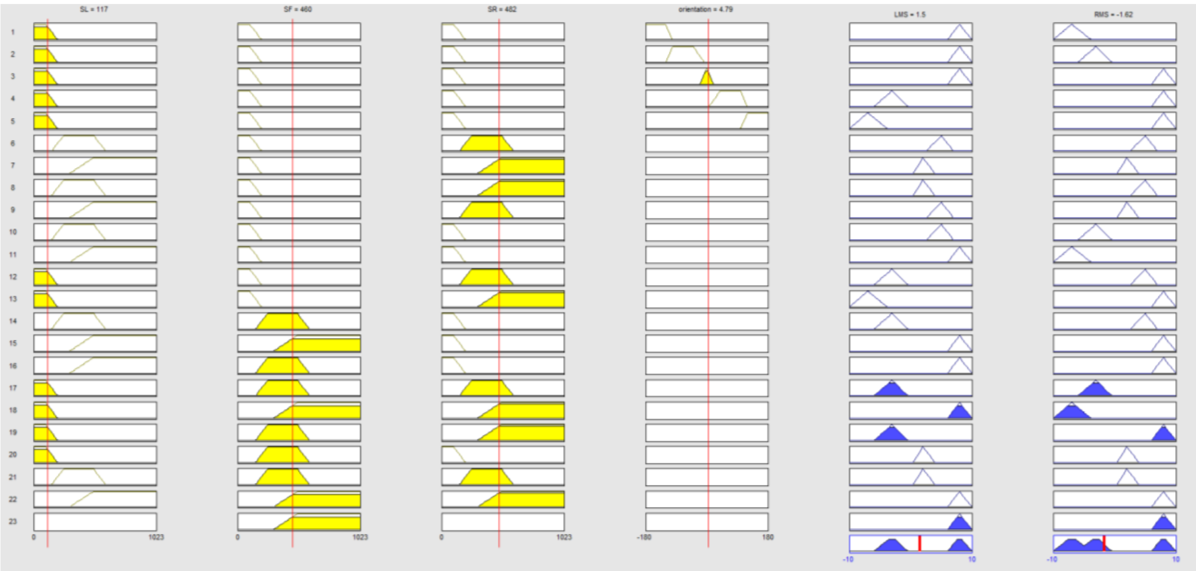


Figure 6. 11. General fuzzy inference system for basic strategy

The heuristic search applied in the fuzzy **avoidance behaviour strategy** finds the optimal path which is divided into many local paths. The local paths are the connections between two nodes as illustrated in Fig. 6.10. The fuzzy system doesn't handle a conflicting behaviour and doesn't activate the deliberative navigation to replan the sub-global path. Thus, the robot falls in the dead-end zone, and it cannot finish its tasks (Fig. 6.12). The robot performs infinitely two behaviours, which are seeking target and avoiding obstacle. In this situation, the robot keeps moving to the right and to left sequentially and endlessly.

The control system is too weak to solve this situation. Thus, the **replanning process of** local paths or/and (sub-global path) **is needed at the global navigation level** to solve such as critical situation.

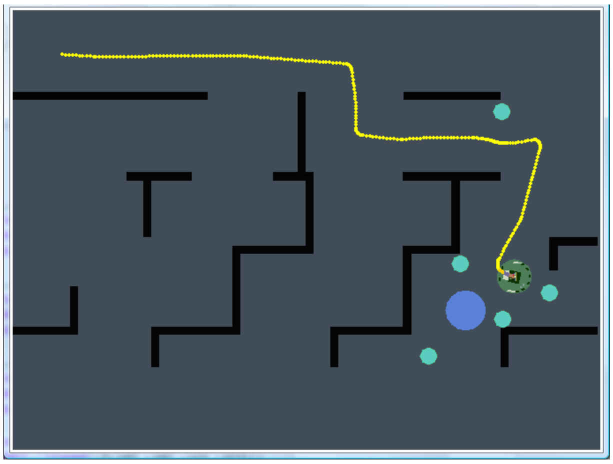


Figure 6. 12. Robot fell in dead-end zone using avoidance behaviour strategy

6.4.2 Dynamic Local Path

In a dynamic environment, dynamic local paths are called Incompletion replanning. In this situation, local paths have different original points between the replanning path and the original planning path as illustrated in Fig. 6.13. Completion replanning has the same original point between the replanning path and the original planning path. Completion of the replanning path can be obtained in a static and completely known environment which is represented as a blue dashed local path in Fig. 6.13. The reactive navigation combines the planning information and the real-time information to perform the new trajectory using avoidance behaviour strategy, (refer to section 4.11.2). Fig. 6.13 represents a dynamic local path.

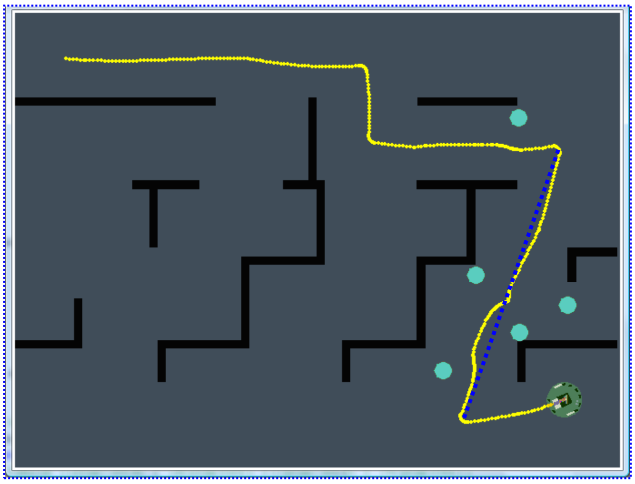


Figure 6. 13. Dynamic local path

6.4.3 Sub-Global Path & Unreachable Local Path

A Sub-global path is needed in many situations such as an unreachable and blocked path. Fig. 6.14 illustrates a global path and a sub-global path. The robot starts executing its local paths until he is confronted with a blocked path. The robot updates the world model. In this situation, the deliberative navigation replans a sub-global path from the robot location to the target. Fig. 6.14 shows how the robot recovers from a dead-end zone and seeks its target executing a new sub-global path. (Refer to section 4.11.3).

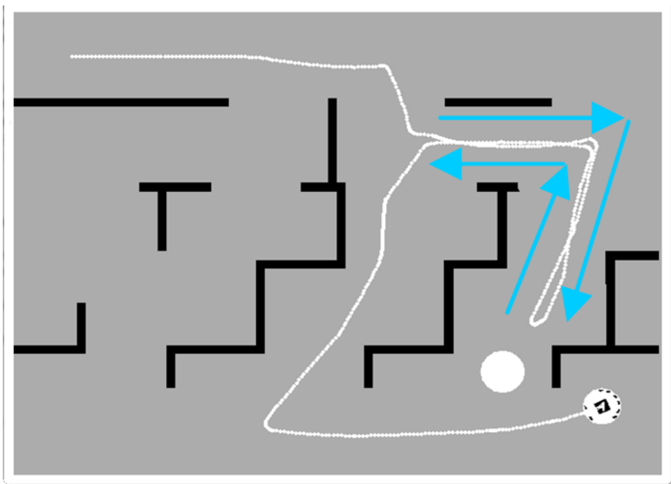


Figure 6. 14. Global and sub-global path

6.4.4 Aggressive Strategy (Aggressive Behaviour)

This system is equipped with aggressive behaviours which are: handle, seek, and push the obstacles. When considering the many local paths, a robot can take, they have to be clear to allow the robot to reach its target. If, however, the robot faces a blocked local path which he considers to be clear, then the genetic algorithm system stops the avoidance strategy and extends the priority to the aggressive strategy, which clears the respective path. This system handles several rules that help the robot to push the obstacles which are situated on the local path axis. Table 12 presents the fuzzy rules base structure.

Table 12: Seeking and Pushing Object (Aggressive Strategy)

S_{left}	S_{front}	S_{right}	ORT	LMS	RMS	Robot behaviours
2	2	1	0	2	4	Seek left front obstacles
3	3	1	0	1	5	
3	2	1	0	2	5	
1	2	2	0	4	2	Seek right front obstacles
1	3	3	0	5	1	
1	2	3	0	5	2	
1	2	1	0	3	3	Seek front obstacle
2	2	2	0	4	4	push the obstacle
3	3	3	0	5	5	

Arming aggressive behaviour (switching behaviour)

In this stage, a multi-task robot is applied handling conflicting behaviours. Many fuzzy-based behaviour systems are used to perform the different strategies, where the navigation control (hierarchy system) switches between two behaviours rules bases depending on the mission objective and the local context conditions, the intelligent behaviour selector applies aggressive behaviour which it considers as one of the solutions to solve an unreachable path. Fig. 6.15 shows how the robot seeks a ball and pushes it away allowing the robot to get to its target. (Refer to Section 4.11.3 unreachable local path in integrated navigation architecture).

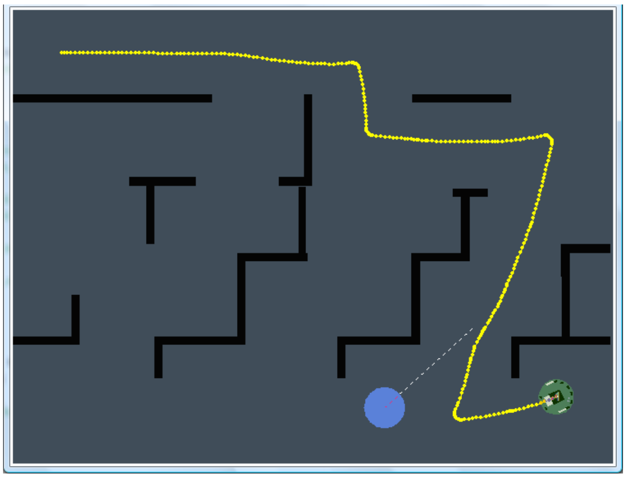


Figure 6. 15. Aggressive behaviour

6.5 Conclusions

The hybrid navigation approach designed and investigated in this Chapter used the deliberative navigation to supervise and control reactive navigation. The design of the robot controller was done to control the behaviour strategies by using Computational Intelligence techniques which have mostly a hierarchical design. This system proved its ability to control multi-behaviours navigation and conflicting behaviours that employ different sets of behaviour-based subsystems.

The hierarchical intelligent system using genetic algorithms was proven to be an efficient tool for controlling fuzzy control systems. The geno-fuzzy system controlled successfully the behaviour strategies providing a range of decisions which helped the robot to perform many conflicted behaviours. The geno-fuzzy system introduced multi-tasks to the robot controller.

The hybrid multi-behaviour navigation has been shown to be a robust and flexible solution for complex and dynamic environments, recovering the robot from dead-end zones and cycling modes. The designed system provides a suitable degree of reactivity and deliberation in a mobile robot giving the optimal global path and optimal sub-global path.

Chapter 7

Conclusions

This thesis presents research and development work related to hybrid multi-behaviour navigation primarily in a large-scale environment. Three main goals have been accomplished: The first goal was to improve and optimize deliberative multi-behaviour navigation in a large-scale, dynamic environment, the second one was to devise hybrid multi-behaviour deliberative-reactive navigation at the local level, and the third one was to design an integrated deliberative-reactive multi-behaviour navigation architecture.

1-Deliberative multi-behaviour navigation

In terms of the development of the deliberative navigation, a hybrid GA-based method was proposed and developed to optimize path planning and navigation for large-area monitoring using pollutant maps generated from RS imagery. The power of the global GA-based search was combined with the speed of a local optimizer. Both optimizers work cooperatively to find the optimal solution, where GA determines the optimal region according to the monitoring strategy, and then the local optimizer takes over to find the best position for acquiring water samples.

Multi-layer maps were employed to generate spatial and functional properties of the environment. Those maps enable the planning system to perceive and interpret environments according to different environment features that characterize, in our case, the pollutants and the acquisition strategy (e.g., the maximum gradient of a pollutant concentration). Layers originating from remote sensing imagery were complemented by meteorological and ancillary data to form a perceptive system in the form of hybrid maps. The multi-layer model is subsequently used by a cost optimizing path planner. The presented approach is general, and can be applied in a variety of environments that require multi-dimensional representations.

Appropriate environment representation models were developed to interpret the global environment. Parametric and nonparametric classification approaches were used to identify the context for optimal interpretation of the measurement data. In the realm of context analysis, the concept of the Area-Of-Interest was given special attention as especially useful in the further investigation of the multi-behaviour navigation process.

Metaheuristic search methods were employed in the planning part, showing their flexibility and good performance. Several hybrid algorithms were designed to solve the navigation problem, where the trajectory consists of different behaviours complying with different environment entities. Hybrid deliberative navigation approach improved the multi-behaviour deliberative navigation employing the different behaviour models. The developed hybrid genetic algorithm dealt with the complex world model on the deliberative level, showing enhanced search behaviours optimized for different contexts. The optimization process created optimal paths that fit the overall mission goals, respecting at the same time the temporal and spatial constraints. It was demonstrated that metaheuristic approach is useful in solving multi-objective path planning problems, where the mobile robot navigates and performs its tasks in a partially unknown and unstructured environment.

2- Hybrid multi-behaviour deliberative-reactive navigation

At the reactive level, due to a partially unknown environment and the complexity of the global context, a dedicated multi-behaviour reactive navigation architecture is needed to deal with these conditions. While a behaviour-based system allows the robot application-specific behaviours where each behaviour is concerned with a sole objective, the presented multi-behaviour navigation, including conflicting behaviours, employed a set of behaviour-based subsystems. The reactive system was divided into many sub-systems where each system handles a group of behaviours. Behaviours which do the same tasks are handled together by one behaviour-base system. Computational intelligence techniques were used to handle multi-task navigation. The design of the robot controller was done around a hierarchical fuzzy system. The hierarchical intelligent system proved its ability to control multi-behaviours navigation. A geno-fuzzy system was designed such as to optimize and control the rules structure of a fuzzy controller. A new set of behaviours is obtained by the generation of a fuzzy rules base by a genetic algorithm. The algorithm generates and optimizes the fuzzy behaviour-based system depending on the objective function and the surrounding area conditions.

3- Novel integrated deliberative-reactive multi-behaviour navigation architecture

A novel integrated deliberative reactive navigation architecture was developed for multi-behaviour navigation in complex and unstructured environments. A formal state transformation type model was presented and examined based on the examples of deliberative and deliberative-reactive navigation. The model provides a general scheme for designing and controlling the relationship between both navigation levels. It incorporates the capacity for multi-behaviour navigation by including the concept of the navigation context. The replanning process can be solved at both levels with the employment of a specific strategy. The integrated system has the ability to activate the suitable global search method as well as the search cost function providing optimal global path and sub-global trajectories.

The concepts and the methods investigated in this thesis were analyzed and validated in two sets of experiments. The results of the first set of experiments showed that the developed intelligent system can perform multi-task navigation including the conflicting behaviour in an environment monitoring mission in different terrain coverage scenarios. The objective function can be modified depending on the mission objective. Constraint functions and ROI were used to guide an acquisition platform to areas of higher interest and to help make a decision on the suitable behaviour. In the second set of experiments, a hybrid navigation approach used deliberative navigation to supervise and to control reactive navigation at a local level. The hybrid navigation was shown to provide a more robust and flexible solution for dynamic environments, allowing the robot to recover from dead-end zones and cycling modes. This system provided a suitable degree of reactivity and deliberation in a mobile robot. Hybrid geno-fuzzy navigation control drove the robot in a dynamic environment applying multi-behaviour navigation.

Future work

- Adaptation and optimization of the developed architectures and methods for Unmanned Aerial Systems (low-cost aerial platforms).

- Extension of the work to other fields of environmental sciences, such as assessment of vegetation dynamics and forests biodiversity, wildlife research and management, changes in freshwater marshes and river habitats, and conservation and monitoring programs.
- Investigation of other hybrid metaheuristic methods system for global search, such as the proposed GAACO.

References

Abdellatif, M. (2008) 'Behavior Fusion for Visually-Guided Service Robots', Computer Vision. Xiong Zhihui (Ed.), ISBN: 978-953-7619-21-3, InTech, p. 538.

Abraham, A. (2003) 'Intelligent Systems: Architectures and Perspectives', in Recent Advances in Intelligent Paradigms and Applications. Physica-Verlag HD, pp. 1–35. doi: 10.1007/978-3-7908-1770-6_1.

Abu-Dakka, F. J., Valero, F. and Mata, V. (2012) 'Evolutionary Path Planning Algorithm for Industrial Robots', Advanced Robotics, 26(11–12), pp. 1369–1392. doi: 10.1080/01691864.2012.689743.

Altuntaş, B. (2003) a Behavior Based Robot Contol System Architecture for Navigation in Environments With Randomly Allocated Walls. Master thesis,Middle East Technical University.

Amin, S., Mamat, R. and Kwong, T. (2005) 'Behaviour Based Mobile Robot Navigation with Dynamic Weighted Voting Technique', in Cutting Edge Robotics chapter 13, pp. 201–223. doi: DOI: 10.5772/4650.

Antonelo, E., Schrauwen, B. and Stroobandt, D. (2008) 'Modeling multiple autonomous robot behaviors and behavior switching with a single reservoir computing network', in 2008 IEEE International Conference on Systems, Man and Cybernetics. IEEE, pp. 1843–1848. doi: 10.1109/ICSMC.2008.4811557.

Baltzakis, H. (2004) A Hybrid Framework for Mobile Robot Navigation Modelling with Switching State Space Networks. Ph. D. Thesis, University of Crete,.

Bao, Q., Li, S., Shang, W. and An, M. (2009) 'A Fuzzy Behavior-Based Architecture for Mobile Robot Navigation in Unknown Environments', in 2009 International Conference on Artificial Intelligence and Computational Intelligence. IEEE, pp. 257–261. doi: 10.1109/AICI.2009.125.

Belkhouche, F. (2009) 'Reactive path planning in a dynamic environment', IEEE Transactions on Robotics, 25(4), pp. 902–911. doi: 10.1109/TRO.2009.2022441.

Binitha, S. and Sathya, S. S. (2012) 'A Survey of Bio inspired Optimization Algorithms', International Journal of Soft Computing and Engineering (IJSCE), 2(2), pp. 137–151. ISSN: 2231-2307

Blum, C. and Roli, A. (2003) 'Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison Metaheuristics in Combinatorial Optimization', ACM Computing Surveys, 35(3), pp. 268–308.

Braünl, T. (2006) Embedded robotics, Springer, ISBN: 9783540343189, doi: 10.1007/3-540-34319-9.

Brom, C. and Bryson, J. (2006) 'Action selection for Intelligent Systems', European net-work for the advancement of artificial cognitive system.

Brooks, R. (1986) 'A robust layered control system for a mobile robot', IEEE Journal on Robotics and Automation. IEEE, 2(1), pp. 14–23. doi: 10.1109/JRA.1986.1087032.

Busquets-Font, D. (2003) A Multiagent Approach to Qualitative Navigation in Robotics. Ph.D. Thesis, Universitat Politecnica De Catalunya.

Châari, I., Koubâa, A., Bennaceur, H., Ammar, A., Trigui, S., Tounsi, M., Shakshuki, E.

and Youssef, H. (2014) ‘On the Adequacy of Tabu Search for Global Robot Path Planning Problem in Grid Environments’, in 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014). Hasselt, Belgium. Available at: www.elsevier.com/locate/procedia (Accessed: 14 June 2016).

Châari, I., Koubâa, A., Bennaceur, H., Trigui, S. and Al-Shalfan, K. (2012) ‘smartPATH: A hybrid ACO-GA algorithm for robot path planning’, in 2012 IEEE Congress on Evolutionary Computation, CEC 2012. Brisbane, Australia, pp. 1–8. doi: 10.1109/CEC.2012.6256142.

Chen, C. and Cheng, Y. (2008) ‘Research on Map Building by Mobile Robots’, in 2008 Second International Symposium on Intelligent Information Technology Application. IEEE, pp. 673–677. doi: 10.1109/IITA.2008.205.

Chen, C., Li, H.-X. and Dong, D. (2008) ‘Hybrid Control for Robot Navigation’, IEEE Robotics and Automation Magazine, (June), pp. 37–47. doi: 10.1109/MRA.2008.921541.

Cheng, C. T., Fallahi, K., Leung, H. and Tse, C. K. (2012) ‘A genetic algorithm-inspired UUV path planner based on dynamic programming’, IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 42(6), pp. 1128–1134, doi: 10.1109/TSMCC.2011.2180526.

Chunlin Chen, C., Han-Xiong Li, H.-X. and Daoyi Dong, D. (2008) ‘Hybrid Control for Robot Navigation - A Hierarchical Q-Learning Algorithm’, IEEE Robotics & Automation Magazine. IEEE, 15(2), pp. 37–47, doi: 10.1109/MRA.2008.921541.

Ciornei, I. and Kyriakides, E. (2012) ‘Hybrid Ant Colony-Genetic Algorithm (GAAPI) for Global Continuous Optimization’, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42(1), pp. 234–245. doi: 10.1109/TSMCB.2011.2164245.

Coello Coello, C. A. (2006) ‘Evolutionary Multi-Objective Optimization: A Historical View of the Field’, IEEE Comput. Intell. Mag., 1(1), pp. 28–36. doi: 10.1109/MCI.2006.1597059.

Colmenares, G., Halal, F. & Zaremba, M.B., 2014. Ant Colony Optimization for Data Acquisition Mission Planning. Management and Production Engineering Review, 5(2), pp.3–11. doi:10.2478/mper-2014-0011.

Deb, K. (2000) ‘An efficient constraint handling method for genetic algorithms’, Computer Methods in Applied Mechanics and Engineering, 186(2–4), pp. 311–338. doi: 10.1016/S0045-7825(99)00389-8.

Deng, Y. and Hu, S. (2011) ‘Route optimization of multi-modal travel based on improved genetic algorithm’, Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE), pp. 1701–1704. doi: 10.1109/tmee.2011.6199539.

Dongbing Gu, D., Huosheng Hu, H., Reynolds, J. and Tsang, E. (2003) ‘GA-based learning in behaviour based robotics’, in 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No.03EX694). IEEE, pp. 1521–1526. doi: 10.1109/CIRA.2003.1222223.

Duan, H., Ma, R., Xu, X., Kong, F., Zhang, S., Kong, W., Hao, J. and Shang, L. (2009) ‘Two-decade reconstruction of algal blooms in China’s Lake Taihu’, Environmental Science and Technology, 43(10), pp. 3522–3528. doi: 10.1021/es8031852.

Ergezer, H. and Leblebicio, K. (2014) ‘3D Path Planning forMultiple UAVs for Maximum Information Collection’, J Intell Robot Syst (2014), 2014(73), pp. 737–762. doi: 10.1007/s10846-013-9895-6.

Floreano, D. and Mattiussi, C. (2008) Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies. MIT Press.

Gao, M. and Tian, J. (2007) ‘Path Planning for Mobile Robot Based on Improved Simulated Annealing’, in Third International Conference on Natural Computation, pp. 1–5.

García, A., Barrientos, A., Medina, A., Colmenarejo, P., Mollinedo, L. and Rossi, C. (2011) ‘3D Path planning using a fuzzy logic navigational map for Planetary Surface Rovers’, in Advanced Space Technologies for Robotics and Automation.

Garcia, M. A. P., Montiel, O., Castillo, O., Sepúlveda, R. and Melin, P. (2009) ‘Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation’, Applied Soft Computing, 9(3), pp. 1102–1110. doi: 10.1016/j.asoc.2009.02.014.

Giesbrecht, J. (2004) Global path planning for unmanned Ground Vehicles (U). Technical report, doi: 10.1016/0022-4898(80)90025-7.

Gower, J., King, S. and Goncalves, P. (2008) ‘Global monitoring of plankton blooms using MERIS MCI’, International Journal of Remote Sensing, 29(21), pp. 6209–6216. doi: 10.1080/01431160802178110.

Gu, D., Hu, H. and Spacek, L. (2003) ‘Learning fuzzy logic controller for reactive robot behaviours’, in Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003). IEEE, pp. 46–51. doi: 10.1109/AIM.2003.1225070.

Halal, F. (2007) Intelligent Hybrid Techniques in Mobile Robot Control. Ph. D. Thesis, ‘Politehnica’ University of Bucharest.

Halal, F. and Dumitrache, I. (2006) ‘Genetic Algorithm In Mobile Robot Control’, Journal of Control Engineering and Applied Informatics, 8(2), pp. 21–30.

Halal, F. and Dumitrache, I. (2007) ‘Evolutionary Learning of a Fuzzy Controller for a Mobile Robot’, U.P.B. Sci. Bull., Series C, 69(3), pp. 105–120.

Halal, F. and Zaremba, M. (2010) ‘Hybrid Geno-Fuzzy Control for Deliberative-Reactive Navigation’, in IFAC workshop on Intelligent Control Systems. Sinaia, Romania, pp. 45–51.

Halal, F. and Zaremba, M. B. (2015a) ‘Acquisition Platform Trajectory Generation for Environment Monitoring’, in IFAC 15 Symposium on Information Control in Manufacturing. Ottawa, pp. 220–221.

Halal, F. & Zaremba, M.B., (2015b). Multi-strategy spatial data acquisition missions using genetic algorithms. IFAC-PapersOnLine, 48(3), pp.778–783.

Halal, F. & Zaremba, M., (2017). Multi-Strategy Navigation for a Mobile Data Acquisition Platform Using Genetic Algorithms. Journal of Automation, Mobile Robotics & Intelligent Systems, 11(1), pp.30–41. doi: 10.14313/JAMRIS_1-2017/4

Hambrook Berkman, J. a. and Canova, M. G. (2007) ‘Algal Biomass Indicators’, in U.S. Geological Survey TWRI Book 9: Biological Indicators, pp. 1–86.

Hank, M. and Haddad, M. (2014) ‘Hybrid Control Architecture for Mobile Robots Navigation in Partially Known Environments’, in Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics. Science and and Technology Publications, pp. 513–521. doi: 10.5220/0005067305130521.

Haupt, R. L. and Haupt, S. E. (2004) Practical Genetic Algorithms. Hoboken, New Jersey.: John Wiley & Sons, Inc. doi: 10.1198/jasa.2005.s45.

Hofmann-Wellenhof, B., Legat, K. and Wieser, M. (2003) Navigation. 1st edn. Springer-Verlag Wien. doi: 10.1007/978-3-7091-6078-7.

Hsu, C.-C. and Liu, Y.-C. (2014) ‘Path planning for robot navigation based on Cooperative Genetic Optimization’, in Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control. IEEE, pp. 316–321. doi: 10.1109/ICNSC.2014.6819645.

Ibrahim, M. T. S., Ragavan, S. V. and Ponnambalam, S. G. (2009) ‘Way point based deliberative path planner for navigation’, IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, pp. 881–886. doi: 10.1109/AIM.2009.5229900.

Jenne, F. (2010) Simulation & Control Optimization of an autonomous sail boat, Master thesis. Swiss Federal Institute of Technology Zurich.

Karray, F. O. and De Silva, C. W. (2004) Soft Computing and Intelligent Systems Design: Theory, Tools and Applications. Addison Wesley, ISBN: 0321116178

Koponen, S., Vepsäläinen, J., Pulliainen, J., Kallio, K., Pyhälahti, T., Lindfors, A., Rasmus, K. and Hallikainen, M. (2005) ‘Using meris data for the retrieval of CHL-A, CDOM and TSS values in the gulf of Finland and lake Lohjanjärvi’, European Space Agency, (Special Publication) ESA SP, 2004(597), pp. 407–413.

Kponyo, J., Kuang, Y. and Zhang, E. (2014) ‘Dynamic Travel Path Optimization System Using Ant Colony Optimization’, in UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, p. 141. doi: 10.1109/UKSim.2014.44.

Kuo, J. Y. and Ou, Y. C. (2009) ‘An evolutionary fuzzy behaviour controller using genetic algorithm in robocup soccer game’, Proceedings - 2009 9th International Conference on Hybrid Intelligent Systems, HIS 2009, 1, pp. 281–286. doi: 10.1109/HIS.2009.63.

Lee-Johnson, C. P. and Carnegie, D. A. (2010) ‘Mobile Robot Navigation Modulated by Artificial Emotions’, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). IEEE, 40(2), pp. 469–480. doi: 10.1109/TSMCB.2009.2026826.

Lee, M.-L., Chung, H.-Y. and Yu, F.-M. (2003) ‘Modeling of hierarchical fuzzy systems’, Fuzzy Sets and Systems, 138, pp. 343–361. Available at: www.elsevier.com/locate/fss (Accessed: 18 July 2016).

Li, B., Chiong, R. and Gong, L. (2014) ‘Search-evasion path planning for submarines using the Artificial Bee Colony algorithm', 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 528–535. doi: 10.1109/CEC.2014.6900224.

Li, Y., Wang, Q., Wu, C., Zhao, S., Xu, X., Wang, Y. and Huang, C. (2011) 'Estimation of Chlorophyll a Concentration Using NIR/Red Bands of MERIS and Classification Procedure in Inland Turbid Water', IEEE Transactions on Geoscience and Remote Sensing, 50(3), pp. 988–997. doi: 10.1109/TGRS.2011.2163199.

Lincoln, N. K., Veres, S. M., Dennis, L. A., Fisher, M. and Lisitsa, A. (2013) 'Autonomous asteroid exploration by rational agents', IEEE Computational Intelligence Magazine, 8(4), pp. 25–38. doi: 10.1109/MCI.2013.2279559.

Liu, S. and You, X. (2009) 'On Multi-Behavior Based Multi-Colony Ant Algorithm for TSP', in 2009 Third International Symposium on Intelligent Information Technology Application. IEEE, pp. 348–351. doi: 10.1109/IITA.2009.464.

Lu Xu, L., Zhang, L.-G., Chen, D.-G. and Chen, Y.-Z. (2007) 'The Mobile Robot Navigation in Dynamic Environment', in 2007 International Conference on Machine Learning and Cybernetics. IEEE, pp. 566–571. doi: 10.1109/ICMLC.2007.4370209.

Luo, C., Yang, S. X., Krishnan, M., Paulik, M. and Chen, Y. (2013) 'A hybrid system for multi-goal navigation and map building of an autonomous vehicle in unknown environments', in 2013 IEEE International Conference on Robotics and Biomimetics, ROBIO, pp. 1228–1233. doi: 10.1109/ROBIO.2013.6739632.

Mai, N. A. and Janschek, K. (2012) 'Generic system architecture for behavior-based mobile robot control using fuzzy logic', in 2012 International Conference on Control, Automation and Information Sciences (ICCAIS). IEEE, pp. 253–258. doi: 10.1109/ICCAIS.2012.6466598.

Mali, A. D. (2002) 'On the behavior-based architectures of autonomous agency', IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 32(3), pp. 231–242. doi: 10.1109/TSMCC.2002.804445.

Min, H. and Dazhi, P. (2014) 'Improved Hybrid Behavior Ant Colony Algorithm to Solve the Vehicle Routing Problem', International Journal of Engineering Inventions (IJEI), 4(5), pp. 11–16. Available at: http://www.ijeijournal.com/pages/v4i5.html.

Morón Hernández, D. C., Díaz González, F. A., Triana Correa, J. S. and Pinzón Cabrera, P. R. (2016) 'Quality Evaluation of Chromatic Interpolation Algorithms for Image Acquisition System', Journal of Aerospace Technology and Management. Departamento de Ciência e Tecnologia Aeroespacial, 8(3), pp. 339–351. doi: 10.5028/jatm.v8i3.540.

Murphy, R. R. (2000) Introduction to AI robotics. Cambridge,Massachusetts London, England: MIT Press. doi: 10.1111/j.1464-410X.2011.10513.x.

Nakhaeinia, D., Tang, S. H., Mohd Noor, S. B. and Motlagh, O. (2011) 'A review of control architectures for autonomous navigation of mobile robots', International Journal of the Physical Sciences, 6(2), pp. 169–174. doi: 10.5897/IJPS10.540.

Narvydas, G., Simutis, R. and Raudonis, V. (2007) 'Autonomous Mobile Robot Control Using Fuzzy Logic and Genetic Algorithm', in 2007 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and

Applications. IEEE, pp. 460–464. doi: 10.1109/IDAACS.2007.4488460.

Nojima, Y. (2009) ‘Multi-objective behavior coordination based on sensory network for multiple mobile robots’, in 2009 IEEE Workshop on Robotic Intelligence in Informationally Structured Space. IEEE, pp. 66–72. doi: 10.1109/RIISS.2009.4937908.

Nolfi, S. and Floreano, D. (2000) Evolutionary robotics : the biology, intelligence, and technology of self-organizing machines. MIT Press. ISBN: 9780262140706.

O’Reilly, J. E., Maritorena, S., Mitchell, B. G., Siegel, D. A., Carder, K. L., Garver, S. A., Kahru, M. and McClain, C. (1998) ‘Ocean color chlorophyll algorithms for SeaWiFS’, J. Geophys. Res., 103(C11), pp. 24937–24953. doi: 10.1029/98JC02160.

Park, S. and Cho, J. (2013) ‘ROI-based visualization of spatial information for a remote-controlled robot’, 2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp. 234–235. doi: 10.1109/URAI.2013.6677352.

Powers, M. and Balch, T. (2009) ‘A Learning Approach to Integration of Layers of a Hybrid Control Architecture’, in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 893–898. doi: 10.1109/IROS.2009.5354386.

Pradhan, S. K., Parhi, D. R. and Panda, A. K. (2009) ‘Fuzzy logic techniques for navigation of several mobile robots’, Applied Soft Computing Journal, 9(1), pp. 290–304.

Raguraman, S. M., Tamilselvi, D. and Shivakumar, N. (2009) ‘Mobile robot navigation using Fuzzy logic controller’, in 2009 International Conference on Control, Automation, Communication and Energy Conservation. Perundurai, Tamilnadu, India: IEEE, pp. 1–5.

Rajappa, G. P. (2012) Solving Combinatorial Optimization Problems Using Genetic Algorithms and Ant Colony Optimization. Ph. D. thesis, University of Tennessee,Knoxville. Available at: http://trace.tennessee.edu/utk_graddiss/1478.

Rekik, C., Jallouli, M. and Derbel, N. (2009) ‘Optimal trajectory of a mobile robot by a Genetic Design Fuzzy logic controller’, in 2009 International Conference on Advances in Computational Tools for Engineering Applications. IEEE, pp. 107–111. doi: 10.1109/ACTEA.2009.5227926.

Rothlauf, F. (2006) Representations for Genetic and Evolutionary Algorithms. Springer-Verlag Berlin, Heidelberg. doi: 10.1007/3-540-32444-5.

Ruiz, E., Acuna, R., Velez, P. and Fern??ndez-L??pez, G. (2016) ‘Hybrid Deliberative Reactive Navigation System for Mobile Robots Using ROS and Fuzzy Logic Control’, Proceedings - 12th LARS Latin American Robotics Symposium and 3rd SBR Brazilian Robotics Symposium, LARS-SBR 2015 - Part of the Robotics Conferences 2015, pp. 67–72. doi: 10.1109/LARS-SBR.2015.24.

Samadi, M. and Othman, M. F. (2013) ‘Global Path Planning for Autonomous Mobile Robot Using Genetic Algorithm’, 2013 International Conference on Signal-Image Technology & Internet-Based Systems, pp. 726–730. doi: 10.1109/SITIS.2013.118.

Schubert, E., Zimek, A. and Kriegel, H. P. (2014) ‘Local outlier detection reconsidered:

A generalized view on locality with applications to spatial, video, and network outlier detection', Data Mining and Knowledge Discovery, 28(1), pp. 190–237. doi: 10.1007/s10618-012-0300-z.

Sedighi, K. H., Ashenayi, K., Manikas, T. W., Wainwright, R. L. and Heng-Ming Tai, H.-M. (2004) 'Autonomous local path planning for a mobile robot using a genetic algorithm', in Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753). IEEE, pp. 1338–1345. doi: 10.1109/CEC.2004.1331052.

Senthilkumar, K. S. and Bharadwaj, K. K. (2009) 'Hybrid Genetic-fuzzy approach to Autonomous Mobile Robot', in 2009 IEEE International Conference on Technologies for Practical Robot Applications. IEEE, pp. 29–34. doi: 10.1109/TEPRA.2009.5339649.

Siegwart, R. & Nourbakhsh, I.R., (2004). Introduction to Autonomous Mobile Robots, MIT Press, p. 321.

Singh, A., Krause, A., Guestrin, C. and Kaiser, W. J. (2009) 'Efficient Informative Sensing using Multiple Robots', Journal of Artificial Intelligence Research, 34, pp. 707–755. doi: 10.1613/jair.2674.

Sun, T. Y., Huo, C. L., Tsai, S. J., Yu, Y. H. and Liu, C. C. (2011) 'Intelligent flight task algorithm for unmanned aerial vehicle', Expert Systems with Applications. Elsevier Ltd, 38(8), pp. 10036–10048. doi: 10.1016/j.eswa.2011.02.013.

Topliss, B. J. and Piatt, T. (1986) 'Passive fluorescence and photosynthesis in the ocean: implications for remote sensing', Deep Sea Research, 33(7), pp. 849–864.

Tunstel, E. (1996) 'Mobile Robot Autonomy via Hierarchical Fuzzy Behavior Control', in Proceedings of 6th Intl Symp on Robotics & Manuf, . Montpellier, France, pp. 837–842. doi: 10.1.1.53.3192.

Tyler, A. N., Svab, E., Preston, T., Présing, M. and Kovács, W. A. (2006) 'Remote sensing of the water quality of shallow lakes: A mixture modelling approach to quantifying phytoplankton in water characterized by high-suspended sediment', International Journal of Remote Sensing. Taylor & Francis , 27(8), pp. 1521–1537. doi: 10.1080/01431160500419311.

Vachtsevanos, G., Tang, L., Drozeski, G. and Gutierrez, L. (2005) 'From mission planning to flight control of unmanned aerial vehicles: Strategies and implementation tools', Annual Reviews in Control, 29(1), pp. 101–115. doi: 10.1016/j.arcontrol.2004.11.002.

Vadakkepat, P., Miin, O. C., Peng, X. and Lee, T. H. (2004) 'Fuzzy Behavior-Based Control of Mobile Robots', IEEE Transactions on Fuzzy Systems. IEEE, 12(4), pp. 559–564. doi: 10.1109/TFUZZ.2004.832536.

Veera Ragavan, S., Ponnambalam, S. G. and Sumero, C. (2011) 'Waypoint-based path planner for mobile robot navigation using PSO and GA-AIS', in 2011 IEEE Recent Advances in Intelligent Computational Systems, pp. 756–760. doi: 10.1109/RAICS.2011.6069411.

Vincent, R. K., Qin, X., Michael, R., Mckay, L., Miner, J., Czajkowski, K., Savino, J. and Bridgeman, T. (2004) 'Phycocyanin detection from LANDSAT TM data for mapping cyanobacterial blooms in Lake Erie', Remote Sensing of Environment, 89(3), p. 381392. doi: 10.1016/j.rse.2003.10.014.

Watanabe, T. (2009) 'Implementation of fuzzy Q-learning based on modular fuzzy model and parallel structured learning', Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, (October), pp. 1338–1344. doi: 10.1109/ICSMC.2009.5346250.

Xiao-Wen Terry Liu (2005) An Intuitive and Flexible Architecture for Intelligent Mobile Robots. Ph. D. thesis, The University of Manitoba.

Yan Yongjie and Zhang Yan (2009) 'The collision avoidance planning in multi-robot system by genetic fuzzy control algorithm', 2008 IEEE International Conference on Robotics and Biomimetics, pp. 1020–1025. doi: 10.1109/ROBIO.2009.4913140.

Yang, X., Member, S., Moallem, M. and Patel, R. V (2005) 'A Fuzzy Logic-based Reactive Navigation Algorithm for Mobile Robots', in IEEE Conference on Control Applications. Toronto, pp. 197–202.

Yongjie, Y., Qidan, Z. and Chengtao, C. (2006) 'Hybrid Control Architecture of Mobile Robot Based on Subsumption Architecture', in 2006 International Conference on Mechatronics and Automation, pp. 2168–2172. doi: 10.1109/ICMA.2006.257629.

Yoshikawa, M. and Terai, H. (2009) 'Car Navigation System Based on Hybrid Genetic Algorithm', in 2009 WRI World Congress on Computer Science and Information Engineering, pp. 62–65. doi: 10.1109/CSIE.2009.558.

Yu, L. and Cai, Z. (2009) 'Robot exploration mission planning based on heterogeneous interactive cultural hybrid algorithm', in 5th International Conference on Natural Computation, ICNC 2009, pp. 583–587. doi: 10.1109/ICNC.2009.15.

Yun, S. C., Parasuraman, S. and Ganapathy, V. (2011) 'Dynamic path planning algorithm in mobile robot navigation', Industrial Electronics and Applications (ISIEA), 2011 IEEE Symposium on, pp. 364–369. doi: 10.1109/ISIEA.2011.6108732.

Zaremba, M., Halal, F., Hirose, T. and Pedrocca, P. (2015) 'Satellite Guided Navigation Control for Environment Monitoring', Annual Review of Earth and Planetary Sciences, 24, pp. 129–144. doi: 10.1007/978-3-319-13853-4-12.

Zein-Sabatto, S., Sekmen, A. and Koseeyaporn, P. (2003) 'Fuzzy Behaviors for Control of Mobile Robots', Systemics, Cybernetics and Informatics, 1(1), pp. 68–74.

Zeng, C., Zhang, Q. and Wei, X. (2011) 'Robotic global path-planning based modified genetic algorithm and A* algorithm', in 3rd International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2011, pp. 167–170. doi: 10.1109/ICMTMA.2011.613.

Zhang, F. and Leonard, N. E. (2005) 'Generating contour plots using multiple sensor platforms', in 2005 IEEE Swarm Intelligence Symposium, SIS 2005. IEEE, pp. 319–326. doi: 10.1109/SIS.2005.1501637.

Author's List of Relevant Publications

Halal, F. & Zaremba, M., 2017. Multi-Strategy Navigation for a Mobile Data Acquisition Platform Using Genetic Algorithms. *Journal of Automation, Mobile Robotics & Intelligent Systems*, 11(1), pp.30–41. doi: 10.14313/JAMRIS_1-2017/4

Zaremba, M. et al., 2015. Satellite Guided Navigation Control for Environment Monitoring. *Annual Review of Earth and Planetary Sciences*, 24, pp.129–144.

Halal, F. & Zaremba, M.B., 2015. Multi-strategy spatial data acquisition missions using genetic algorithms. *IFAC-PapersOnLine*, 48(3), pp.778–783. Available at: http://linkinghub.elsevier.com/retrieve/pii/S2405896315004164 [Accessed June 15, 2016].

Halal, F., Zaremba, M.B., 2015. Acquisition Platform Trajectory Generation for Environment Monitoring, *15th symposium on information control problem in manufacturing*, pp, 220-222, Ottawa.

Colmenares, G., Halal, F. & Zaremba, M.B., 2014. Ant Colony Optimization for Data Acquisition Mission Planning. *Management and Production Engineering Review*, 5(2), pp.3–11.

Halal, F. & Zaremba, M.B., 2014. Deliberative control for satellite-guided water quality monitoring. In *CIVEMSA 2014 - 2014 IEEE Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications, Proceedings*. pp. 100–105.

Halal, F. et al., 2014. Remote-sensing based adaptive path planning for an aquatic platform to monitor water quality. In *2014 IEEE International Symposium on Robotic and Sensors Environments (ROSE) Proceedings*. IEEE, pp. 43–48. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6952981 [Accessed June 17, 2016].

Halal, F. & Zaremba, M., 2010. Hybrid Geno-Fuzzy Control for Deliberative-Reactive Navigation. In *IFAC workshop on Intelligent Control Systems*. Sinaia, Romania, pp. 45–51.

Halal, F., 2007. *Intelligent Hybrid Techniques in Mobile Robot Control*. *Ph.D. Thesis*, "Politehnica" University of Bucharest.

Halal, F. & Dumitrache, I., 2007. Evolutionary Learning of a Fuzzy Controller for a Mobile Robot. *U.P.B. Sci. Bull., Series C*, 69(3), pp.105–120.

Halal, F. & Dumitrache, I., 2006. Genetic Algorithm in Mobile Robot Control. *Journal of Control Engineering and Applied Informatics*, 8(2), pp.21–30.

Appendices

Appendix 1

Satellite sensor data

1.1. MODIS

MODIS spectral domain is divided into the four following spectral regions which are as follows:

- Visible (VIS) (0.412 to 0.551 μm),
- Near infrared (NIR) (0.650 to 0.940 μm),
- Short wavelength/medium wavelength infrared (SWIR/MWIR) (1.240 to 4.565 μm),
- Long wavelength infrared (LWIR) (6.715 to 14.235μm).

MODIS has 36 spectral bands with center wavelengths ranging from 0.412 μm to 14.235 μm;
MODIS spectral bands are divided into 5 categories depending on its application as follows

1. Land and cloud boundaries/property bands,
2. Ocean colour bands,
3. Atmosphere/cloud bands,
4. Thermal bands,
5. Thermal bands for cloud height & fraction.

Where band 1 and band 2 are imaged at a nominal resolution of 250m at nadir, band 3 to band 7 are imaged at a nominal resolution of 500m. The remaining bands are imaged at a nominal resolution of 1000m. Bands 13 and 14 each have two gain settings, 13 low, 13 high, 14 low, and 14 high, telemetered from the instrument. All bands are telemetered at 12 bits.

Each spectral region has a lens assembly for imaging scene energy onto the corresponding focal plane.

Each focal plane consists of rows of detectors aligned in along of the track direction so as to image 10km in along of track direction of the scan. Then

- There are 10 detectors along the track in the 1000m bands,
- 20 detectors along the track in the 500m bands,
- 40 detectors along the track in the 250m bands.

The ground track direction is defined as the sensor movement and is designated as the x+ direction. The scan direction is orthogonal to the track direction. MODIS swath is 2200 km long and 10 km wide at nadir. MODIS needs 1.471second to scan the swath. A swath contains 1354 frames per scan. A frame size is 1km in the scan direction and 10km in the track direction at nadir.

MODIS SPECTRAL BANDS SPECIFICATION

LAND AND CLOUD BOUNDARIES/PROPERTIES BANDS

BAND	λ	IFOV	Bandwidth	PURPOSE
1	645 nn	250 m	50 nn	Veg.
2	858 nn	250 m	35 nn	Cloud
3	469 nn	500 m	20 nn	Soil,
4	555 nn	500 m	20 nn	Green
5	1240 nn	500 m	20 nn	Leaf/Canopy
6	1640 nn	500 m	24.6 nn	Snow/Cloud
7	2130 nn	500 m	50 nn	Land

OCEAN COLOUR BANDS

BAND	λ	IFOV	Bandwidth	PURPOSE
8	412 nn	1000 m	15 nn	Chlorophyll
9	443 nn	1000 m	10 nn	Chlorophyll
10	488 nn	1000 m	10 nn	Chlorophyll
11	531 nn	1000 m	10 nn	Chlorophyll
12	551 nn	1000 m	10 nn	Sediments
13	667 nn	1000 m	10 nn	Sediments,
14	678 nn	1000 m	10 nn	Chlorophyll
15	748 nn	1000 m	10 nn	Aerosol
16	869 nn	1000 m	15 nn	Aerosol/Atmospheric

ATMOSPHERE/CLOUD BANDS

BAND	λ	IFOV	Bandwidth	PURPOSE
17	905 nn	1000 m	30 nn	Cloud/Atmospheric
18	936 nn	1000 m	10 nn	Cloud/Atmospheric
19	940 nn	1000 m	50 nn	Cloud/Atmospheric

THERMAL BANDS

BAND	λ	IFOV	Bandwidth	PURPOSE
20	3.75 μm	1000 m	0.18 μm	Sea
21	3.96 μm	1000 m	0.059 μm	Forest
22	3.96 μm	1000 m	0.059 μm	Cloud/Surface
23	4.05 μm	1000 m	0.061 μm	Cloud/Surface
24	4.47 μm	1000 m	0.065 μm	Tropospheric
25	4.52 μm	1000 m	0.067 μm	Tropospheric
26	1375 μm	1000 m	30 μm	Cinfraredrus
27	6.72 μm	1000 m	0.36 μm	Mid-Tropospheric
28	7.33 μm	1000 m	0.30 μm	Upper-Tropospheric
29	8.55 μm	1000 m	0.30 μm	Surface
30	9.73 μm	1000 m	0.30 μm	Total ozone
31	11.03 μm	1000 m	0.50 μm	Cloud/Surface Temperature

THERMAL BANDS

BAND	λ	IFOV	Bandwidth	PURPOSE
32	12.02 μm	1000 m	0.50 μm	Cloud Height & Surface Temperature
33	13.34 μm	1000 m	0.30 μm	Cloud Height & Fraction
34	13.64 μm	1000 m	0.30 μm	Cloud Height & Fraction

35	13.94 µm	1000 m	0.30 µm	Cloud Height & Fraction
36	14.24 µm	1000 m	0.30 µm	Cloud Height & Fraction

1.2. MERIS

MERIS is a programmable, medium-spectral resolution, imaging spectrometer operating in the solar reflective spectral range.

The satellite's motion provides scanning in the along-track direction using linear CCD arrays to provide spatial sampling across the track.

The instrument's 68.5° field of view around nadir covers a swath width of 1150 km. This wide field of view is shared between five identical optical modules arranged in a fan shape configuration. MERIS is carried aboard the ESA's Envisat satellite.

Technical Characteristics (ESA)

- Accuracy: Ocean colour bands typical S:N = 1700
- Spatial Resolution: Ocean: 1040m x 1200 m, Land & coast: 260m x 300m.
- Swath Width: 1150 km, global coverage every 3 days
- Waveband: VIS-NIR: 15 bands selectable across range: 390 nm to 1040 nm (bandwidth programmable between 2.5 and 30 nm)

This system has 15 spectral bands with center wavelengths ranging from the 390 - 1040 nm which represent the visible and near infrared part of the electromagnetic spectral. The following table lists represent MERIS lists spectral bands and their potential applications.

Band	Band centre (nm)	Potential Applications
1	412.5	Yellow substance, turbidity
2	442.5	Chlorophyll absorption maximum
3	490	Chlorophyll, other pigments
4	510	Turbidity, suspended sediment, red tides
5	560	Chlorophyll reference, suspended sediment
6	620	Suspended sediment
7	665	Chlorophyll absorption
8	681.25	Chlorophyll fluorescence
9	705	Atmospheric correction, red edge
10	753.75	Oxygen absorption reference
11	760	Oxygen absorption R-branch
12	775	Aerosols, vegetation
13	865	Aerosols corrections over ocean
14	890	Water vapour absorption reference
15	900	Water vapour absorption, vegetation

Appendix 2

Selected MATLAB Code

3.1. Modified Genetic Algorithm Tool Box

```
function [x,endPop,bPop,traceInfo] =
ga(evalFN,startPop,termFN,termOps,selectFN,selectops,xOverFNs,xOverOps
,mutFNs, mutOps)
% GA run a genetic algorithm
% function
[x,endPop,bPop,traceInfo]=ga(bounds,evalFN,evalOps,startPop,opts,
%
termFN,termOps,selectFN,selectOps,
%
xOverFNs,xOverOps,mutFNs,mutOps)
%
% Output Arguments:
%   x            - the best solution found during the course of the
run
%   endPop       - the final population
%   bPop         - a trace of the best population
%   traceInfo    - a matrix of best and means of the ga for each
generation
%
% Input Arguments:
%   bounds       - a matrix of upper and lower bounds on the variables
%   evalFN       - the name of the evaluation .m function
%   evalOps      - options to pass to the evaluation function ([NULL])
%   startPop     - a matrix of solutions that can be initialized
%                  from initialize.m
%   opts         - [epsilon prob_ops display] change required to
consider two
%                  solutions different, prob_ops 0 if you want to
apply the
%                  genetic operators probabilisticly to each solution,
1 if
%                  you are supplying a deterministic number of
operator
%                  applications and display is 1 to output progress 0
for
%                  quiet. ([1e-6 1 0])
%   termFN       - name of the .m termination function
(['maxGenTerm'])
%   termOps      - options string to be passed to the termination
function
%                  ([100]).
%   selectFN     - name of the .m selection function
(['normGeomSelect'])
%   selectOpts   - options string to be passed to select after
%                  select(pop,#,opts) ([0.08])
%   xOverFNS     - a string containing blank seperated names of
Xover.m
%                  files (['arithXover heuristicXover simpleXover'])
%   xOverOps     - A matrix of options to pass to Xover.m files with
the
```

```
%                  first column being the number of that xOver to
perform
%                  similiarly for mutation ([2 0;2 3;2 0])
%   mutFNs       - a string containing blank seperated names of
mutation.m
%                  files (['boundaryMutation multiNonUnifMutation ...
%                           nonUnifMutation unifMutation'])
%   mutOps       - A matrix of options to pass to Xover.m files with
the
%                  first column being the number of that xOver to
perform
%                  similiarly for mutation ([4 0 0;6 100 3;4 100 3;4 0
0])

% Binary and Real-Valued Simulation Evolution for Matlab
% Copyright (C) 1996 C.R. Houck, J.A. Joines, M.G. Kay
%
% C.R. Houck, J.Joines, and M.Kay. A genetic algorithm for function
% optimization: A Matlab implementation. ACM Transactions on
Mathmatical
% Software, Submitted 1996.
%
% This program is free software; you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation; either version 1, or (at your option)
% any later version.
%
% This program is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
% GNU General Public License for more details. A copy of the GNU
% General Public License can be obtained from the
% Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139,
USA.



nn=1
 opts = [1e-6 1 0];

n=nargin;

%if n<3 %Default evalation opts.
 % evalOps=[];
%end

%evalFN = 'COST_FUNCTION_M_G';
evalFN = 'COST_FUNCTION_N_C_1';
if any(evalFN<48) %Not using a .m file
  if opts(2)==1 %Float ga
    e1str=['x=c1; c1(xZomeLength)=', evalFN ';'];
    e2str=['x=c2; c2(xZomeLength)=', evalFN ';'];
  else %Binary ga
    e1str=['x=b2f(endPop(j,:),bounds,bits);
endPop(j,xZomeLength)=',...
    evalFN ';'];
  end
else %Are using a .m file
  if opts(2)==1 %Float ga
    e1str=['[c1 c1(xZomeLength)]=' evalFN '(c1,[gen evalOps]);'];
```

```
    e2str=['[c2 c2(xZomeLength)]=' evalFN '(c2,[gen evalOps]);'];
  else %Binary ga
    e1str=['x=b2f(endPop(j,:),bounds,bits);[x v]=' evalFN ...
    '(x,[gen evalOps]); endPop(j,:)=[f2b(x,bounds,bits) v];'];
  end
end


if n<4 %Default termination information
  termOps=[50];
  termFN='maxGenTerm';
end
if n<9 %Default muatation information
  if opts(2)==1 %Float GA
  mutFNs=[ 'multi_point_Mutate_4_4_N'];
    mutOps=[4 0 0];
   end
end

if n<7 %Default crossover information
  if opts(2)==1 %Float GA
    xOverFNs=['arithXover heuristicXover multi_point_Xover_1_S'];
    xOverOps=[2 0;2 3;2 0];
  end
end
if n<6 %Default select opts only i.e. roullete wheel.
  selectOps=[];
end
%Default select info
  selectFN=['normGeomSelect'];
  selectOps=[0.08];

if n<4 %Default termination information
  termOps=[500];
  termFN='maxGenTerm';
end
if n<2 %No starting population passed given
  startPop=[];
end


xOverFNs=parse(xOverFNs);
mutFNs=parse(mutFNs);

xZomeLength  = size(startPop,2);    %Length of the
xzome=numVars+fittness
numVar       = xZomeLength-1;       %Number of variables
popSize      = size(startPop,1);    %Number of individuals in the pop
endPop       = zeros(popSize,xZomeLength); %A secondary population
matrix
c1           = zeros(1,xZomeLength);    %An individual
c2           = zeros(1,xZomeLength);    %An individual
numXOvers    = size(xOverFNs,1);    %Number of Crossover operators
?????????????????????????
numMuts      = size(mutFNs,1) ;     %Number of Mutation operators
??????????????????????????????
epsilon      = opts(1);                 %Threshold for two fittness to
differ
oval         = max(startPop(:,xZomeLength)); %Best value in start pop
```

```
bFoundIn     = 1;           %Number of times best has changed
done         = 0;                        %Done with simulated evolution
gen          = 1;           %Current Generation Number
collectTrace = (nargout>3);         %Should we collect info every gen
floatGA      = opts(2)==1;               %Probabilistic application of
ops
display      = opts(3);                  %Display progress



%modify
%%%%%%%%%%%%%%

sort_endPop = zeros(popSize,xZomeLength) ;          %NEW SORTED
POPULATION
%xover_Pop   = zeros((2*numXOvers),xZomeLength)    %new crossover
solution    %%%%%%%%%%%%%%%%%%%%%%
mut_Pop     = zeros((2*numMuts),xZomeLength);      %new new mutation
solution
xOver_nm    = floor(0.4 * (popSize/2)) *2;    %crossover function
(option )
mut_nm      = floor(0.05 * popSize)*2 ;           %mutation function
(option )
xover_Pop   = zeros(xOver_nm,xZomeLength);
mut_Pop     = zeros(mut_nm,xZomeLength);
TOTAL_SOL   = popSize + xOver_nm + mut_nm ;
sort_pop    = zeros(TOTAL_SOL,xZomeLength );
adap_Pop     = zeros(20,xZomeLength);

for i = 1:popSize
startPop(i,xZomeLength)= COST_FUNCTION_N_C_1(startPop(i,:));
end
%%%%%%%%%%%%%%%%

sameFitnessValue = 0;
bValAnt = -1;

while(~done && sameFitnessValue < 100)
    gen
  %Elitist Model
  [bval,bindx] = max(startPop(:,xZomeLength)); %Best of current pop
  best =  startPop(bindx,:);

  if collectTrace
    traceInfo(gen,1)=gen;                            %current
generation
    traceInfo(gen,2)=startPop(bindx,xZomeLength);    %Best fittness
    traceInfo(gen,3)=mean(startPop(:,xZomeLength));  %Avg fittness
    traceInfo(gen,4)=std(startPop(:,xZomeLength));
  end

  if ( (abs(bval - oval)>epsilon) | (gen==1)) %If we have a new best
sol
    if display
      fprintf(1,'\n%d %f\n',gen,bval);          %Update the display
    end
    if floatGA
      bPop(bFoundIn,:)=[gen startPop(bindx,:)]; %Update bPop Matrix
    else
```

```
      bPop(bFoundIn,:)=[gen
b2f(startPop(bindx,1:numVar),bounds,bits)...
      startPop(bindx,xZomeLength)];
    end
    bFoundIn=bFoundIn+1;                      %Update number of
changes
    oval=bval;                                %Update the best val
  else
    if display
      fprintf(1,'%d ',gen);               %Otherwise just update num
gen
    end
  end

  endPop = feval(selectFN,startPop,[gen selectOps]); %Select



%%%%%%%%%%%%%%%%%% modify
  for i = 1:popSize
      [X,Y]= max (endPop(:,xZomeLength));
      sort_endPop(i,:) = endPop(Y,:);
      endPop(Y,xZomeLength)= 0;
 end
 % sort_endPop = sortrows(endPop, - xZomeLength);
 endPop= sort_endPop;

 %%%%%%%%%%%%%%%

     %The parents for crossover are from the Elite solution
     for i= 1:xOver_nm

           a = randi([1 (popSize* 0.2)],1,1) ;        %Pick a  good
parent
           b = randi([1 (popSize* 0.2)],1,1) ;        %Pick another
good parent
          % multi_point_Mutate4_4_PN
           %(endPop(a,:));
           %(endPop(b,:));
                      %Get the name of crossover function
           [c1 c2] = simpleXover (endPop(a,:),endPop(b,:));
            xover_Pop((2*i-1),:) = c1;
            xover_Pop((2*i),:)   = c2;

     end



      %The parents for crossover are randomly selected from the
solution

        for j= 1:xOver_nm


           a = randi([1 popSize],1,1) ;           %Pick a random
parent
           b = randi([1 popSize],1,1);            %Pick another random
parent
          [c1 c2] = multi_point_Xover_1_S(endPop(a,:),endPop(b,:));
```

```
            xover_Pop((2*(j) -1),:) = c1;
            xover_Pop((2*(j)),:)    = c2;
         end


         %evalute crossover solution (fitness function)
           xover_pop= size (xover_Pop,1);
           for i= 1:xover_pop

               xover_Pop(i,xZomeLength)= COST_FUNCTION_N_C_1
(xover_Pop(i,:));
           end




      % The parent for mutation is from the Elite solution
     for i=1:mut_nm,
          a = randi([1 (popSize* 0.2)],1,1) ;

          c1 =  multi_point_Mutate_4_4_PN (endPop(a,:));

          %c1 = feval(deblank(mutFNs(i,:)),endPop(a,:),[gen
mutOps(i,:)]);
         mut_Pop(i,:)= c1;
     end

     %The parent for mutation is randomly selected from the solution
     for i=1:mut_nm,
         a = randi([1 popSize],1,1) ;
         c1 =  multi_point_Mutate_4_4_PN (endPop(a,:));
         %c1 = feval(deblank(mutFNs(i,:)),endPop(a,:),[gen
mutOps(i,:)]);
         mut_Pop(i+numMuts,:)= c1;
     end

     %evalute mutation solution (fitness function)
     M_pop= size (mut_Pop,1);
     for i = 1 :M_pop

         mut_Pop(i,xZomeLength)= COST_FUNCTION_N_C_1  (mut_Pop(i,:));
     end




%%%%%%%%   create_new population




big_generation = vertcat(endPop,xover_Pop,mut_Pop);
endPop;
xover_Pop;
mut_Pop;

 big_g = sortrows(big_generation, - xZomeLength);
```

```
 % TOTAL_SOL= size (big_generation,1);
  %for i= 1:TOTAL_SOL
     % [X,Y]= max (big_generation(:,xZomeLength));
      %sort_pop(i,:) = big_generation(Y,:);
      %big_generation(Y,xZomeLength)= 0;
  %end


   % create new  POPULATION  (GLOBAL PATH) new adaptive search
  for i = 1:30
     % adap_pop(i,:)= adaptive_search_3_behaviour_R (big_g(i,:));
    adap_pop(i,:)= adaptive_search (big_g(i,:));
    adap_pop(i,xZomeLength)  = COST_FUNCTION_N_C_1(adap_pop(i,:));

 end




 all_generation = vertcat(adap_pop,big_g);
 big_gene = sortrows(all_generation, - xZomeLength);
 endPop= big_gene(1:popSize,:);


% endPop= big_g(1:popSize,:);

  %ant colony optimisation
     % AC_POP_NEW  = ACO_optimize_1( AC_POP)


  %%%% new optimised population
 % opt_pop =vertcat (sort_pop,AC_POP_NEW)


  %%%%%%% sort and slect the new optimised population

  %total_pop= size (opt_pop,1)
  %for i= 1:total_pop
     % [X,Y]= max (opt_pop(:,xZomeLength));
     % new_pop(i,:) = opt_pop(Y,:)
     % opt_pop(Y,xZomeLength)= 0;
 % end

  % endPop= sort_pop(1:popSize,:);


  gen=gen+1;
  done=feval(termFN,[gen termOps],bPop,endPop); %See if the ga is done
  startPop=endPop;          %Swap the populations

   %%%%end while%

[bval,bindx] = max(startPop(:,xZomeLength));
```

```
if(bval == bValAnt)
    sameFitnessValue = sameFitnessValue +1;
else
    sameFitnessValue = 0;
end
bValAnt = bval;

if display
  fprintf(1,'\n%d %f\n',gen,bval);
end

%%%%%%%%%%%%%%%%%%%%%%%%%


%%%reactive control

%for i= 1:2
     % reactive_eval(i,:)= startPop(i,:)
     % reactive_eval(i,xZomeLength)  = reactive( reactive_eval(i,:));
%end


    % [X,Y]= max (reactive_eval(:,xZomeLength));
    % reactive_best(nn,:)= reactive_eval(Y,:);
    % nn= nn+1;

    end   %%end while%




x=startPop(bindx,:);
if opts(2)==0 %binary
  x=b2f(x,bounds,bits);
  bPop(bFoundIn,:)=[gen b2f(startPop(bindx,1:numVar),bounds,bits)...
      startPop(bindx,xZomeLength)];
else
  bPop(bFoundIn,:)=[gen startPop(bindx,:)];
end




gen
A=startPop(bindx,xZomeLength);
B=(startPop(:,xZomeLength));
if collectTrace
  traceInfo(gen,1)=gen  ;       %current generation
  traceInfo(gen,2)=startPop(bindx,xZomeLength)  ;  %Best fittness
  traceInfo(gen,3)=mean(startPop(:,xZomeLength));  %Avg fittness
end
% [X,Y]= max (reactive_best);
%reactive_best_solution=(reactive_best(Y,:)

plot (traceInfo(:,1),traceInfo(:,2))
hold on
plot (traceInfo(:,1),traceInfo(:,3))
%reactive_best
```

3.2. Cost Function

```
function [F] = objectivefunction (genom, params)
%COST_FUNCTION AND NO CONSTRAIN
%path generator
%SAMPLES VALUES
%TSS& MCI MISSION
load ('MCI_N_G.mat');
load ('TSS_N_G.mat');
%load ('TSS_ROI_VI_ZONE.mat');
%load ('TSS_N_ROI_3.mat');
%load ('TSS_N_ROI_4.mat');
load ('G_ROI.mat')





maximum_samples_number=40;  %  the longest chrommosome
muximum_trip_distance=350000;% metre
MEAN_VELOCITY = 40000;             % m/ hour
real_pexel_dis=300;
%TSS= zeros(26,1);
TS_V  = 0;
TSS_V = 0;
SAMPLE_COLLECTION = 3;           %  3 minute
maximum_SAMPLING_TIME = 120;  % 2 hours
TSS_MCI_V=0;
DIS =0;
PDS =0;
m=0;
G_ROI_AWARD= 0;
%genom;
geno=genom;


% SAMPLING TIME FACTOR
 nrgene = length(geno) / 2;
 n=1;
    for  j= 1:nrgene
        x = geno(2*j-1);
        y = geno(2*j);


        if  ( x  ~= 0 && y  ~= 0 );

          new_G(2*n-1)= geno(2*j-1);
          new_G(2*n)=geno(2*j);
          n=n+1;
     end
    end
    n=n-1;
    sampling_time = (n*(SAMPLE_COLLECTION))/ 60;
    ST= ((n*(SAMPLE_COLLECTION))/(  maximum_SAMPLING_TIME));
    %ST= 1 -((n*(SAMPLE_COLLECTION))/(  maximum_SAMPLING_TIME))^2;
%for time
```

```
    %constraint exp.


    %  ONother formula for sampling time
    %  ST = 1- (( n*SAMPLE_COLLECTION)/ maximum_SAMPLING_TIME)


geno = new_G;


% SAMPLE VALUE (THE TRIP GAIN)
nrgene = length(geno) / 2;    % length's chromozome
%nrgene= genom(201);
    for  j= 1:nrgene;
        x = geno(2*j-1);
        y = geno(2*j);
         TS_V = TSS_N_G(x,y)+ MCI_N_G(x,y);
         %TSS_MCI(j)=TS_V;
       TSS_MCI_V= TSS_MCI_V+ TS_V;
       %TSS_MCI_F = TSS_MCI_V;
       TSS_MCI_F = (TSS_MCI_V/maximum_samples_number);


    end

% ROT AWARD

 for  j= 1:nrgene;
         x = geno(2*j-1);
         y = geno(2*j);


      if  G_ROI(x,y)== 1 && TSS_N_G(x,y)> 0.3 && MCI_N_G(x,y)> 0.5 ;
          MCI_TSS_ROI_AWARD = 1;
      elseif  G_ROI(x,y)== 1 && TSS_N_G(x,y)< 0.3 && MCI_N_G(x,y)>
0.5;
           MCI_TSS_ROI_AWARD = 0.9;
      elseif  G_ROI(x,y)== 1 && TSS_N_G(x,y)> 0.3 && MCI_N_G(x,y)<
0.5;
           MCI_TSS_ROI_AWARD = 0.9;
      else
         MCI_TSS_ROI_AWARD = 0;
      end

      G_ROI_AWARD = G_ROI_AWARD +MCI_TSS_ROI_AWARD;

    end

    ROI_AWARD_FT= (G_ROI_AWARD/maximum_samples_number);



%THE SHORTEST DISTANCE HAS HIGHER VALUE



  local_path_factor= 0;
```

```
    for j= 1:nrgene-1;
        DIS = sqrt ((genom(2*j+1)-genom(2*j-1)).^2 + (genom(2*j+2)-
genom(2*j)).^2);
        PDS=PDS+DIS;
      % if DIS < 20
         %   local_path_factor = local_path_factor+1;

      % end
    end
        total_local_path_factor
=(local_path_factor/(2*maximum_samples_number));
    Path_length= PDS ;   % the path lenght
    real_path_length= Path_length * real_pexel_dis;
    path_length_value =( real_path_length /( muximum_trip_distance));
     %path_length_value =(1-(( real_path_length /( 1.6 *
muximum_trip_distance)))^2);
  % path_length_value = -(real_path_length/  muximum_trip_distance)^2;
    % TRAVEL TIME
   TRAVEL_TIME =  real_path_length / MEAN_VELOCITY;
   MISSION_time = sampling_time + TRAVEL_TIME;

if ( real_path_length > 350000 || MISSION_time > 12)
   F = 0;
else

F = TSS_MCI_F + path_length_value + ST+ ROI_AWARD_FT ;
en
```

3.3. Adaptive Search 3 Zones

```
%prune and reproduce algorithm
function [F] = objectivefunction (genom, params)
load ('MCI_N_G.mat');
load ('TSS_N_G.mat');
load ('NEW_G_ROI.mat');
load ('MG_P.mat');
load ('MCI_ROI.mat');
load ('M_G_ROI.mat');
load ('TSS_ROI_3.mat');
load ('M_G_2.mat');

 max_local_path_MCI= 15;
 x_newValue= zeros (25 ,1);
 y_newValue=zeros  (25 ,1);
 chv=zeros (25 ,1);
 MG=zeros  (25 ,1);
 CHTSS=zeros (25 ,1);



geno=genom;
```

```
G_ROI = NEW_G_ROI;


nrgene = length(geno) / 2;
 n=1;
    for  j= 1:nrgene;
        x = geno(2*j-1);
        y = geno(2*j);


        if  ( x  ~= 0 && y  ~= 0 );

          new_G(2*n-1)= geno(2*j-1);
          new_G(2*n)=geno(2*j);
          n=n+1;
     end
    end
    n=n-1;





geno= new_G;

% prun the way point which fall autside the ROI ('G_ROI.)EXCLUDING the
% start and the target point

k=1;
prun_s(2*k-1)= geno(1);
prun_s(2*k)=geno(2);

k=2;

for  j= 2:n-1;
        x = geno(2*j-1);
        y = geno(2*j);


        if G_ROI(x,y)>0 ;

          prun_s(2*k-1)= geno(2*j-1);
          prun_s(2*k)=geno(2*j);
          k=k+1;
     end
end

prun_s(2*k-1)= geno(2*n-1);
prun_s(2*k)=geno(2*n);

%Introducing new waypoint randomly in the ROI

 pro_s= prun_s;

 m = n-k ;
```

```
 m = floor (m);

  l= k-1;

 for p= 1: m;
       p=p+1;
       j = randi([2 l],1,1);



for s= j+1: k;
          pro_s(2*(s+1)-1) = prun_s(2*s-1);
          pro_s(2*(s+1))   =prun_s(2*s);

end


    x = pro_s(2*j-1);
    y = pro_s(2*j);

if MCI_ROI(x,y)>0
  M= 1 ;
for M=1:200 ;
      M=M+1;
      times = 1;
s=1;
while s<2

      x_newValue(M) = randi([x-8 x+8],1, 1);        % Now mutate that
point
      y_newValue(M) = randi([y-10 y+10],1, 1);


if MCI_ROI(x_newValue(M),y_newValue(M))>0;
    s=s+1;
     dis=sqrt ((x_newValue(M)-x).^2 + (y_newValue(M)-y).^2);
    chv(M)= (1- (0.5*(dis/ max_local_path_MCI)))*MCI_N_G(x,y);

end

  times= times + 1;

   if times > 120;
   break;
   end
end
end
   k=k+1;

  [X,b]= max (chv);

   pro_s(2*(j+1)-1)  = x_newValue(b);             % Make the child
   pro_s(2*(j+1))    = y_newValue(b);                 % Make the child
   prun_s= pro_s;
```

```
   elseif M_G_ROI(x,y)>0
  M1= 1;
for M1=1:200 ;
      M1=M1+1;
      times = 1;
      s=1;
while s<2

      x_newValue(M1) = randi([x-3 x+3],1, 1);        % Now mutate that
point
      y_newValue(M1) = randi([y-5 y+5],1, 1);

if M_G_ROI(x_newValue(M1),y_newValue(M1))>0;
    s=s+1;
    MG(M1)= M_G_2(x_newValue(M1),y_newValue(M1));

end

  times= times + 1;

  if times > 120
  break;
  end

end
end
   k=k+1;
   [X b ]= max (MG);
   pro_s(2*(j+1)-1)  = x_newValue(b);             % Make the child
   pro_s(2*(j+1))    = y_newValue(b);                 % Make the child
   prun_s= pro_s;



   else TSS_ROI_3(x,y)>0
  M2= 1;
for M2=1:200;
      M2=M2+1;
      times = 1;
      s=1;
while s<2

      x_newValue(M2) = randi([x-8 x+8],1, 1);        % Now mutate that
point
      y_newValue(M2) = randi([y-10 y+10],1, 1);


if TSS_ROI_3(x_newValue(M2),y_newValue(M2))>0;
    s=s+1;
  dis=sqrt ((x_newValue(M2)-x).^2 + (y_newValue(M2)-y).^2);
  CHTSS(M2)= ((dis / max_local_path_MCI)*(MCI_N_G(x,y)+
TSS_N_G(x,y)));

end
```

```
times= times + 1;

   if times > 120
   break;
   end
end
end

   k=k+1;
   [X b ]= max (CHTSS);
   pro_s(2*(j+1)-1)  = x_newValue(b);               % Make the child
   pro_s(2*(j+1))    = y_newValue(b);                   % Make the child
   prun_s= pro_s;



end

end


   % Introducing new waypoints in the neighbour of the maximum samples
in the ROI



for f= 1:k-1
     genom(f)=pro_s(f);
end


 %genom  = pro_s;
 F= genom;
```

Appendix 3

3.1. Articles

Satellite Guided Navigation Control for Environment Monitoring

Marek Zaremba, Fadi Halal, Thomas Hirose and Pablo Pedrocca

Abstract This paper addresses issues inherent to the design of navigation control systems required for adaptive acquisition of in situ reference data for environment monitoring systems using satellite imagery. The development is motivated by the application to adaptive inland water sampling by mobile platforms for an autonomous algal blooms observing and prediction system. The sampled field, used to derive optimal paths for the mobile platforms equipped with measurement sensors, is defined as a multi-objective spatial function. Conflicting demands, introduced by resource demands and management of uncertainties, are discussed. A hybrid control approach is presented, where the navigation planning module supervised the reactive navigation. Due to the tasks complexity, the control architecture features fuzzy system modules which handle different control strategies. A fuzzy selector is used to select the appropriate system response depending on the surrounding environment, in order to deal with conflicting control scenarios. The versatility of the proposed system makes its application possible for the control of mobile platforms of a different degree of autonomy.

Keywords Satellite imagery · Navigation control · Remote sensing · Path planning · Environment monitoring · Sampleacquisition

1 Introduction

The algal blooming in inland lakes and in coastal waters has become a critically important issue for its impacts not only on local natural and social environments, but also on global human community. Authorities responsible for water quality,

M. Zaremba (✉) · F. Halal · P. Pedrocca
Department of Computer Science and Engineering, Université du Québec en Outaouais, Gatineau, QC J8Y 3G5, Canada
e-mail: zaremba@uqo.ca

T. Hirose
Noetix Research Inc, Ottawa, ON K1S 2E1, Canada

J. Sąsiadek (ed.), *Aerospace Robotics II*, GeoPlanet: Earth and Planetary Sciences,
DOI 10.1007/978-3-319-13853-4_12

environmental protection, economic development and public health must develop and implement plans and strategies for prediction and mitigation the effects of algal blooms. This requires means to detect and monitor the occurrence of the blooms. Modeling of the underlying phenomena that lead to algal blooms is complex and is a subject of ongoing research (Duan et al. 2009). Remote detection techniques provide significant advantages over ground-based monitoring in terms of spatial and temporal coverage and cost-efficiency. In Vincent et al. (2004) a set of algorithms was developed to derive phycocyanin, Chlorophyll-a, and sediment for detecting of blue-green algal blooms in lake Erie based on Landsat ETM images; A model to quantify chlorophyll-a in Lake Balaton using Landsat ETM imagery was discussed in Tyler et al. (2006). The existing algorithms depend on water quality and remote sensing sensors. There are currently no available algorithms which are suitable for most of inland waters and remote sensing sensors with minimal modification. The core approach to the Harmful Algal Blooms (HAB) detection in this project was automatic analysis of multi-temporal multi-spectral image sequences, mostly from MODIS and MERIS multi-spectral sensors. An overall objective was to develop adaptive models, the development of which calls for the application of machine learning techniques. Calibration of the parametric models as well as the training of statistical models using machine learning requires the availability of reference data obtained by in situ data collection. A major technical and theoretical problem that has to be resolved to develop statistically viable models is the scarcity of the reference data. Acquisition of the reference data is usually a costly and time-consuming process. In the application area addressed by this project, it implies a data collection by a specially equipped mobile platform, such as a cruise ship, a glider or a floating robot. Our study uses information obtained from in situ measurements performed for Lake Winnipeg in Canada. Critical to this research are reliable, efficient, and adaptive control strategies that ensure mobile sensor platforms collect data of greatest value. In addition, a large size of the lake, the tenth largest lake in the world, and the use of a large ship for data acquisition missions make the development of optimal navigation control important.

In this context, we propose a hybrid control, proposed earlier for vision guided mobile robot navigation (Halal 2007), which combines deliberative path planning level with local reactive control along trajectories dependent on selected local strategies. The navigation is determined by the surrounding environment and by the data acquisition task using multi-source data. Fuzzy reactive navigation strategies execute the global path by dividing this path into many local paths. At the same time, only one fuzzy selector system can drive the mobile platform, making a decision on the appropriate behavior for a specific local path.

The structure of the paper is as follows. Section 2 presents the detection of the concentration of chlorophyll from medium-resolution satellite imagery. After the discussion of local acquisition conditions in Sect. 3, the hybrid navigation control approach in introduced in Sect. 4. A more detailed description of the control scheme is presented in Sect. 5.

2 Chlorophyll Concentration from Satellite Imagery

A major technical challenge is the fact that the detection of the chlorophyll-a concentration—the main indicator of algal blooms—is a multimodal process, i.e., the clusters of high concentration are located in different areas of the feature space defined by spectral distributions of the reflected light. Consequently, no single model will capture the intrinsic relationship between the chlorophyll-a concentration and the combinations of satellite image features with sufficient accuracy. This modality issue has been successfully resolved in this project by introducing an additional stage of processing remote sensing data, which is the classification of water characteristics, and the development of adaptive non-parametric models for both the classification and regression tasks. Based on the results of the classification, an appropriate model is automatically selected for the assessment of the chlorophyll concentration. The same type of data processing architecture can be applied to the assessment of other environmentally important lake characteristics, such as the Total Solid Sediment (TSS) concentration or the Dissolved Organic Carbon (DOC) concentration. The influence of chlorophyll content on spectral shape is depicted on Fig. 1.

Estimation of the chlorophyll-a concentration is typically obtained by using indices that exploit chlorophyll absorption/reflectance wavelengths (Topliss and Platt 1986). The Fluorescent Line Height (FLH) Index developed for MODIS wavelengths is illustrated in Fig. 2.

In our study, we augmented the FLH model by using a set of models, with additional indices, dedicated for specific water conditions. In the case of MODIS data, these indices are given below:

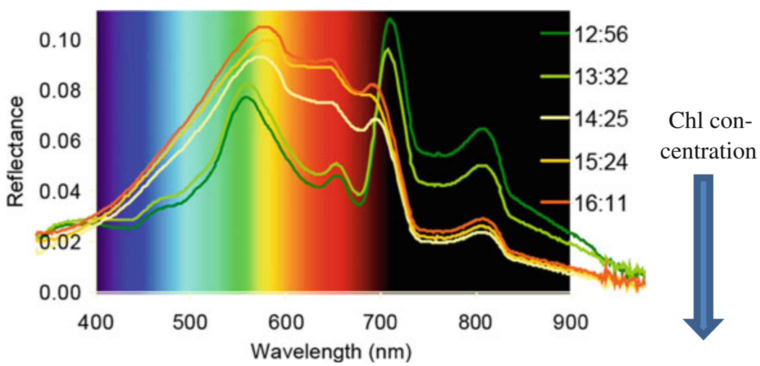


Fig. 1 Spectral signature of chlorophyll content

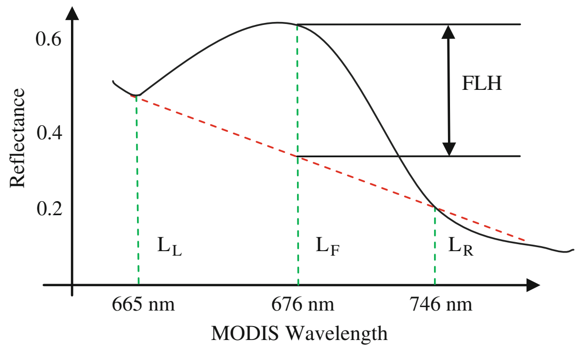


Fig. 2 Definition of the FLH index

$$
\begin{aligned}
C_1 &= \frac{R_{rs}(\lambda 7)}{R_{rs}(\lambda 8)} = \frac{R_{rs}(\lambda_{555})}{R_{rs}(\lambda_{645})}\\
C_2 &= \frac{R_{rs}(\lambda 6)}{R_{rs}(\lambda 4)} = \frac{R_{rs}(\lambda_{551})}{R_{rs}(\lambda_{488})}\\
C_3 &= \frac{R_{rs}(\lambda 8)}{R_{rs}(\lambda 9)} = \frac{R_{rs}(\lambda_{645})}{R_{rs}(\lambda_{667})}\\
C_4 &= \frac{R_{rs}(\lambda 7)}{R_{rs}(\lambda 9)} = \frac{R_{rs}(\lambda_{555})}{R_{rs}(\lambda_{667})}\\
C_5 &= \frac{R_{rs}(\lambda 7)R_{rs}(\lambda 11)}{R_{rs}(\lambda 8)R_{rs}(\lambda 10)} = \frac{R_{rs}(\lambda_{555})R_{rs}(\lambda_{748})}{R_{rs}(\lambda_{645})R_{rs}(\lambda_{678})}
\end{aligned}
\tag{1}
$$

In the set of Eq. (1), C_n represents the index used in the classification process, R_{rs} is the reflectance value, and λ_{xyz} represents the waveband centered at the frequency of *xyz* nanometers. This frequency corresponds to the MODIS band λX (for example, $\lambda 7 = \lambda_{555}$).

The classification was performed using multi-class Support Vector Machine (SVM) algorithms with polynomial kernel of degree $d = 4$ and parameters $\alpha = 0.2$ and $\beta = 5$.

$$K(\mathbf{x}, \mathbf{x}') = (\alpha \mathbf{x}^{\mathbf{T}} \mathbf{x} + \beta)^d \tag{2}$$

The learning precision was 97.17 %, and the testing one was 92.10 %. Figure 3 depicts the flow diagram for the selection of the optimal model and the procedure for calculating chlorophyll concentration. Model parameters are obtained in a training process, and subsequently applied at the operational stage. Figure 4 compares the results obtained using FLH index (Fig. 4a) and those using FLH and band ratios developed for the type of water with higher concentration of solid sediments (Fig. 4b). The error of the chlorophyll assessment decreases by an order of magnitude.

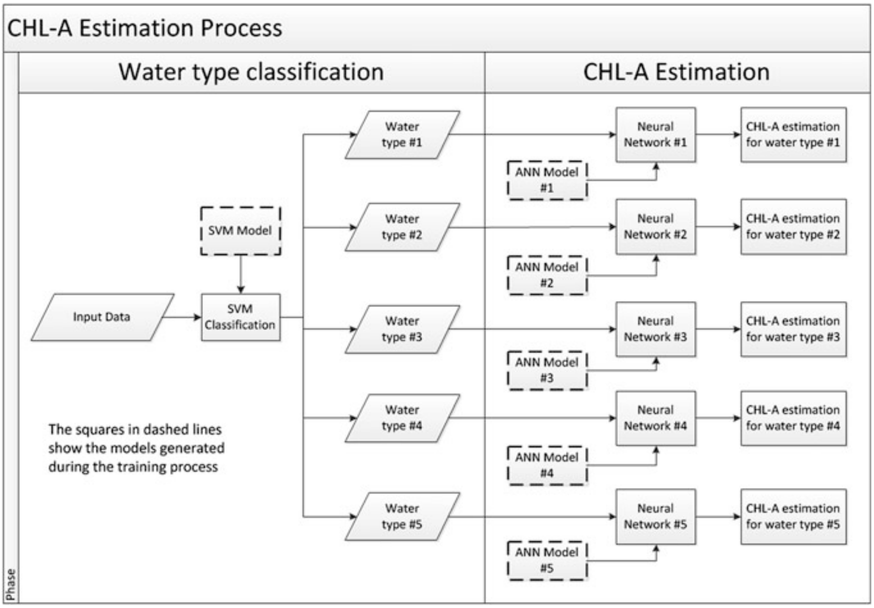


Fig. 3 Flow diagram for multi-model assessment of chlorophyll concentration

An example of the distribution of chlorophyll concentration in Lake Winnipeg obtained using the Maximum Chlorophyll Index method is shown in Fig. 5.

3 Local Data Acquisition

The in situ data included air pressure, wind speed, wind direction, temperature, water transparency, water temperature, PH, salinity, dissolved oxygen (DO), Chemical Oxygen Demand (COD), active phosphates, nitrites, ammonium salt, phytoplankton types and concentration, zooplankton types and concentration, and Chlorophyll-a.

Our goal is to design a mobile sampling network to take measurements of scalar and vector fields and collect the best data set (Chen and Cheng 2008). A cost function, or sampling metric, must be defined in order to give meaning to the term optimal data set. For example, the performance metric that we consider in this paper defines an optimal data set as one in which the total error of the estimate of the chlorophyll concentration field is minimized. Complementary metrics can be utilized, emphasizing the sampling of regions of highest dynamic variability or focus on areas of high sensibility to the degradation of water quality.

An example of a trajectory of ship equipped with the data acquisition equipment and a laboratory setup for measuring water quality parameters is shown in Fig. 6.

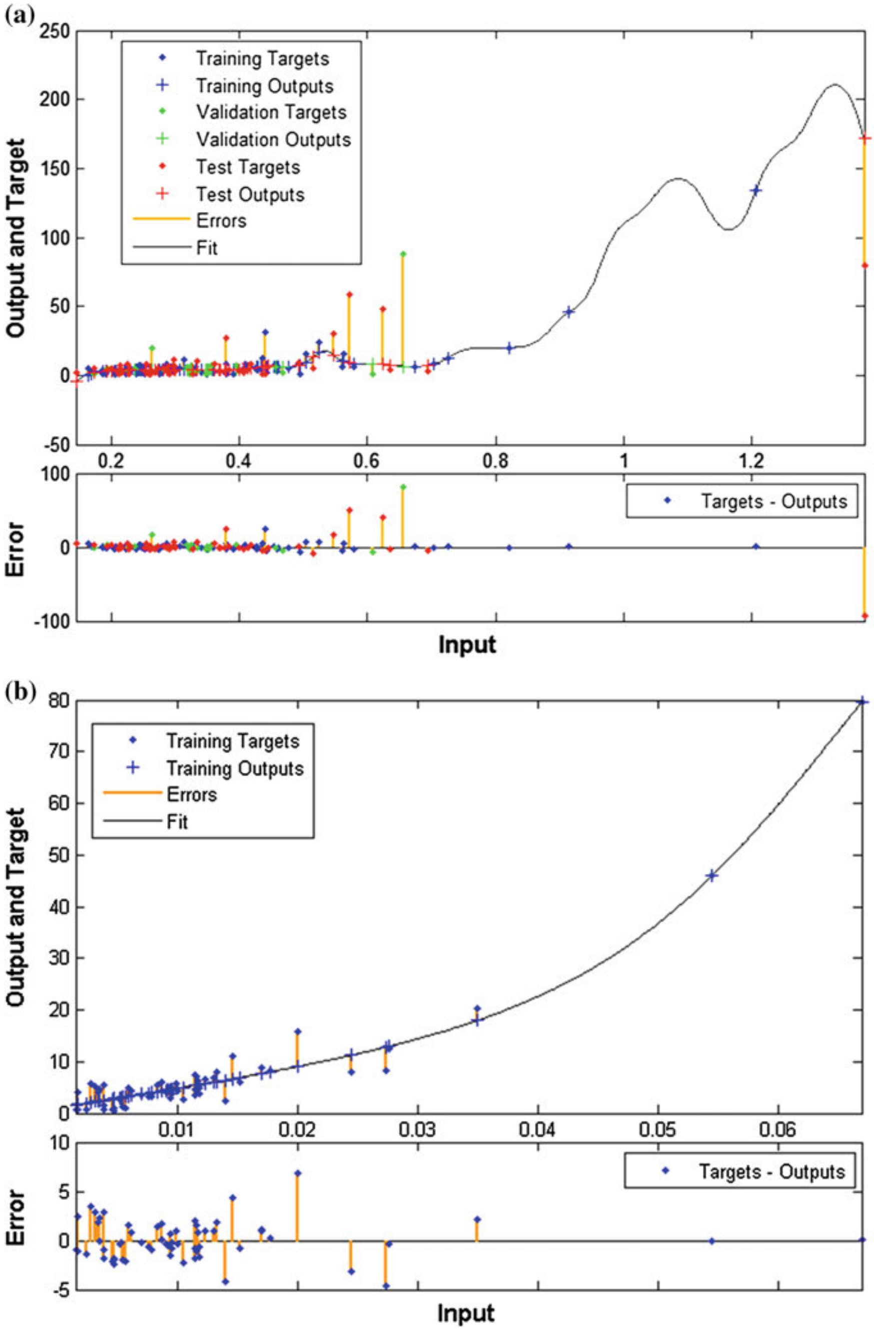


Fig. 4 Chlorophyll concentration assessment: **a** FLH index, **b** enhanced FLH index

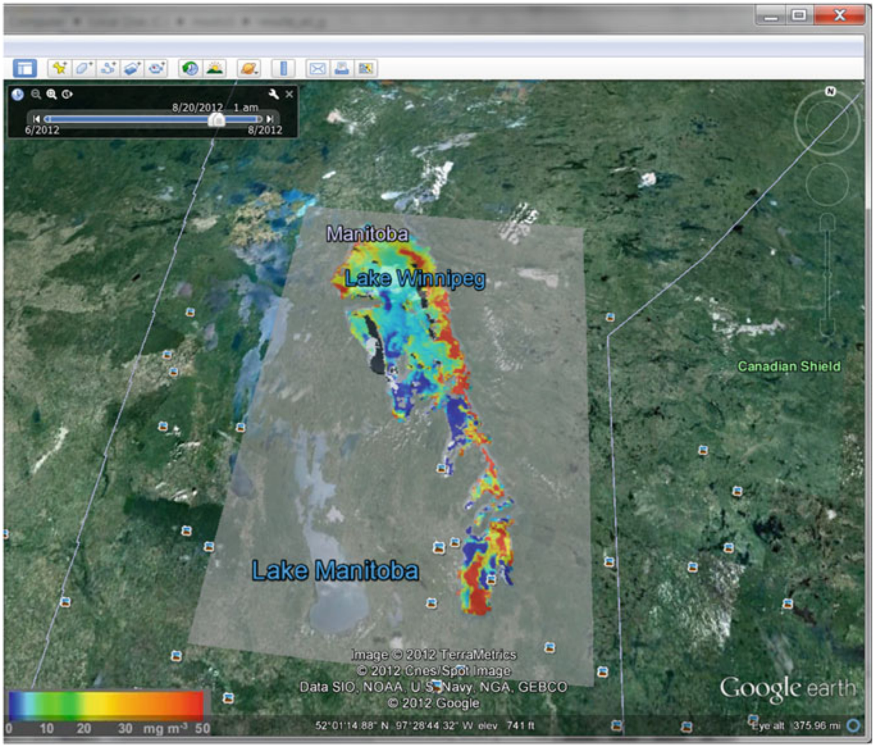
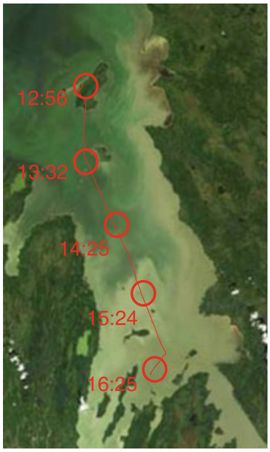


Fig. 5 Chlorophyll concentration in Lake Winnipeg (*Noetix Research*)

Fig. 6 Trajectory of the mobile sample acquisition platform



The specifications of the ship for dynamics simulation (Benedict 2003) are the following: gross tonnage: 328 tons, cruising speed: 12 knots, length: 33.62 m, breadth: 8.53 m, and draft: 2.13 m.

4 Control Scheme

4.1 Requirements

The navigation control scheme has to address a number of challenges and constraints in order to assure the central objective of the optimized in situ data collection. Conflicting demands, introduced by resource demands and management of uncertainties, require tradeoffs. The design methodology should also allow for autonomous operation of the water observation and prediction system and its easy extension to different architectures of the data acquisition platform. Major issues include the following.

1. *Multiple sampling fields*

As discussed in the preceding section, there is more than one field to be sampled simultaneously. Consequently, a choice needs to be made as to how to weight the importance of different fields in the sampling strategy. Apart from the distribution of the chlorophyll, DOC, and TSS fields, the sampling process is also guided by environment features, such as water turbidity, which are taken into account in the acquisition trajectory planning process. The sampling field issue also involves the selection of the metric that should be defined to obtain the best and richest data set. Gradient climbing strategies (Marthaler and Bertozzi 2003; Zhang and Leonard 2005) are an effective way to enable a mobile sensor to track and sample boundaries of phytoplankton patches in a chlorophyll concentration field.

2. *Multiple scales*

The issue of different spatial scales arises in reference to the resolution of measurement sensors and sampling fields as well as to the sampling strategy. As defined by the resolution of the satellite, the spatial scale can range from under one meter per pixel for high-resolution satellites (QuickBird, Ikonos) down to over 1 km per pixel (SeaWIFS). In the context of our study, the spatial scale corresponds to the resolution of medium-resolution satellites (MODIS, MERIS). In terms of sampling strategies, such feature-tracking strategies as gradient climbing strategies are particularly useful for sampling at relatively small spatial scales. Strategies that provide synoptic coverage are best suited for larger spatial scales. In this case, the goal is typically to minimized error in the estimate of the field of interest over the region in space and time, without taking redundant measurements.

3. *Obstacle configuration*

The classical obstacle avoidance problem has to be resolved by considering both hard obstacles in the form of islands, coastal areas, ships and other floating objects, and soft obstacles in the form of haze or cloud patches. Obstacles that result in a reduced visibility affect the navigation strategy at a different level of importance, and are a major requirement for the application of fuzzy logic components in the control architecture.

4. *Multiple sources of incertitude*

The major sources of incertitude are:

- Errors of chlorophyll concentration estimation;
- Top of Atmosphere (TOA) reflectance measurements;
- The Sun and observing geometry;
- Motion of moving obstacles;
- Atmospheric scattering and atmospheric absorption;
- Haze and cloud cover.

Certain sources of the incertitude, for example the atmospheric effects, can be mitigated by precise correction algorithms. The impact of some sources, especially the moving obstacles, has to be dealt with at the level of the control algorithm. The uncertainty introduced by haze can only partly be corrected by dedicated algorithms, and to a large extent affects the navigation strategy.

4.2 General Architecture

A pure reactive control system is responsive, flexible and robust while the deliberative planning system has slow responsiveness and abstract representational knowledge. The deliberative planning systems and purely reactive control systems have their limitations. Then hybrid deliberative-reactive robotic architecture can combine the above characteristics. Table 1 compares the two robot control approaches (Orgen 2003). The global path is long term plan and consists of many segments of local path which maybe vary in the shape or the behaviors. The reactive navigation (Belkhouche 2009) is responsible to execute the local paths

Table 1 Comparison of the planning and the reactive control

Planning	Reactive
World model dependent	World model free
Slower response	Real-time response
High level AI	Low level AI
Variable latency	Fast and simple computations

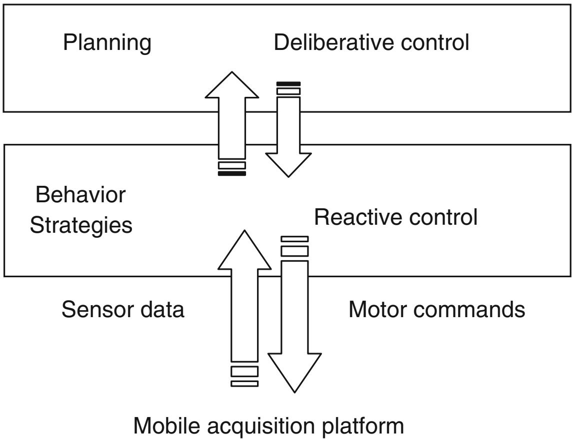


Fig. 7 Hybrid architecture for navigation control

using many behaviors strategies depending on the system. A fuzzy behaviors selector is used to select the appropriate behavior depending on the surrounding environment conditions.

Figure 7 illustrates the hybrid deliberative-reactive robotic architecture.

The deliberative control level supervises the reactive control and plans the global path from the start point to the target point. The reactive control module executes the global path into many local paths which do not necessarily match the global path, due to real time conditions or unpredicted changes in the surrounding environment (Halal and Zaremba 2010).

5 Deliberative/Reactive Control

5.1 Control Model

A functional diagram depicting the deliberative/reactive control of the cruise ship is shown in Fig. 8.

The planning module generates a global path using multi-band satellite images. Many sources of data are used to interpret the surrounding environment. The bathymetric map allows the ship to navigate safely preventing the ship from any accident with the lake floor. The MODIS chlorophyll-a map provides a good water classification which helps the planning module to generate a global trajectory to optimize the ship gathering samples.

The genetic algorithm controls the fuzzy parameters. We used genetic algorithm to control the fuzzy rules base structures. The genetic algorithm has a supervised role on the fuzzy systems which have the same input and output variable and the same membership function set. By changing the fuzzy rule base the robot can perform different kind of tasks. A combination of genetic search with the fuzzy system permits to handle different behaviors, including conflicting behaviors.

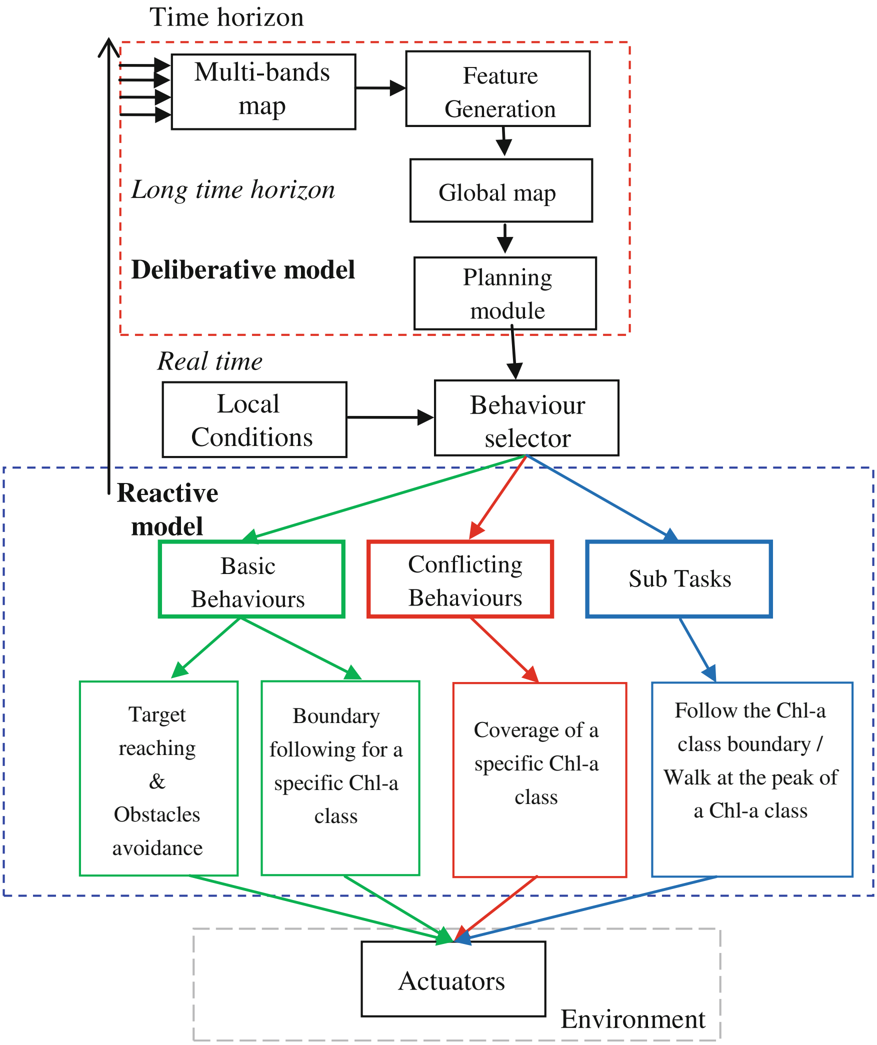


Fig. 8 Navigation control system

Examples of specific types of behavior:

- *Basic Behavior*
  - Avoid shallow areas
    If (Bathymetric is Shallow) and (Heading is Hnegative) and (Obstacle is Far) and (target is Hnegative) then (Power is Lbackward) (Steering is Hnegative)
  - Avoid obstacles
    If (Bathymetric is Deep) and (Heading is Lnegative) and (Obstacle is close) then (Power is Slow) (Steering is Hpositive)

- *Conflicting Behavior* (navigate inside a specific chlorophyll class)
  - If (Bathymetric is Average) and (Heading is Heading) and (Class is Close) then (Power is Fast) (Steering is Forward)

The basic behavior allowed to robot to gather samples from the maximum gradient concentration simultaneously satisfying such constraints as the water depth. This strategy can be compared to the wall following, where the wall is equivalent to a specific class boundary. When the robot gather samples from areas such as zones covered by haze or to enter inside a specific chlorophyll class, a conflicting behavior strategy has to be followed. Additional expertise on the interpretation of the level of precision of the remote sensing data and the local geographical and meteorological conditions have to be incorporated in the decision process.

5.2 Path Planning

The path planning system generates an optimal path with the goal of maximizing the number and the value of the collected samples during the acquisition mission. In order to obtain the best and richest data set, an appropriate metric should be defined over the sampling field. Based on the acquisition strategies and the sample types, such as the types of chemical component, pollution types, phytoplankton and zooplankton types and concentration, the samples were divided into M classes. The number N and location of the samples are determined by maximizing the following function.

$$C = \sum_{j=1}^{M} \left(\sum_{i=1}^{N} \left(V_i^j + \alpha(d) D_{i,i+1}^j - T_{i,i+1}^j \right) - T^{j,j+1} \right) \quad (3)$$

where V signifies the sample value, D is the distance between the consecutive sample acquisition locations, α is a scaling factor, and T represents the cost of the ship travel between the different class regions.

An example of the acquisition path is shown in Fig. 9. The Lake Winnipeg samples were divided into four classes. The first two classes represent the maximum concentration and maximum gradient metrics applied to the chlorophyll samples. The other two classes are defined for Total Suspended Sediment (TSS) and Dissolvent Organic Carbon (DOC) samples. Initially, the path direction aims at DOC type area (The TSS type zone is too distant, and the sampling cost is prohibitive). The reactive control level takes over the control of trajectory generation in the vicinity of the moving obstacle. After the acquisition of the DOC type samples, the ship proceeds to the acquisition of chlorophyll samples applying the strategy of maximum gradient following.

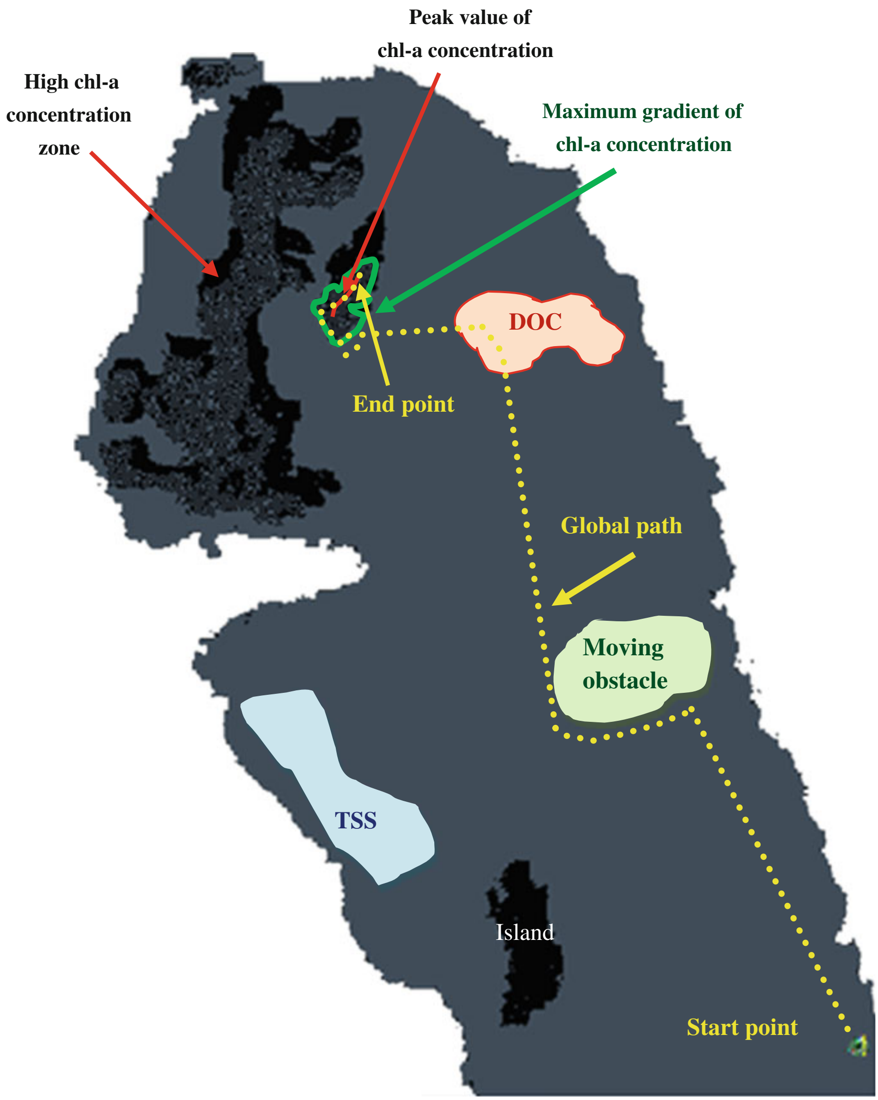


Fig. 9 Optimized global path

5.3 Implementation Issues

This paper discusses mainly the situation where the input images are obtained in the form of general-purpose satellite imagery, not readily available in real time. Those images are subsequently used by the path planner and the reactive controller. In the situation when the vision data come from dynamically moving platforms such as drones, the saliency of the incoming data becomes an important issue. The control system has to quickly obtain the relevant information, at the right spatial scale, and at the right location.

The Attentive Vision Technology (AVT) offers automated object detection based on adaptive, multi-scale descriptor extraction and application-specific fusion of multimodal and multi-temporal imagery (Palenychka and Zaremba 2012). The scientific underpinning of AVT is formed by a combination of computational derivatives of the Theory of Visual Attention (Nieburand and Koch 1997) with Machine Learning procedures. An iterative learning process that aims at defining optimal spatial filters for salient object detection is depicted in Fig. 10. Introduction of multi-scale and multi-component attention operators, such as MIMF or SIFT allows the user to extract attention (feature) points from image sequences serving as feature points for object recognition in reliable and computationally efficiently manner. The primitive features are obtained through a statistical learning process using ground truth data.

Figure 11 illustrates a saliency map generated by the filtering procedures. The consecutive maxima of the resulting spatial form can be used to quickly localize the position of the target points (maximum values or maximum gradient values of the water pollutants) in Fig. 9.

The core AVT technology is designed to a great extent generic, and can be adapted to a variety of applications, mainly those involving fast and adaptive detection of events of interest (e.g., abnormality, fault, novelty, alarm, etc.).

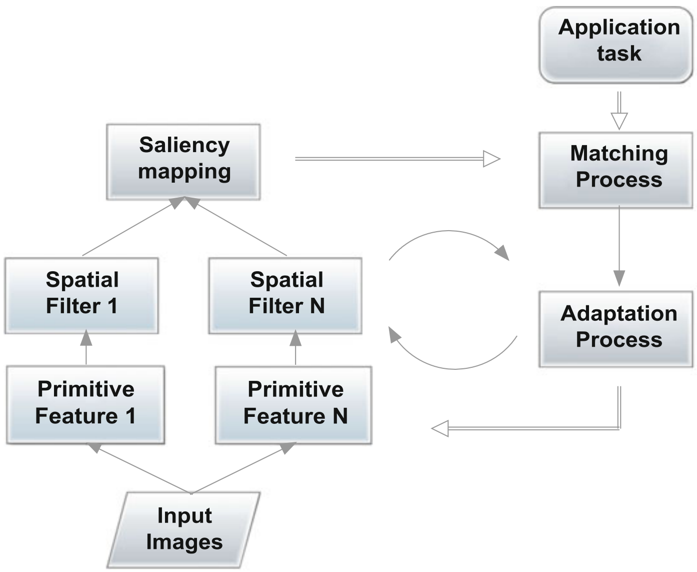
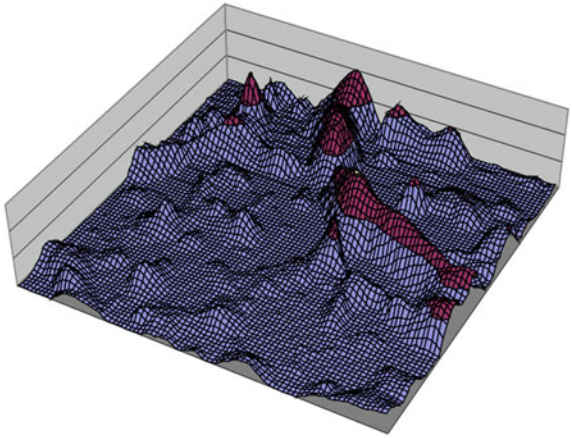


Fig. 10 Adaptive saliency filter bank

Fig. 11 Saliency map



6 Conclusions

The problem of environment monitoring using satellite imagery was investigated in this paper in a comprehensive way by addressing the detection of pertinent environment features, in this case the chlorophyll concentration in inland waters, and the acquisition of reference data required for non-parametric learning and calibration of the environment feature models. This paper proposes a hybrid navigation approach using deliberative navigation to supervise and control reactive navigation at a local level. Hybrid navigation has been shown to be a more robust and flexible solution for complex and dynamic environment, recovering the cruise ship trajectory from dead zones and cycling modes. This system provided a suitable degree of reactivity and deliberation in a mobile platform providing the optimal global path and optimal sub-global trajectories. Geno-fuzzy system is able to control successfully the behaviour strategies in the presence of conflicted tasks in a dynamic and complex environment. It can also be adapted to mobile sensor networks, comprised of a fleet of sensor-equipped autonomous vehicles, to monitor large areas with time-varying, spatially distributed fields.

Acknowledgments The authors acknowledge support of the International Science and Technology Program (ISTP Canada), project CCRD09-102, and the Natural Science and Engineering Council of Canada, NSERC-I2I program.

References

Belkhouche F (2009) Reactive Path Planning in a Dynamic Environment. IEEE Trans Robot 25 (4):902–911

Benedict K, Baldauf M, Felsenstein CM, Kirchhoff M (2003) Computer-based support for the evaluation of ship handling simulator exercise results. In: International conference on marine simulation and ship manoeuvrability, Kanazawa, Japan

Chen C, Cheng Y (2008) Research on map building by mobile robots. In: 2nd international symposium on intelligent information technology application. IEEE. doi:10.1109/IITA.2008.205
Duan H, Ma R, Xu X, Kong F, Zhang S, Kong W, Hao J, Shang L (2009) Two-decade reconstruction of algal blooms in China's Lake Taihu. Environ Sci Technol 43(10):3522–3528
Halal F (2007) Intelligent hybrid techniques in mobile robot control. PhD thesis, Technical University of Bucharest, Romania
Halal F, Zaremba M (2010) Hybrid geno-fuzzy control for deliberative-reactive navigation. IFAC workshop on intelligent control systems, Sinaia, Romania
Marthaler D, Bertozzi AL (2003) Tracking environmental level sets with autonomous vehicles. In: Butenko S et al (eds) Recent developments in cooperative control and optimization. Kluwer, Norwell
Nieburand E, Koch C (1997) Computational architectures for attention. In: Parasuraman R (ed) The attentive brain. MIT Press, Cambridge
Ögren P (2003) Formations and obstacle avoidance in mobile robot control. PhD thesis, University of Stockholm, Sweden
Palenychka R, Zaremba M (2012) Method and multi-scale attention system for spatiotemporal change determination and object detection. US Patent Application No. 13/530,981
Topliss BJ, Platt T (1986) Passive fluorescence and photosynthesis in the ocean: implications for remote sensing. Deep Sea Res 33:849–864
Tyler AN, Svab E, Preston T, Présing M, Kovács WA (2006) Remote sensing of the water quality of shallow lakes: a mixture modelling approach to quantifying phytoplankton in water characterized by high-suspended sediment. Int J Remote Sens 27(8):1521–1537
Vincent RK, Qin X, McKay RML, Miner J, Czajkowski K, Savino J, Bridgeman T (2004) Phycocyanin detection from LANDSAT TM data for mapping cyanobacterial blooms in Lake Erie. Remote Sens Environ 89(3):381–392
Zhang F, Leonard NE (2005) Generating contour plots using multiple sensor platforms. In: Proceedings of the IEEE Swarm intelligence symposium. doi:10.1109/SIS.2005.1501637

Multi-Strategy Navigation for a Mobile Data Acquisition Platform Using Genetic Algorithms

Submitted: 5th December 2016; accepted: 25th January 2017

Fadi Halal, Marek B. Zaremba

DOI: 10.14313/JAMRIS_1-2017/4

Abstract:
Monitoring of biological and chemical pollutants in large bodies of water requires the acquisition of a large number of in-situ measurements by a mobile sensor platform. Critical to this problem is an efficient path planning method, easily adaptable to different control strategies that ensure the collection of data of the greatest value. This paper proposes a deliberative path planning algorithm, which features the use of waypoints for a ship navigation trajectory that are generated by Genetic Algorithm (GA) based procedures. The global search abilities of Genetic Algorithms are combined with the heuristic local search in order to implement a navigation behaviour suitable to the required data collection strategy. The adaptive search system operates on multi-layer maps generated from remote sensing data, and provides the capacity for dealing with multiple classes of water pollutants. A suitable objective function was proposed to handle different sampling strategies for the collection of samples from multiple water pollutant classes. A region-of-interest (ROI) component was introduced to deal effectively with the large scale of search environments by pushing the search towards ROI zones. This resulted in the reduction of the search time and the computing cost, as well as good convergence to an optimal solution. The global path planning performance was further improved by multipoint crossover operators running in each GA generation. The system was developed and tested for inland water monitoring and trajectory planning of a mobile sample acquisition platform using commercially available satellite data.

Keywords: *genetic algorithms, path planning, monitoring system, remote sensing, navigation control, heuristic search*

1. Introduction

Acquisition of a large number of in-situ measurements by a mobile platform is a basic task in the process of monitoring biological and chemical pollutants in large bodies of water. Monitoring of environmental phenomena in inland waters requires measuring a variety of physical processes, such as nutrient concentration, wind effects, and solar radiation [26]. Remote sensing (RS) techniques provide significant advantages in terms of spatial and temporal coverage and cost-efficiency. The maps of large environment areas are often obtained through the processing of satellite imagery. The multi-spectral data can subsequently be used to obtain models of water pollutants, such as the concentration of chlorophyll (Chl-a) or total suspended sediments (TSS) [17], by applying such measures as the maximum chlorophyll index (MCI) [10] or the ocean chlorophyll 4 algorithm (OC4v4) [21]. In many situations the remote sensing data have to be augmented and updated by *in situ* measurements. This is due to the need for precise local measurements, for the calibration of satellite imagery in varying water conditions, and for the purpose of precise local decision making.

Critical to this sample acquisition problem is an efficient path planning method, easily adaptable to different control strategies that ensure the collection of data of the greatest value. Acquisition of different types of samples may require appropriate behaviours that implement different collection strategies. Designing a multi behaviour search system for a mobile sample acquisition platform requires answering the following questions. Which is the suitable navigation mode for a specific water pollutant? How to compute the cost of the solution? How can the solution of the path planning problem deal with multiple patches of high concentration of the pollutant?

In general, the path planning procedure designs a trajectory that visits a given set of points such that the optimisation process minimises the total travel distance. This task can be defined in terms of a combinatorial optimization problem with a globally optimal solution that satisfies all hard and soft constraints. The optimal solution or a set of globally optimal solutions minimises or maximises the objective function. The path finding problem is typically defined in terms of the Travelling Salesman Problem (TSP) [7] or a more general Vehicle Routing Problem (VRP) [4]. Determining the optimal solution is an NP-hard problem, so the size of problems that can be solved optimally is limited [3]. In the situation of environment monitoring systems, the problem is even more complex because exact positions of the sampling points are not known a priori. In practice, therefore, solutions to optimal path planning problems have to incorporate heuristic methods.

A variety of heuristic methods have been investigated. Evolutionary algorithms have been employed in many variants. In [6] an ant colony optimization system was presented to solve the problem of designing an optimal trajectory for a mobile data acquisition platform. Luo *et al.* [20]an intelligent mobile vehicle is required to reach multiple goals with a shortest path that, in this paper, is capable of being implemented in TSP (Traveling Salesman Problem proposed a hybrid GA and D* algorithm for real-time map building and navigation for multiple goals purpose. Yoshikawa and Terai [32] proposed a car navigation system using hybrid genetic algorithms and D algorithm. Their system finds a route which has several passing points before arriving at the final destination. In [18] the path planning problem for a submarine navigation application was solved using the artificial bee colony algorithm. The use of a cultural hybrid algorithm to solve the mission planning was reported in [33]. An improved simulated annealing artificial network to plan the path for a mobile robot was employed in [8].

Genetic algorithms have been frequently used in NP-hard problems due to their flexibility and high quality of the search results [25]. They can provide a solution without any advance knowledge about the environment, and are largely unconstrained by the limitations of the classical search methods [24]. By mimicking natural evolution processes, they have the ability to adaptively search large spaces in near-optimal ways. In practical terms, GA methods are easy to interface with simulation models. An important feature that should be considered in implementing GA techniques is that they are problem specific. Due to the constraints of a particular problem and the operation of crossover and mutation mechanisms, feasible offsprings often cannot be obtained by applying exclusively genetic algorithms. In order to ensure the feasibility, additional algorithms should be incorporated. For example, [34] developed an improved genetic algorithm, where an obstacle avoidance algorithm and the distinguish algorithm are combined with a GA algorithm to select only the feasible paths and to improve the path planning efficiency. The distinguish algorithm is designed for distinguishing whether the path is feasible or not.

In this paper we present a hybrid GA-based method developed to optimize path planning and navigation using pollutant maps generated from RS imagery. The power of the global GA search is combined with the speed of the local optimizer. Both optimizers work cooperatively to find the optimal solution, where GA determines the optimal region, and then the local optimizer takes over to find the best position for acquiring water samples [13]. In order to deal effectively with the large-scale environment, the following modifications to the state-of-the-art approaches were introduced. In the first place, this paper implements an improved combination of a GA with an obstacle avoidance algorithm and the distinguish algorithm proposed initially in [34]. This algorithm puts a feasible path in the feasible group and deletes an infeasible path or keeps it in the infeasible group, which markedly improves the efficiency of the path planning. The big family pool was adopted in our system, which consists of all old-generation solutions and current-generation offsprings obtained after mutation and crossover operations combined with different meta-heuristic solutions. Based on the Cooperative Genetic Optimization Algorithm [14], it offers a greater search selection diversity and gives the system the ability to save the elite searching experience from one population to the next one.

Multi-layered maps were employed to generate spatial and functional properties of the environment. Those maps enable the planning system to perceive and interpret environments according to different environment features. ROI maps can be extracted from the multi-layer map as additional layers. The ROI approach facilitates the planning system in directing the search toward desirable patches by paying additional attention to desired regions, and assuring at the same time the generation of feasible solutions [11]easily adaptable to different control strategies that ensure the collection of data of the greatest value. This paper proposes a hybrid Genetic Algorithm (GA.

In general, each optimization problem to be solved by a GA method requires a unique fitness function that represents a performance criterion used in the evaluation of the performance of all chromosomes in the population. Many functions, such as travelling distance, time window and the sample values (weights) should be optimized simultaneously. This may involve a combination of maximization and minimization criteria [5]. Individual objective functions are usually combined into a single composite function by weighting the objectives with a weight vector. The result of the optimization should reach a reasonable solution that compromises multiple objectives [23]. For mission planning of an unmanned aerial vehicle (UAV), [29] used the distance, the hazard, and the maneuvering of the route as components of their cost function. Each component has a weight factor assigned according to the objectives of the mission. The hazard is related to the existence of obstacles near the path, and the maneuvering refers to the maneuvers required to perform target tracking. For efficient determination and search of the best flight (UAV) routes, an objective function was created in [27] which involves the timeliness and the smoothness of the path. The objective function discussed in [9] included several components: the cost of the motion from the start node to the current node, the heuristically estimated value of getting from the current node to the goal, the terrain traversability component, the direction change cost, and the cost of navigating in shadow areas. Each component has a corresponding coefficient factor used to weight the objective function components according to its importance to the mission. An optimized path planning for skid-steered mobile robots [16] uses a cost function which consists of the terrine properties, longitudinal motion and turning of the robot. In this work, an objective function proposed to deal with the experiment conditions comprises the following components: the samples value, the ROI award, the distance, and the sampling time.

The waypoint technique was used in the path planning process as an approach appropriate for large monitoring environments [30]. Waypoints are defined as abstract points [15] used to determine local positions [28] through which a mobile platform can navigate, reach its region-of-interest destination, and collect the water pollutant samples [22]. In the application discussed in this paper, waypoints correspond to sampling points. In order to deal with multiple sampling areas, multi-point crossover (MPC) was implemented. The MPC operator works to build the final solution which consists of valuable segments of local paths from many search strategies. The mutation operator improves the local search and helps the population to avoid local minima. The evolution process optimizes the path planning by designing new chromosomes which consist of best value samples from many global paths.

Experiments were conducted on data from Lake Winnipeg located in Manitoba, Canada. The adaptive search techniques presented in the paper were applied to optimize the location of the sampling points for different pollution indices and behaviours: the concentration of individual pollutants and their combinations, and the maximum gradient of pollutant concentration.

The structure of the paper is as follows. Section 2 addresses the sample acquisition problem using remote sensing data. A discussion of the proposed hybrid GA-based architecture for path planning and the optimisation of the multi-behaviour sample acquisition is presented in Section 3. Experimental results are discussed in Section 4.

2. Multi-Strategy Sample Acquisition Mission

2.1. Problem Statement

The problem addressed in this paper consists in planning a trajectory for precise acquisition of water pollutants by a mobile platform, when the planning process is guided by prior rudimentary information about the distribution of pollutants obtained from remote sensing data. The acquisition mission should incorporate different acquisition strategies.

The sample acquisition mission is performed within a more general procedure consisting of the following phases:

1) Determination of water regions and their types, sample location zones, and water pollutants to be sampled;
2) Identification of the pollutant detection indices, coverage methods (e.g., uniform coverage, maximum concentration gradient) and the number of samples;
3) Selection of the sources of remote sensing data and their calibration methods;
4) Selection of the ancillary data from *in situ* sensors (e.g. wind, temperature);
5) Determination of the acquisition mission parameters (e.g., total mission time).

Most of the above factors and conditions affect the strategies that have to be incorporated in the planning procedure. Mission strategies can be classified in two categories:

(1) Water pollutant concentration strategies

In this type of strategies the aquatic acquisition platform collects the most valuable samples from different pollutant classes and their combinations, such as

- Chl-a,
- Chl-a & (TSS),
- Chl-a & Dissolved Organic Carbon (DOC),
- Chl-a & TSS & DOC.

In this class of strategies, specific samples should be collected while neglecting other samples within a certain time window. Time windows can be imposed because of the deterioration of the quality of samples over a period of time. Time requirements for Chl-a concentration sampling are discussed in [12].

With respect to the types of pollutants, the RS data have to be pre-classified. The final path maximizes the value of the collected samples along a trajectory that traverses regions of different distributions of the pollutant concentration. As a result, the planning algorithm works on many maps created to represent different concentration levels for different water pollutant classes. The optimal strategy directs the path to the best Region of Interest (ROI) zone. The samples values (weights) vary depending on the mission objective.

(2) Local coverage strategies:

In this mode the platform executes a specific navigation and collection behaviour depending on the shape of the sample spatial distribution. We distinguish here such sampling strategies as the uniform coverage of high-concentration areas, sampling at local concentration maxima, and sampling along maximum gradient lines, which is of interest in many environment monitoring applications [36]. The sampling process can be different in each patch to comply with the general and local mission goals.

Both types of strategies execute under some specific constraints. Time window constraints can be imposed on certain pollutant patches, and travel distance constraints on other patches. Also, a certain number of samples have to be collected in a specific patch before heading to another one.

2.2. GA-Based Planning System

Due to the complexity of the mission trajectory optimization problem, a hybrid GA/Adaptive Search system is proposed and investigated in this paper. The general architecture of the planning system is based on the deliberative architecture model [19]. As illustrated in Fig. 1, the deliberative level comprises the

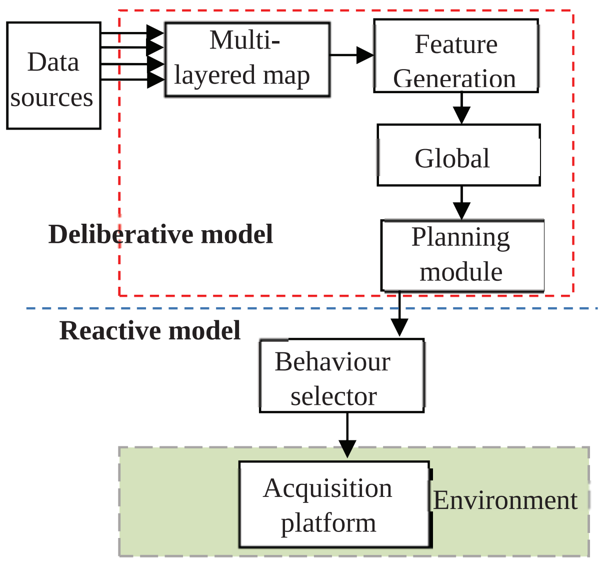


Fig. 1. General architecture of the GA- based planning system

environment modelling level, which operates on the remote sensing data and the ancillary information, and the adaptive GA-based trajectory generation level.

Water wave reflection can be exploited to determine the concentration of water pollutants. Examples of spectral signatures for different samples of chlorophyll pigment and TSS are shown in Fig. 2.

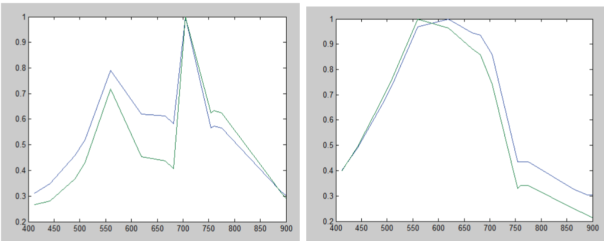


Fig. 2. Spectral signatures: a) Chl-a, b) TSS

The following two models were applied to measure the concentration of TSS [17] and Chl-a [10], [1] using different spectral bands of satellite images:

$$\mathrm{TSS} = 53.7\left[\frac{\mathrm{L}_{709}}{\mathrm{L}_{560}+\mathrm{L}_{665}}\right] - 17.0 \qquad (1)$$

where Lxxx is the radiance value of the band at wavelength xxx, and

$$\mathrm{MCI} = \mathrm{L709} - \mathrm{L681} - 0.389\ (\mathrm{L753} - \mathrm{L681}) \qquad (2)$$

The factor 0.389 is calculated as the wavelength ratio (709–681) / (753–681).

The input data structure used to generate the information required for multi-strategy path planning is implemented in the form of a multi-layer map (Fig. 3), which consists of a set of overlaying grid-based maps.

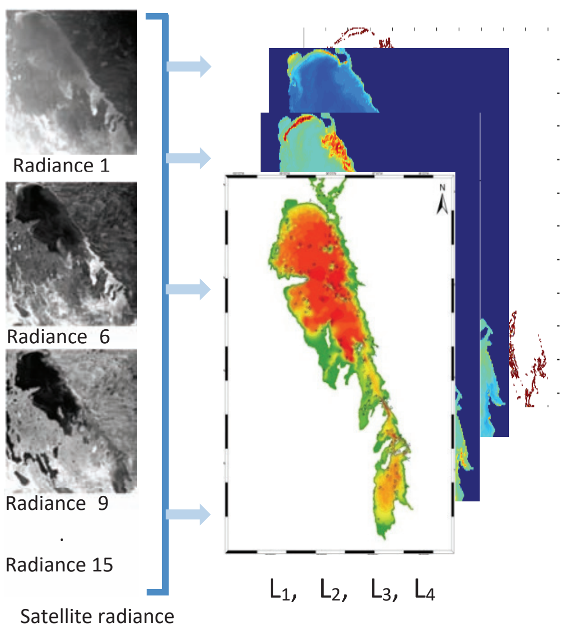


Fig. 3. Multi-layer map

The maps provide, for each spatial point (pixel), the numerical values NLi of the measured pollutants. The spatial resolution of the maps corresponds to the resolution of satellite images. Figure 3 shows the following layers: bathymetric map (L1), Chlorophyll-a (L2), TSS (L3), and the maximum gradient of chlorophyll-a (L4).

The overall goal of the acquisition mission is to maximize the quantity and the quality of the collected water pollutant samples V during the mission:

$$\max V = \max \sum_{j-1}^{M}\left[\sum_{i=1}^{N_j}\left(V_i^j(x,y)\right)\right] \qquad (3)$$

where V is the value of the sample, Nj is the number of the samples for each pollutant, and M is the number of water pollutant classes.

3. GA Method for Path Planning

3.1. Genetic Algorithm Architecture

The basic operation of the proposed GA-based path planning procedure can be summarized as follows (Fig. 4). The sampling points correspond to the

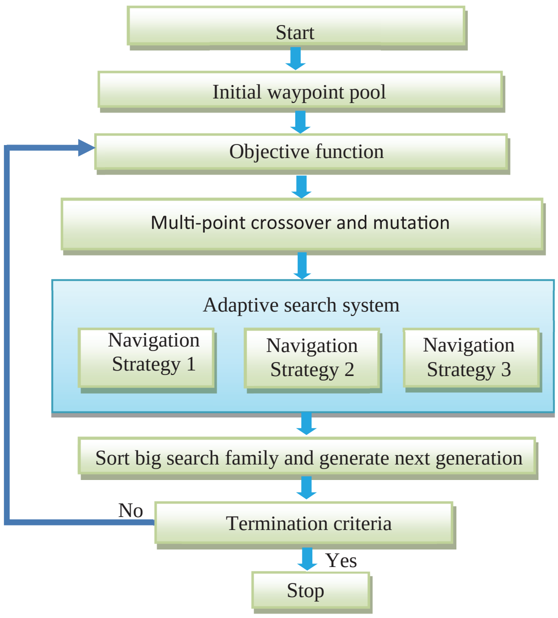


Fig. 4. Genetic Algorithm based path planning

waypoints of the global path of the mobile platform. Thus, the global path consists of several local paths, which are the arcs between two waypoints with a directed connection between them. The initial population of waypoints is pruned to generate collision free paths, subsequently stored in the initial chromosome pool population. Unfeasible solutions are deleted.

The adaptive search (AS) system improves the elite paths (the best 10 solutions) and returns efficient paths adapted to the local navigation behaviour. The big family pool consists of all old-generation solutions and current-generation offsprings obtained after the mutation and crossover operations combined with AS solutions. It gives the system the ability to save the elite search experience from one population to the next one [14]. The big family search results are sorted and pruned to form the next generation (Fig. 5).

A more detailed description of individual steps of the algorithm follows below.

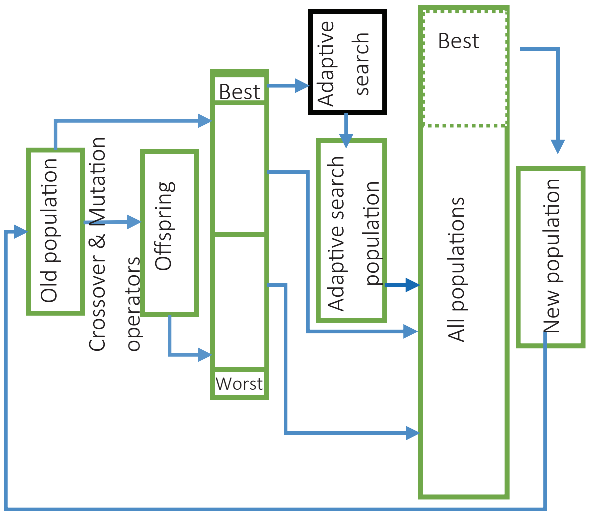


Fig. 5. Big family search [14]

3.2. Path Planning and Initial Waypoint Population

In the GA-based path planning procedure the population is represented, as in the Vehicle Routing Problem, by ordered sets of waypoints. Each feasible set is considered to be an individual in the population. Each waypoint, which is a sample candidate, represents a location in the environment (x,y). The initial genotype can be represented by a cell array, where each pair of cells represents the local path length and the heading angle towards the subsequent waypoint.

The path planning generator works as follows:

1) Determine the first waypoint in the path, i.e., the starting point, with the initial angle equals to zero.
2) While the path planning doesn't reach the desired target, generate a random number of L, the path length, between L_{min} and L_{max}, and a random heading angle β between β_{min} and β_{max} obtaining the next waypoints[31]. A maximum number of waypoints is given for each search strategy.
3) Different strategies are applied to water pollutant patches by adjusting L and β. Each path planning strategy handles different number of samples depending on the search path.

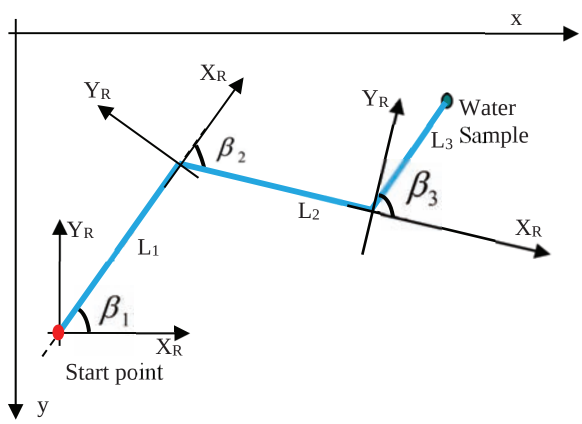


Fig. 6. Waypoint generation scheme

4) Continue with another patch or return to the starting point, depending on constraints, such as the maximum travel distance or the maximum number of water samples.

Figure 6 illustrates the path planning generator.

The chromosomes are encoded as an integer string. Each gene consists of two variables, the local path length and the heading angle as shown in Fig. 7a. Depending on the start point and the chromosome, the waypoint generation produces records as in Fig. 7b. The path planning waypoints are represented in the form of a long array as depicted in Fig. 7c. The GA search finds the waypoints between the starting point of the mission and the destination point.

a)

Heading angle	Travel distance	Heading angle	travel distance	..	Heading angle	travel distance

b)

heading	travel distance	previous waypoint (start point)	waypoint$_1$ x coordinate (Latitude)	Waypoint$_1$ y coordinate (Longitude)

c)

Starting point	Node 1	Node 2	.	……..	Node n	Destination point

Fig. 7. Chromosome and waypoint array. a) GA chromosome; b) Waypoint representation; c) Waypoint array

An obstacle free path planning algorithm [35] was adopted to deal with spatial constraints. It produces a feasible path that satisfies the conditions that the waypoints should be located outside the obstacles, in the sampling space, and the local path should not intersect with the obstacles.

In order to comply with the feasibility constraints and to enhance the efficiency of the path, a certain number of the waypoints in the elite solutions can be modified for each generation by applying three possible operations: waypoint deletion, insertion, or replacement [2] a tabu search system model is designed and a tabu search planner algorithm for solving the path planning problem is proposed. A comprehensive simulation study is conducted using the proposed model and algorithm, in terms of solution quality and execution time. A comparison between our results with those of A* and genetic algorithms (GA. Waypoint deletion eliminates all waypoints in the clear water body. The waypoint insertion operation explores the neighbourhood and inserts a new waypoint, according to a predefined behaviour for each water pollutant type. After deleting and inserting the waypoints the algorithm evaluates the path, conducts a neighbourhood search to replace the lowest waypoint value with a new one, and builds another feasible path Pn that satisfies the mission constraints.

3.3. Fitness Function

The fitness function is a particular type of the objective function that quantifies the optimality of a solution and evaluates the suitability of a solution with respect to the overall goal. In our navigation problem, it maximizes the collected information, directs the robot towards the ROI, and incorporates distance and time penalties.

The proposed fitness function F consists of 4 components, calculated for each candidate sample

$$F = SV + ROI + DIS + ST \tag{4}$$

where:

SV – data set value, which determines the value of acquired samples according to Eq. 5;

$$SV \left[\frac{\sum_{i=1}^{t} samples\ values}{Maximume\ number\ of\ collected\ samples}\right] \tag{5}$$

where sample values are calculated as the values of *V* in Eq. 3.

ROI – the region of interest award, introduced in order to optimize the convergence of the search for quality samples (Eq. 6):

$$ROI \left[\frac{\sum_{i=1}^{t} ROI\_samples}{Maximume\ number\ of\ collected\ samples}\right] \tag{6}$$

where: *DIS* – distance factor; *ST* – sampling time factor.

Two objective functions with different forms of *DIS* and *ST* factors were tested to assess their impact on the effectiveness of the sample acquisition mission:

Objective function 1 linearly maximizes the sample value and the ROI award and exponentially minimizes the sampling time and the mission travel distance. The distance and the time become, as the sample acquisition mission progresses, quadratically more expensive.

Objective function 2 linearly maximizes the sample value as well as the sampling time and the ROI award, and linearly minimises the mission travel distance.

Objective function 1
$DIS = 1 - \left[\frac{\sum_{i=1}^{t}\sqrt{(y_t - y_{t-1}) + (x_t - x_{t-1})}}{Maximume\ allowable\ distance}\right]$
$ST = 1 - \left[\frac{\sum_{i=1}^{t} sampling\ time}{Maximume\ allowable\ sampling\ time}\right]$
Objective function 2
$DIS = 1 - \left[\frac{\sum_{i=1}^{t}\sqrt{(y_t - y_{t-1}) + (x_t - x_{t-1})}}{Maximume\ allowable\ distance}\right]$
$ST = \left[\frac{\sum_{i=1}^{t} sampling\ time}{Maximume\ allowable\ sampling\ time}\right]$

Fig. 8. Linear and nonlinear DIS and ST components of the fitness function

3.4. Multi-Behaviour Operation

The basic idea of the multi-strategy GA-based path planning is that the acquisition platform explores water pollutant patches using different behavioural characteristics depending on the sampling requirements in each patch. The behaviours affect the local search optimization where the best evaluated neighbour is selected according to the adopted behaviour. The following behaviours represent different sampling strategies.

Behaviour 1– Short local path and high sample values. The sampling process selects the best sample according to equation

$$SV = \left[1 - \frac{local\ path_{i\ j}}{Max\ local\ path}\right] * V_{chlo\ j} \tag{7}$$

where *i* is the departure waypoint, *j* is the destination waypoint, and is the chlorophyll concentrations in cell (x,y) of the MCI layer.

Behaviour 2 – Maximum gradient (MG) sampling. Valuable samples (bigger than a given threshold number) are selected along a short local path according to the following equation:

$$SV = \left[1 - \frac{local\ path_{i\ j}}{Max\ local\ path}\right] * V_{MG} \tag{8}$$

The sampling behaviour for other samples maximizes the local path according to equation

$$SV = \left[\frac{local\ path_{i\ j}}{Max\ local\ path}\right] * V_{MG} \tag{9}$$

Behaviour 3 – Multiple pollutant patches.
The AS procedure selects the best sample value (Eq. 10), with the maximum local path range distance and the highest sample weight.

$$SV = \beta \left[\frac{local\ path_{i\ j}}{Max\ local\ path}\right] * V_{chlo\ j} * V_{TSS\ j} \tag{10}$$

where and are Chl-a and TSS concentrations in cell (*x,y*) taken from the MCI and TSS maps.

Behaviour 4: Long local path and TSS sampling
The AS procedure selects the best sample value as defined by equation (11), where the value sample corresponds to the maximum local path range distance and the highest sample weight;

$$SV = \left[\frac{local\ path_{i\ j}}{Max\ local\ path}\right] * V_{TSS\ j} \tag{11}$$

An example of water pollutant patches obtained for different behaviours from a 3-layer map (MCI, TSS and MG) is shown in Fig. 8.

3.5. Multi-point Crossover

Multi-point crossover is used to enhance the process of selecting valuable samples located in distant zones. The crossover procedure is explained in Fig. 10. Parent chromosomes, P1 and P2, are cut at multiple random locations, and the portions of the chromosomes between the cuts are swapped. The

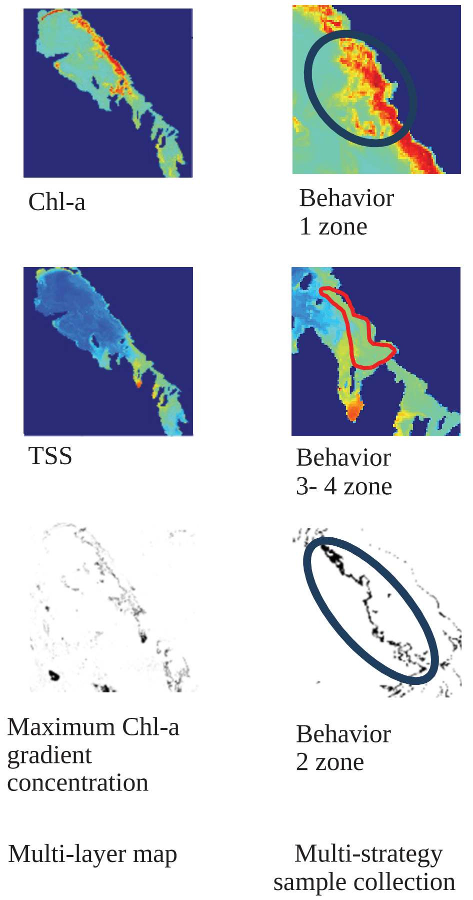


Fig. 9. Water pollutant zones for multi-behaviour navigation

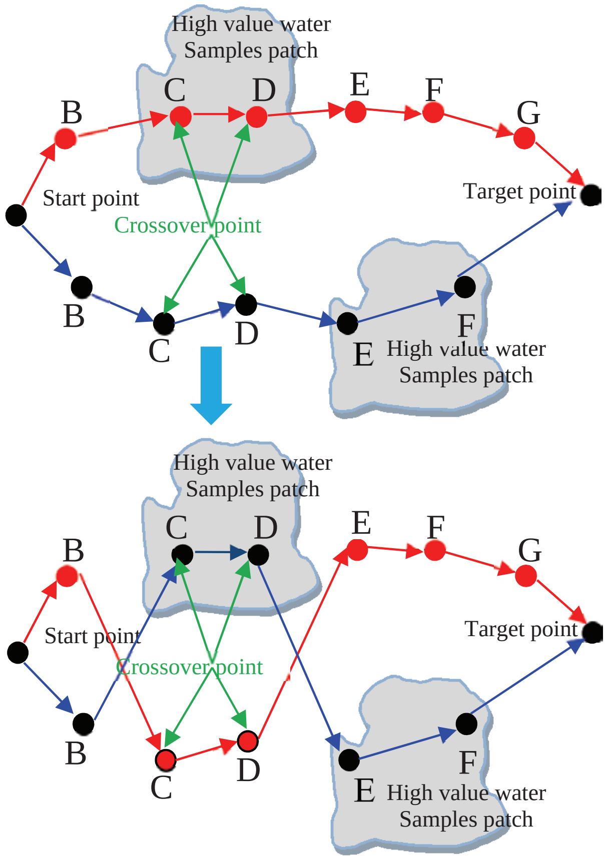


Fig. 10. Multi-point crossover. a) A two-chromosome and two-point crossover. b) Two offsprings

result is a pair of offsprings I1 and I2. The crossover is applied on the best-fitness chromosomes chosen from the pool. Due to the difference in the chromosome length, the crossover point should be applied to the shorter chromosome.

3.6. Planning Process

Figure 11 represents the overall architecture of the developed adaptive GA-based mission planning system. The mission objective is defined and accompanied with a strategy definition to achieve the mission goal. A multi-layer map is generated to interpret the global environment and to weight the importance of different water pollutants in the sampling strategy. A set of ROIs is generated to guide the search toward specific patches associated with their acquisition strategies.

An adaptive search algorithm improves the multi-strategy path planning in different patches employing local search optimising procedures. A suitable fitness function evaluates the chromosome in the search for maximizing the mission goal.

4. Experimental Results

4.1. Experimental Framework

The experiments were carried out using satellite data from the northern basin of Lake Winnipeg for a path starting at the point located at longitude (99°02'08") W and latitude (55°35'18") N and the destination point at longitude (96° 50' 24") W and latitude (51°55'51") N. The direct distance between the start point and the target is around 236 km. The maps used in the experiments were in the form of a raster grid, where the dimensions of cells corresponded to the resolution of the MERIS satellite sensor, i.e., 260 m × 300 m. Each cell had an associated value Vx,y obtained from the multi-layer map as discussed in Section 2.

ROI maps guide the multi-strategy sampling to orient the acquisition platform toward the valuable samples in the ROI zones using the penalty/award mechanism. Figures 12 a) b) and c) show regions of interest for MCI, TSS and the maximum gradient of the chlorophyll concentration. The regions are defined as the concentration of TSS bigger than 0.3 from the normalised TSS model, and the concentration of chlorophyll-a bigger than 0.5 from the MCI normalised model. Figure 12d represents the overall ROI formed from the MCI and TSS zones. Figure 12e illustrates three ROI zones, which are MCI, TSS and maximum gradient chlorophyll concentration, used in the experiments.

Matlab Genetic Algorithm Optimization Toolbox (GAOT) was used to program the proposed hybrid system. Table 1 shows the Genetic Algorithm parameters chosen for the optimization process.

Four experiments were conducted with two objective functions (Fig. 8) tested. Objective function 2 (linear optimization) was incorporated in the fitness function used in experiments 1 and 2, and objective function 1 (exponential optimization) in experiments 3 and 4. Hard distance and time constraints were implemented in the first two experiments. The mission

Table 1. Parameters of the Genetic Algorithm

Genetic Parameters	Magnitude
Number of generations	150
Population size	120
Crossover rate	60% randomly and the elite
Mutation rate	5% randomly and the elite
Type of crossover	Single-point and multi-point crossover
Type of mutation	4 point random & 4 maximum points
Selection type	Roulette Wheel

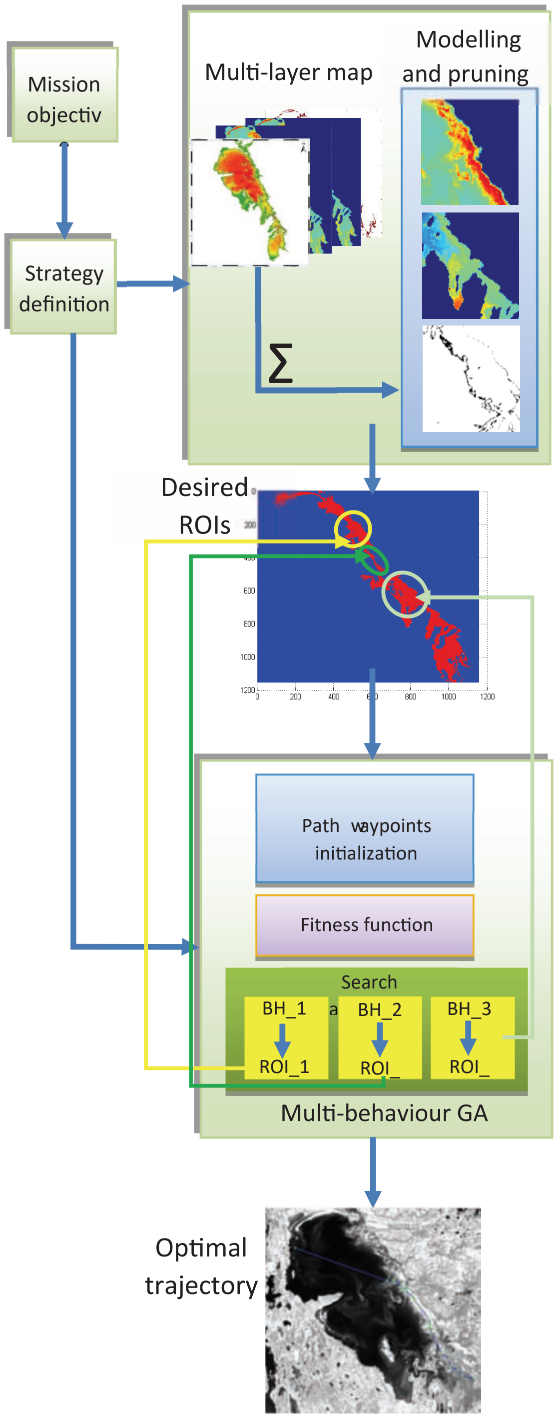


Fig. 11. Adaptive GA-Based Navigation System

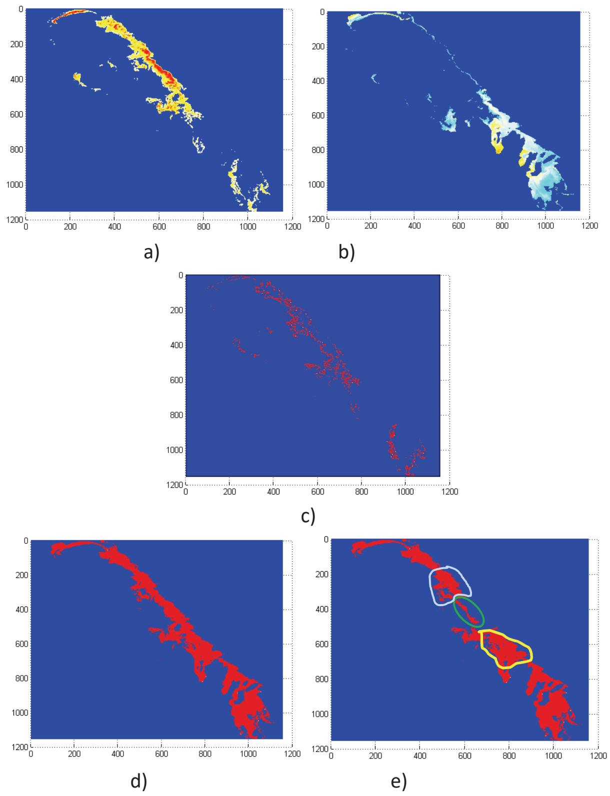


Fig. 12. a) Chl-a ROI (MCI > 0.5); b) TSS ROI (TSS > 0.3); c) Chl-a Max Gradient ROI; d) Combined Chl-a & TSS regions of interest, and e) Combined Chl-a & TSS & MG regions of interest

time was bounded by the value of 12 hours, and the travel distance was limited to 400 km. In experiments 3 and 4, the mission time had to be less than 9 hours, and the travel distance was limited to 330 km.

4.2. Path Planning Experiments

In the first experiment, the sample value (SV) was the sum of the TSS and Chl-a sample values. The results show that the path includes 10 samples from the clear water zone (outside the ROI zone), as shown in Fig. 13. The obtained results provide the rationale for hybridising the GA-based search for optimal samples.

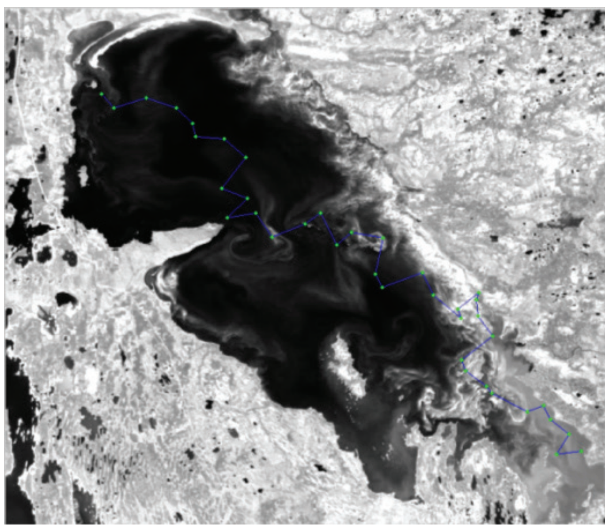


Fig. 13. Sample acquisition paths: Experiment 1

Experiment 2.

A simple adaptive search, consisting in limiting the search to ROIs, was introduced in the second experiment. However, no specific behaviour guided the waypoint generation. Figure 14 presents the path generated by the modified system. The sampling area is located entirely in the ROI. Table 2 compares the performance of the two experiments.

Table 2. Results of experiments 1 and 2

	Experiment 1 (GA)	Experiment 2 (ROI-optimized GA)
Sampling time	0.475 @ 38 samples	0.475 @ 38 samples
path length (m)	3.9989e+005	3.4364e+005
Samples value	0.7004	0.8304
ROI award	0.3675	0.5550

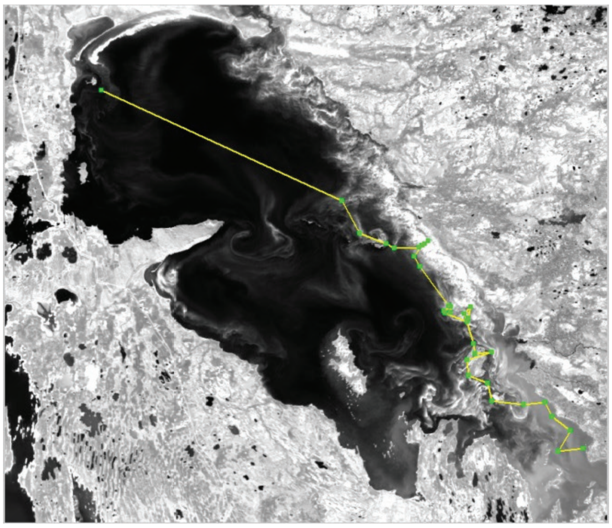


Fig. 14. Sample acquisition paths: Experiment 2

The path in the second experiment was approximately 56 km shorter and the value of the samples increased by about 13 percent, while keeping the number of samples at the same level.

4.3. Multi-Behaviour Navigation

In order to assess the multi-behaviour performance of the system and to further improve the path quality – in the context of the GA methodology – different behaviours were introduced to the local adaptive search the next two experiments. The third experiment explores the local behaviour optimization which performs two collection strategies depending on the types of the samples. Therefore, the ROI set consists of two zones, Chl-a and TSS. The search minimises the local path in the MCI patch according to Eq. 7, and maximises the local path in the TSS patch according to Eq. 11. The neighbourhood of a solution is explored, and the best neighbor is selected according to the adopted behaviour in each patch. Objective function 1 was used to optimise this experiment. The multi behaviour navigation shows good sampling performance in the two different patches, as shown in Fig. 15.

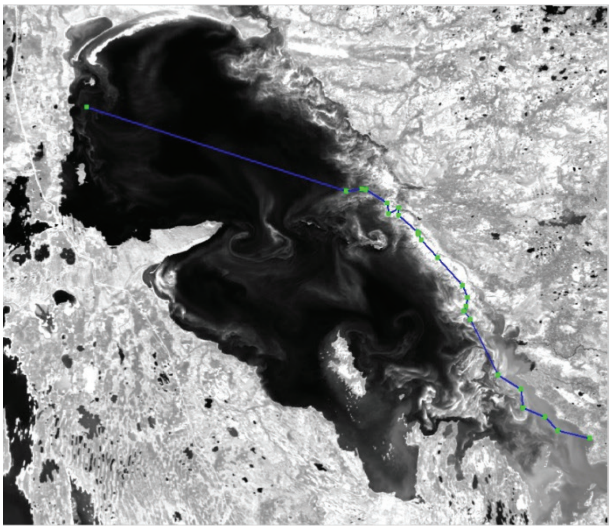


Fig. 15. Sample acquisition path from experiment 3

The mission collects 22 pure chl-a samples and 6 TSS samples along a 282 km long path. The samples value is 0.645, and ROI award equals to 0.6125. The distances between the chlorophyll samples are shorter than between the TSS samples, which is a consequence of applying the behaviour equation (Eq. 7) and high award for the Chl-a ROI. The longer local path between the six TSS samples results from the behaviour equation (Eq. 11). The total mission time is 8 hours and 54 minutes. The travel time is 7 hours and 14 minutes.

In the fourth experiment, the zone of the maximum gradient of chlorophyll concentration was introduced, which produced three separate patches with three different local search behaviours. Due to the behaviour conflict between the maximum gradient and the maximum value of the chlorophyll concentration, a new ROI zone was created. Thus, the three separate ROIs were generated as follows: the Chl-a zone, the maximum gradient of chlorophyll concentration, and the chlorophyll and TSS concentration zone. Figure 16 depicts the ROI map which was used in this experiment. The Chl-a samples were treated as the highest value samples with the shortest local path in the search algorithm (Eq. 7). In the maximum gradient zone, the search made the acquisition platform navi-

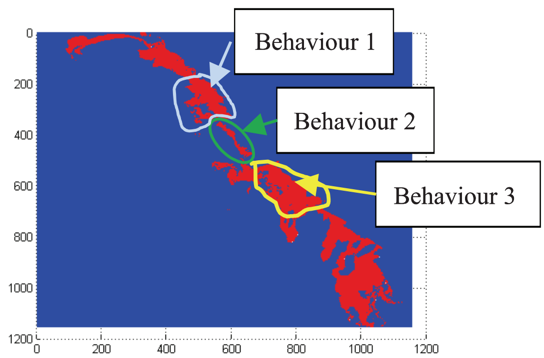


Fig. 16. Multi behaviour sampling for different patches

gate in adaptive way to follow the maximum gradient curve, using Eq. 8 and Eq. 9, and to maintain a proper distance between the samples. In the chlorophyll and

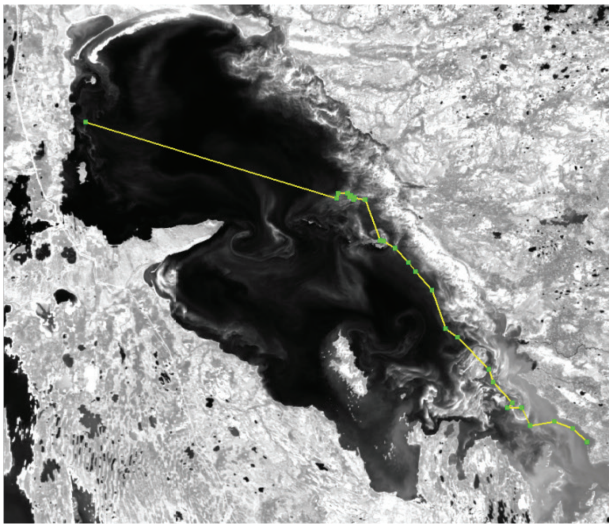


Fig. 17. Sample acquisition path from experiment 4

TSS zone, the behaviour model as in Eq. 10 was adopted. All behaviour optimization algorithms explored the neighbourhood and selected new waypoints in order to enhance the quality of the solution. Figure 17 shows an example of the planned path.

The path planning algorithm produced 28 samples as follows: 9 samples from the TSS & Chl-a zone; 5 samples from the MG zone; 14 samples from Chl-a zone including the start waypoint. The samples were collected along a path 285 km long. The normalized sample value was 0.5040 with the ROI award equal to 0.5650.

4.4. Convergence Analysis

To improve the convergence of the GA-based search, two crossover and two mutation operations were employed. The solutions to these operators were divided into two categories as follows: the first one consists of the elite solutions, and randomly selected solutions represent the second category.

The simulation results show that:

(1) The new procedure effectively enhanced the global search ability and improved the local searching ability;

(2) High convergence rate was obtained.

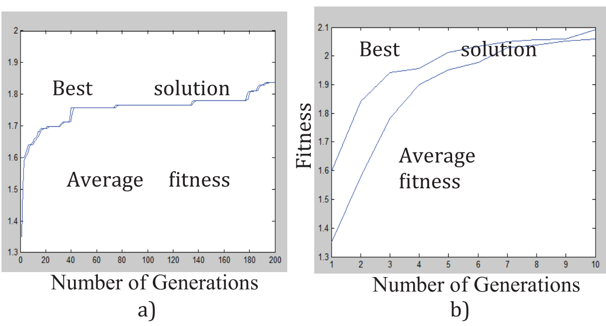


Fig. 18. Convergence in experiment 1 & 2

The results without the enhancement are shown in Fig. 18a. Both the quality of the solution and the speed of the optimization are enhanced by an order of magnitude by applying the improved operations (Fig. 18b).

The repeatability of the results is depicted, for experiments 3 and 4, in Figures 19a and 19b respectively. The convergence of both the best solution and the average solution is high.

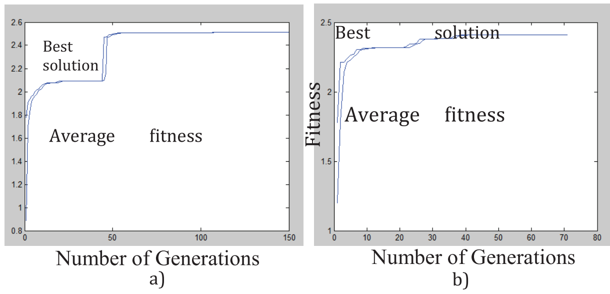


Fig. 19. Convergence in experiment 3 & 4

5. Conclusions

In this paper, hybrid genetic algorithms were proposed for navigation in a partly known environment, where the objective of the planning task is to find the optimal path for a mobile sample acquisition platform. The total quantity and quality of water samples is to be maximized according to navigation goals specified for each acquisition zone. Sampling in each patch may be guided by different patterns of behaviour for different purposes. Thus, the acquisition system is able to execute different behaviours along the global path. A hybrid genetic search was developed to deal with such a complex environment. The adaptive search algorithm models behaviours in different surrounding areas and executes them in each generation at the level of local path navigation. The locality of the navigation was defined in terms of regions of interest (ROI). In the process of generating the waypoints, the adaptive search deletes and inserts new waypoints in each solution depending on the ROI behaviour. This enhances the flexibility and the efficiency of path planning. The ROI component was introduces also in the fitness function, greatly speeding up the convergence of the planning process. Tests were conducted using medium-resolution satellite imagery. Multi-layered maps provided a rich context to the adaptive search system to perform flexible local search behaviours.

The experiments performed on large area environment show that the adaptive GA-based path planning method offers robust search capabilities and supports different sample acquisition strategies, ensuring the collection of meaningful data over multiple areas of interest.

ACKNOWLEDGEMENTS

The authors acknowledge funding from the International Science and Technology Partnership (ISTP – Canada) and the Natural Sciences and Engineering Research Council of Canada, grant 9227.

AUTHORS

Fadi Halal*, Marek B. Zaremba – Département d'informatique et d'ingénierie Université du Québec en Outaouais Gatineau, QC, J8X 3X7, Canada.
E-mail: half02@uqo.ca; zaremba@uqo.ca

*Corresponding author

REFERENCES

[1] K. Alikas, K. Kangro, A. Reinart, "Detecting cyanobacterial blooms in large North European lakes using the maximum chlorophyll index", *Oceanologia*, vol. 52, no. 2, 2010, 237–257. DOI: 10.5697/oc.52-2.237.

[2] I. Châari, A. Koubâa, H. Bennaceur, et al., "On the Adequacy of Tabu Search for Global Robot Path Planning Problem in Grid Environments". In: *5th Int. Conf. Ambient Syst. Networks Technol.* ((ANT-2014), Hasselt, Belgium, 2014, *Procedia Computer Science,* vol. 32, 2014, 604–613. DOI: 10.1016/j.procs.2014.05.466.

[3] I. Châari, A. Koubâa, H. Bennaceur, S. Trigui, K. Al-Shalfan, "SmartPATH: A hybrid ACO-GA algorithm for robot path planning". In: *2012 IEEE Congr. Evol. Comput.* (CEC 2012), Brisbane, Australia, 2012, 1–8. DOI: 10.1109/CEC.2012.6256142.

[4] C.T. Cheng, K. Fallahi, H. Leung, C.K. Tse, "A genetic algorithm-inspired UUV path planner based on dynamic programming", *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev*., vol. 42, no. 6, 2012, 1128–1134. DOI: 10.1109/TSMCC.2011.2180526.

[5] C.A. Coello Coello, "Evolutionary Multi-Objective Optimization: A Historical View of the Field", *IEEE Comput. Intell. Mag*., vol. 1, no. 1, 2006, 28–36. DOI: 10.1109/MCI.2006.1597059.

[6] G. Colmenares, F. Halal, M.B. Zaremba, "Ant Colony Optimization for Data Acquisition Mission Planning", *Manag. Prod. Eng. Rev.*, vol. 5, no. 2, 2014, 3–11. DOI: 10.2478/mper-2014-0011.

[7] H. Ergezer, K. Leblebicio, "3D Path Planning for-Multiple UAVs for Maximum Information Collection", *J. Intell. Robot. Syst.*, vol. 73, 2014, 737–762. DOI: 0.1007/s10846-013-9895-6.

[8] M. Gao, J. Tian, "Path Planning for Mobile Robot Based on Improved Simulated Annealing". In: *3rd Int. Conf. Nat. Comput.*, 2007, vol. 3, 8–12. DOI: 10.1109/ICNC.2007.547.

[9] A. García, A. Barrientos, "3D Path planning using a fuzzy logic navigational map for Planetary Surface Rovers". In: *ASTRA 2011 11th Symposium on Advanced Space Technologies in Robotics and Automation*, 12th–14th April 2011

[10] J. Gower, S. King, P. Goncalves, "Global monitoring of plankton blooms using MERIS MCI", *Int. J. Remote Sens*., vol. 29, no. 21, 2008, 6209–6216. DOI: 10.1080/01431160802178110.

[11] F. Halal, M.B. Zaremba, "Multi-strategy spatial data acquisition missions using genetic algorithms", IFAC-PapersOnLine, vol. 48, no. 3, 2015, 778–783. DOI: 10.1016/j.ifacol.2015.06.177.

[12] J. A. Hambrook Berkman, M.G. Canova, "Algal Biomass Indicators", *U.S. Geol. Surv. TWRI* book 9: Biol. Indic., 2007, chapter A.7, section 7.4.

[13] R.L. Haupt, S.E. Haupt, *Practical Genetic Algorithms*, John Wiley & Sons, Inc., Hoboken, New Jersey., 2004.

[14] C.-C. Hsu, Y.-C. Liu, "Path planning for robot navigation based on Cooperative Genetic Optimization". In: *Proc. 11th IEEE Int. Conf. Networking*, Sens. Control, 2014, 316–321. DOI: 10.1109/ICNSC.2014.6819645.

[15] M.T.S. Ibrahim, S.V. Ragavan, S.G. Ponnambalam, "Way point based deliberative path planner for navigation", *IEEE/ASME Int. Conf. Adv. Intell. Mechatronics* (AIM), 2009, 881–886. DOI: 10.1109/AIM.2009.5229900.

[16] P. Jaroszek, "Any-angle Global Path Planning for Skid-Steered Mobile Robots on Heterogeneous Terrain", *Journal of Automation, Mobile Robotics, and Intelligent Systems*, vol. 10, no. 2, 2016, 50–55. DOI: 10.14313/JAMRIS_2-2016/15.

[17] S. Koponen, J. Vepsäläinen, J. Pulliainen, et al., "Using meris data for the retrieval of CHL-A, CDOM and TSS values in the gulf of Finland and lake Lohjanjärvi", Eur. Sp. Agency, (Special Publ. ESA SP. 2004, 2005, 407–413.

[18] B. Li, R. Chiong, L. Gong, "Search-evasion path planning for submarines using the Artificial Bee Colony algorithm". In: *2014 IEEE Congr. Evol. Comput.*, July 2014, 528–535. DOI: 10.1109/CEC.2014.6900224.

[19] N.K. Lincoln, S.M. Veres, L.A. Dennis, M. Fisher, A. Lisitsa, "Autonomous asteroid exploration by rational agents", *IEEE Comput. Intell. Mag*., vol. 8, no. 4, 2013, 25–38. DOI: 10.1109/MCI.2013.2279559.

[20] C. Luo, S.X. Yang, M. Krishnan, M. Paulik, Y. Chen, "A hybrid system for multi-goal navigation and map building of an autonomous vehicle in unknown environments". In: *2013 IEEE Int. Conf. Robot. Biomimetics*, ROBIO 2013, 1228–1233.

[21] J.E. O'Reilly, S. Maritorena, B.G. Mitchell, et al., "Ocean color chlorophyll algorithms for SeaWiFS", *J. Geophys. Res*., 103, 1998, 24937–24953.

[22] S. Park, J. Cho, "ROI-based visualization of spatial information for a remote-controlled robot". In: *2013 10th Int. Conf. Ubiquitous Robot. Ambient Intell.*, 2013, 234–235.

[23] G.P. Rajappa, *Solving Combinatorial Optimization Problems Using Genetic Algorithms and Ant Colony Optimization*, University of Tennessee, 2012.

[24] F. Rothlauf, R*epresentations for Genetic and Evolutionary Algorithms*, Springer-Verlag Berlin, Heidelberg, 2006.

[25] M. Samadi, M.F. Othman, "Global Path Planning for Autonomous Mobile Robot Using Genetic Algorithm". In: *2013 Int. Conf. Signal-Image Technol. Internet-Based Syst.*, 726–730. DOI: 10.1109/SITIS.2013.118.

[26] A. Singh, A. Krause, C. Guestrin, W.J. Kaiser, "Efficient informative sensing using multiple robots", *J. Artif. Intell. Res.*, vol. 34, 2009, 707–755. DOI: 10.1613/jair.2674.
[27] T.Y. Sun, C.L. Huo, S.J. Tsai, Y.H. Yu, C.C. Liu, "Intelligent flight task algorithm for unmanned aerial vehicle", *Expert Syst. Appl.*, vol. 38, no. 8,2011, 10036–10048. DOI: 10.1016/j.eswa.2011.02.013.
[28] Tomasz Gawron, Maciej M. Michalek, "Planning the Waypoint-Following Task for A Unicycle-like robot in Cluttered Environments, *Journal of Automation, Mobile Robotics, and Intelligent Systems*, vol. 9, no. 1, 2015, 77–90. DOI: 10.14313/JAMRIS_1-2015/10.
[29] G. Vachtsevanos, L. Tang, G. Drozeski, L. Gutierrez, "From mission planning to flight control of unmanned aerial vehicles: Strategies and implementation tools", *Annu. Rev. Control.*, vol. 29, no. 1, 2005, 101–115. DOI: 10.1016/j.arcontrol.2004.11.002.
[30] S. Veera Ragavan, S.G. Ponnambalam, C. Sumero, "Waypoint-based path planner for mobile robot navigation using PSO and GA-AIS". In: *2011 IEEE Recent Adv. Intell. Comput. Syst.* (RAICS 2011), 756–760. DOI: 10.1109/RAICS.2011.6069411.
[31] J. Xiao-Ting, X. Hai-Bin, Z. Li, J. Sheng-De, "Flight Path Planning Based on an Improved Genetic Algorithm". In: *2013 Third Int. Conf. Intell. Syst. Des. Eng. Appl.*, 775–778.
[32] M. Yoshikawa, H. Terai, "Car Navigation System Based on Hybrid Genetic Algorithm". In: *World Congr. Comput. Sci. Inf. Eng.*, vol. 5, 2009, 60–64.
[33] L. Yu, Z. Cai, "Robot exploration mission planning based on heterogeneous interactive cultural hybrid algorithm". In: *5th Int. Conf. Nat. Comput. ICNC 2009*, 583–587.
[34] S.C. Yun, V. Ganapathy, L.O. Chong, "Improved Genetic Algorithms based Optimum Path Planning for Mobile Robot". In: *11th Int. Conf. Control. Autom. Robot. Vis.*, Singapore, 2010, 1565–1570. DOI: 10.1109/ICNC.2009.15.
[35] C. Zeng, Q. Zhang, X. Wei, "Robotic global path-planning based modified genetic algorithm and A* algorithm". In: *3rd Int. Conf. Meas. Technol. Mechatronics Autom. ICMTMA 2011*, 167–170. DOI: 10.1109/ICMTMA.2011.613.
[36] F. Zhang, N.E. Leonard, "Generating contour plots using multiple sensor platforms". In: *Proc. IEEE Swarm Intelligence Symposium*, 2005, 309–316.

U.P.B. Sci. Bull., Series C, Vol. 69, No. 3, 2007 ISSN 1454-234x

EVOLUTIONARY LEARNING OF A FUZZY CONTROLLER FOR A MOBILE ROBOT

FADI HALAL[1], I. DUMITRACHE[2]

Tehnicile inteligente bazate pe logica fuzzy, rețelele neurale şi algoritmi genetici sunt folosite cu mult succes în conducerea roboților autonomi. Sistemele hibride bazate pe combinații ale acestor tehnici pot maximiza eficiența acestor tehnici. În această lucrare, se prezintă un sistem hibrid geno-fuzzy care foloseşte un algoritm genetic pentru optimizarea unui sistem de conducere cu logică fuzzy pentru un robot Khepera care trebuie să atingă un anumit punct în spațiul de lucru pe traseul cel mai scurt. Algoritmul genetic optimizează funcțiile de apartenență şi generează reguli optimale. Rezultatele prezentate în această lucrare demonstrează validitatea abordării hibride bazată pe combinații ale tehnicilor inteligente de conducere.

Fuzzy control systems, neural networks and genetic algorithms can be cooperatively used for designing robot control systems. This paper presents a hybrid geno-fuzzy system based on a genetic algorithm that optimizes the membership functions and the rule structure of a fuzzy controller. The robot is a Khepera mobile robot that has to follow a track and find a target. The presented results demonstrate the validity of such a hybrid approach.

Keywords: geno-fuzzy system, fuzzy logic, genetic algorithm, mobile robot.

1. Introduction

Intelligent robots sharing city roads with humans and other vehicles is not a simple dream. Autonomous driving will enable a robot vehicle to drive independently along the road. [1] In this paper we used fuzzy control to drive a Khepera robot on a given in a simulation environment named Kiks. The first results have given us the reason to apply a hybrid geno-fuzzy control system, which has successfully driven the robot along the track. The geno-fuzzy system improves the design process and the performance of the fuzzy control system. [2] The first section presents the robot control architecture. The second section shows the fuzzy control system while the third section explains how to apply geno-fuzzy

[1] PhD student, Dept. of Automatic Control and System Engineering , Faculty of Automatic and Computer Science, University POLITEHNICA of Bucharest, Romania, fadi_halal@yahoo.com
[2] Prof, Dept. of Automatic Control and System Engineering , Faculty of Automatic and Computer Science, University POLITEHNICA of Bucharest, Romania

system. We developed the experiment and compared the results in the fourth section.

2. Robot control architecture

The control of a robot under parameter variations and load disturbances is an important problem [3]. Fig. 1 is illustrating an approach to build a control algorithm for a mobile robot, which is the “sense- plan-act” architecture. The robot problem was decomposed into a vertical series slice, which has the following functionalities:

- Observe the surrounding environment
- Make an internal plan of the area
- Adapt the robot plan
- Execute the plan
- Create a new plan when some thing was changed

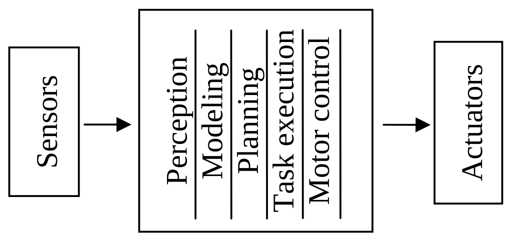


Fig.1. Traditional decomposition of a mobile robot control system into function modules.

In 1986 Rodney Brooks came with a new approach, which decompensate the problem into behaviors instead of function components [4], and this is illustrated into horizontal series slice, as shown in Fig. 2. Behaviors could be obstacle avoidance, wall-following, exploration or target seeking. A certain number of behaviors run as parallel processes, while each behavior can access all sensors, only one behavior can have control over the robot actuators.

In competitive control methods only one behavior affects the motor output of the robot in a particular moment. In cooperative control methods different behaviors may contribute to a single motor action although with different strength [5].

We decided to use a fuzzy control system to handle behavior selection, for controlling a Khepera mobile robot. This control structure has 3 sensor inputs which are S_{left}, S_{front} and S_{right}, corresponding to the sensors on the left, front and right hand side of the robot. This control will generate two outputs that are left motor speed and right motor speeds, respectively named LMS and RMS; these variables select the currently active behavior and cause a robot action. The control system it self developed with genetic algorithm designed to optimize a fitness function describing the task criteria.

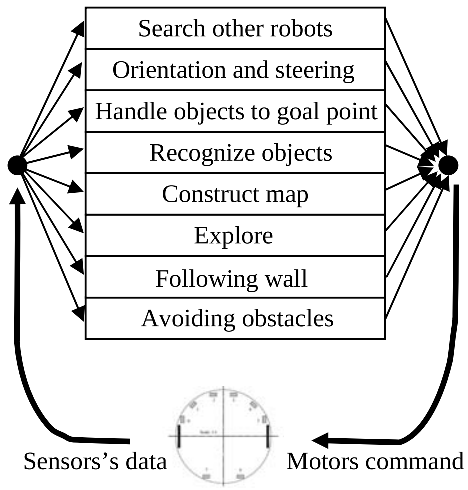


Fig.2. A decomposition of a mobile robot control system based on task achieving behaviors.

3. Fuzzy control system

3.1. Fuzzy controller design

The control input variables are the six sensors input (S0 …S5), and robot's coordinate. We ignored the two back sensors input (S6 and S7) that have no effect on the fuzzy control. The output variables are the left motor speed and right motor speed (LMS and RMS).

Sensors simplification was used as follows to reduce the number of the sensor inputs:

$S_{left} = ((S0 + S1)/2)$

$S_{front} = ((S2 + S3)/2)$

$S_{right} = ((S4 + S5)/2)$

as shown in Fig. 3 [5]. Each input has three trapezoidal linguistics membership functions, which are called *near*, *med*, and *far*, as shown in Fig. 4.A, denoting the distance from an obstacle. These inputs have the same membership function shape and design for each sensor input.

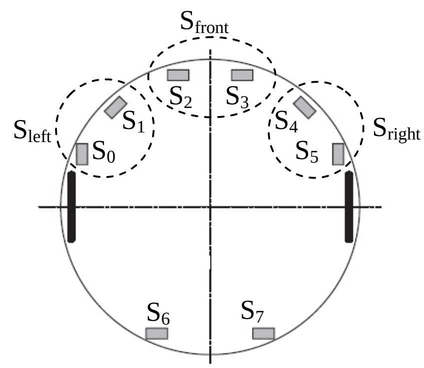
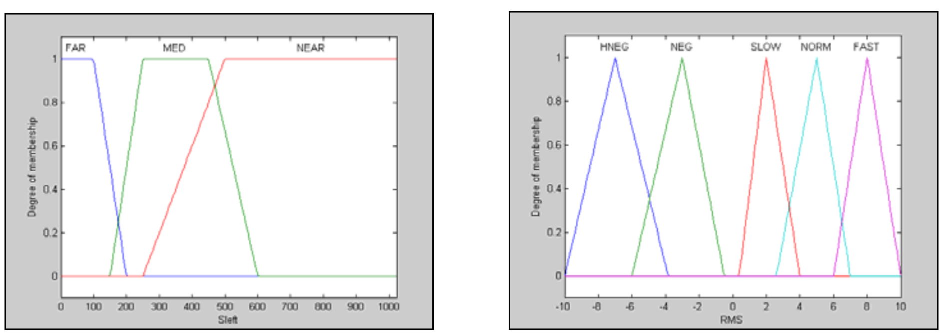


Fig.3. Simulation model of the Khepera robot

The output variables are LMS and RMS, respectively. Each output has five triangular membership functions, that are named as (*hneg*, *neg*, *slow*, *norm*, *fast*) representing the motor speed, as shown in Fig. 4.B. The fuzzy control system has 18 rules representing the robot behaviors, like left wall following, right wall following, walk through the corridor, obstacle avoidance, steering and tracking behavior.



(A) (B)

Fig.4. A: *Sleft* input membership function, B: *RMS* output membership function

3.2. Genetic representations of the fuzzy controller

We have encoded the rule base into a chromosome for a genetic algorithm in order to optimize these rules. We encoded the membership functions in each sensor input (Far, Med and Near), respectively coded as 1, 2 and 3, and for each output membership function (Hneg, Neg, Slow, Norm, and Fast), respectively coded as 1, 2, 3, 4 and 5. Fig. 5 illustrates the encoded input membership function, and the encoded output membership function.

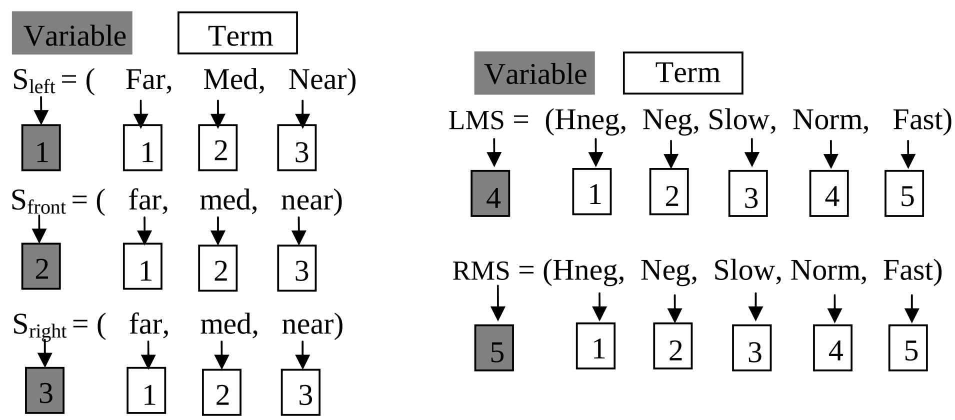


Fig.5. The input and the output membership function encoded

For example, the rule *If S_{left} = near and S_{front} = far and S_{right} = med Then LMS = fast and RMS = norm*, can be encoded as a string vector 3 1 2 5 4 and the chromosome is illustrated as a string vector, as shown in Fig. 6.

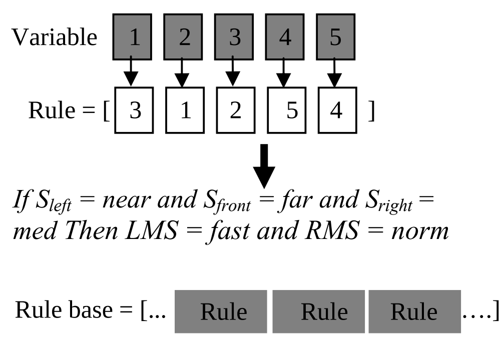


Fig.6. The rules base chromosome encoded

3.3. Rule base

The rule set had to be simplified. This simplification was accomplished by eliminating rules with low or even no probability to occur, and rules that cause the same effect in the robot movement. The final rules base is presented in table 2 which has been divided into eight basic groups: straight movement, when the robot has either no obstacle in the target direction or the obstacle is far; walk through the corridor, where the robot walks along the corridor; left wall following, where the robot followed the left wall; right wall following, where the robot followed the right wall, while avoiding obstacles behaviors were divided into four groups: Avoiding left front obstacles, Avoiding right front obstacles, Avoiding front obstacle, Avoiding blocked zone.

Table 1

Fuzzy rule base encoded

S_{left}	S_{front}	S_{right}	LMS	RMS	Robot's behaviors
1	1	1	5	5	straight movement
2	1	2	4	4	walk along the corridor
3	1	3	3	3	
2	1	3	3	4	
3	1	2	4	3	
2	1	1	4	2	left wall following
3	1	1	5	1	
1	1	2	2	4	right wall following
1	1	3	1	5	
2	2	1	4	2	avoiding left front obstacles
3	3	1	5	1	
3	2	1	5	2	
1	2	2	2	4	avoiding right front obstacles
1	3	3	1	5	
1	2	3	2	5	
1	2	1	1	5	avoiding front obstacle
2	2	2	2	2	avoiding blocked zone
3	3	3	2	2	

4. Experiments

In this paper we used a genetic algorithm to optimize the performance of the fuzzy system. Table 2 shows the advantages and the disadvantages for fuzzy systems and genetic algorithms. The genetic algorithm was used for its ability to learn [2]. Fig. 8 shows the structure of the hybrid geno-fuzzy control system that was used to control the robot.

Table 2

Advantages and drawbacks of fuzzy logic and GAS

Properties	Fuzzy systems	Genetic algorithm
Store knowledge	Explicit	None
learns	No	Ability to learn
Optimizes	None	Powerful
Fast	Yes	Yes
Handle nonlinearity	Yes	Yes

The general system architecture is composed of the mobile robot, the fuzzy control system, the evolution strategy that adapts the fuzzy membership functions and the rules base, a simulation environment (KIKS simulator for Khepera robots), a fitness function to evaluate the quality of robot behaviors, as is shown in Fig. 7. The environment is shown in Fig. 8, where the control task is to drive the robot along the grey track in order to reach the target point as fast as possible.

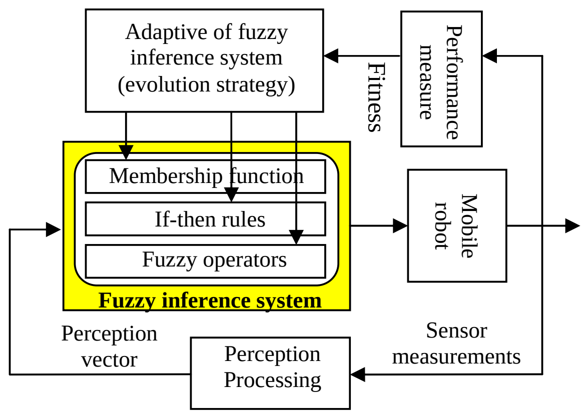


Fig.7. Architecture of the geno-fuzzy system to control a mobile robot.

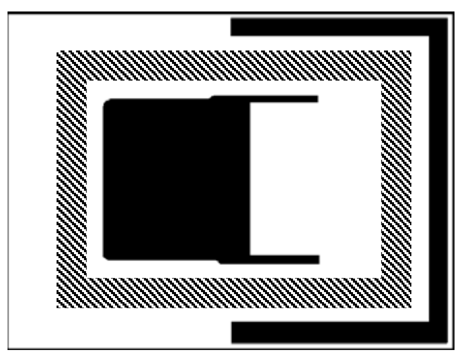
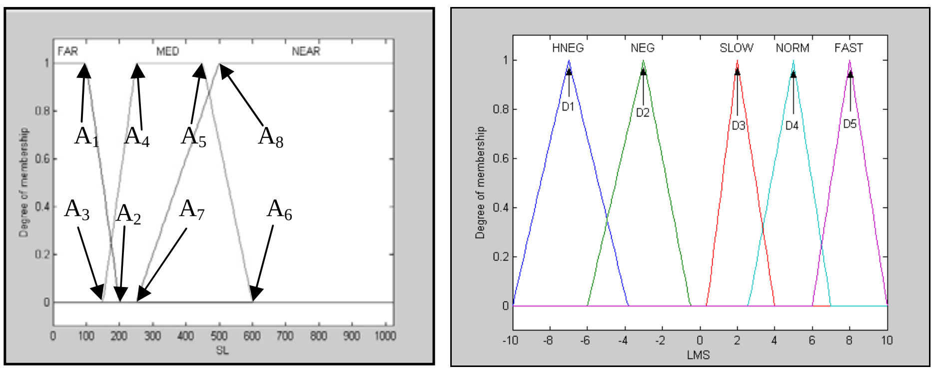


Fig.8. Simulation environment

4.1. Chromosome

In this paper is presented a novel chromosome encoding algorithm used to optimize the membership functions and the output rule base. So an optimal fuzzy control system is obtained which drove the Khepera mobile robot to achieve its target with good performance and optimal behaviors. We have encoded 34 parameters from the input and output membership function to form a chromosome segment in order to optimize these functions shape. Fig. 9 shows the eight genes that were encoded from S_{left} input function; these genes are called respectively A_1, A_2, A_3, A_4, A_5, A_6, A_7 and A_8. From the S_{front} and S_{right} input we encoded the two other chromosomes segment B and segment C, where each of them contain eight genes. They are named respectively B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, C_1, C_2, C_3, C_4, C_5, C_6, C_7 C_8, and Fig. 10 shows the membership functions chromosome segment. The genes D_1, D_2, D_3, D_4, D_5, E_1, E_2, E_3, E_4, and E_5 are encoded for tuning respectively the LMS and RMS output membership functions.



A_1	….	A_6	A_7	A_8	B_1	B_2	..	B_8	C_1	C_2	…	C_8	D_1	...	D_5	E_1	…	E_5

Fig.9. Chromosome's segment for encoding membership functions

In our fuzzy system we suppose 18 antecedence rules and the genetic algorithm optimizes the output of the rules base. The inputs values are predictable because the robot moved in its simulation environment. The chromosome output rules segment contained 36 genes that present the supposed LMS linguistic term and RMS linguistic term, as shown in Fig. 10.

<table>
<tr><td>F_1</td><td>F_2</td><td>F_3</td><td>F_4</td><td>…</td><td>…</td><td>F_{33}</td><td>F_{34}</td><td>F_{35}</td><td>F_{36}</td></tr>
<tr><td colspan="2">Output rule 1</td><td colspan="2">Output rule 2</td><td></td><td></td><td colspan="2">Output rule 17</td><td colspan="2">Output rule 18</td></tr>
</table>

Fig.10. Chromosome Segment F

Fig.11 shows the whole chromosome and his genes.

A_1	**--**	**A_8**	**B_1**	**--**	**B_8**	**C_1**	**--**	**C_8**	**D_1**	**--**	**D_5**	**E_1**	**--**	**E_5**	**F_1**	**F_2**	**--**	**F_{35}**	**F_{36}**

Fig.11.The whole chromosome

4.2. Multipoint crossover

The multi-point crossover is the best genetic operator method that can be used in this problem in order to increase the number of string segments exchanged. The parent chromosomes, P_1 and P_2, are cut virtually at multiple random locations, and the portions of the chromosome between the cuts were exchanged. The result is two offspring I_1 and I_2, as is shown in Fig. 12. We used multi-point crossover because the genes have integer values, the genes values of the output rules base are between 1 and 5, whereas the genes values of membership function chromosome segment are between 100 and 800, depending on the membership function itself. On the other hand, the genes had bounds in case to keep overlaps between the membership functions and the search for output rule base will be heuristic.

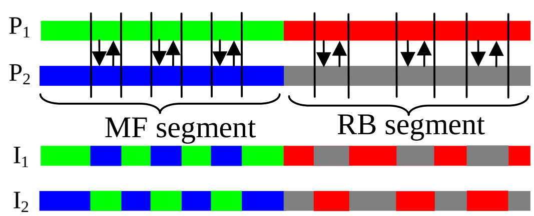


Fig.12. Multi-point crossover

4.3. Fitness function

This fitness function is a performance criterion that evaluates the performance of each chromosome. Higher fitness values are better when we want to maximize the function [5]. In this paper the fitness function trains the fuzzy control to optimize the robot path, thus the robot moved along its track with performance behaviors, and with the suitable speed without any collision. Practical the evolution process optimized the membership functions shape and the output rule base of the fuzzy controller.

The fitness function is:

$$F = S + Bon + A + TR + TI$$

And is calculated and summed over 1300 robot steps. Of its five components, *S* improves the speed; *A* improves collision avoidance, *Bon* gives the robot a bonus when it walks along the desired track. These three components are calculated and summed each robot step. *TR* and *TI* teach the robot to get to its target in a the shortest time possible. The time is either the number of steps that the robot needs to get to its target or is 1300 steps. The values of the *TI* and *TR* will be 0 when the robot doesn't get its target.

$S = \sum_{i=1}^{t} \frac{\lvert M_L + M_R \rvert}{2M_{MAX}} / t$	Where: - Mmax: maximum robot's speed (equal to 10); - (M_L, M_R): left motor and right motor speed - Smax: maximum sensor's reading (equal to 1023). - S_t: proximity-sensor (S_{left}, S_{front}, S_{right}) highest activity at step t. -*t*: the number of total steps
Bon= $\sum_{i=0}^{t} \frac{pp}{1300}$ pp= 1 if the robot is on it track pp=0 1 if the robot isn't on it track	
A $= \left[1\text{-}\sum_{i=0}^{t} \frac{S_t}{S_{MAX}} / t \right]$	
TR =1: If the robot gets the target TR=0: If the robot doesn't get the target	
$TI = \left[\frac{1300 - t}{1300} \right]$ If the robot gets the target *TI* = 0: If the robot doesn't get the target	

5. Experiments development and comparison of results

5.1 Experiments development

The genetic algorithm generates fuzzy parameters set for each population in any generation. The fuzzy controller drives the robot to make its task within a fixed time. The robot gets sensor data and then decides the suitable behavior. Avoiding obstacle will apply if there is an obstacle near the robot and if there is a wall then the robot will follow it. If the area is clear the robot seeks its target. The first priority is to keep the robot away from any obstacle, and then following wall in navigation mode and seeking target behavior. A flow chart for geno-fuzzy control system design is given in Fig. 13.

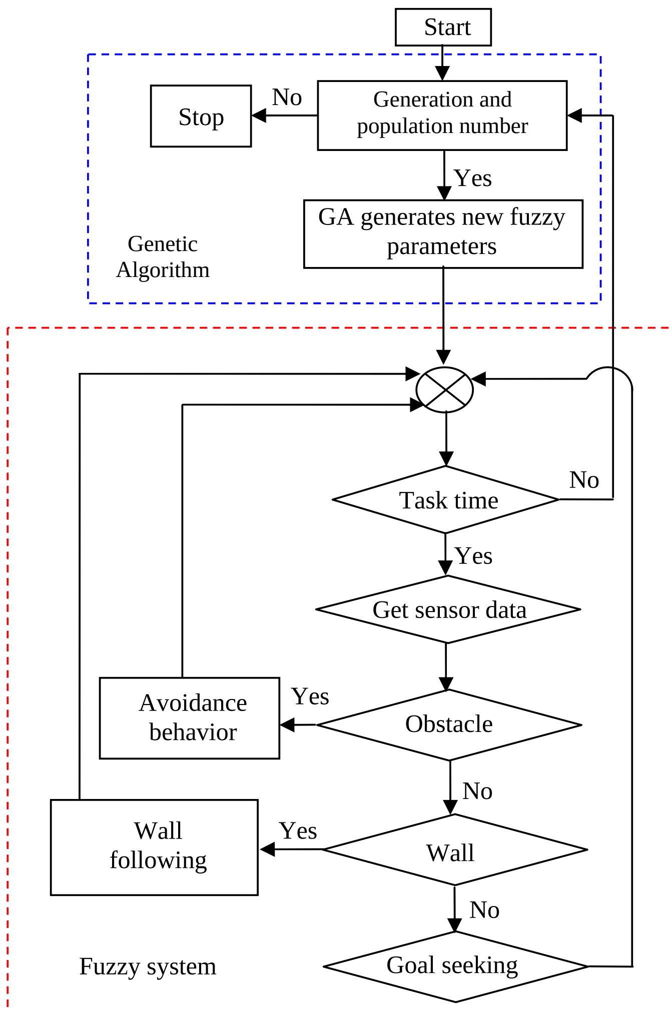
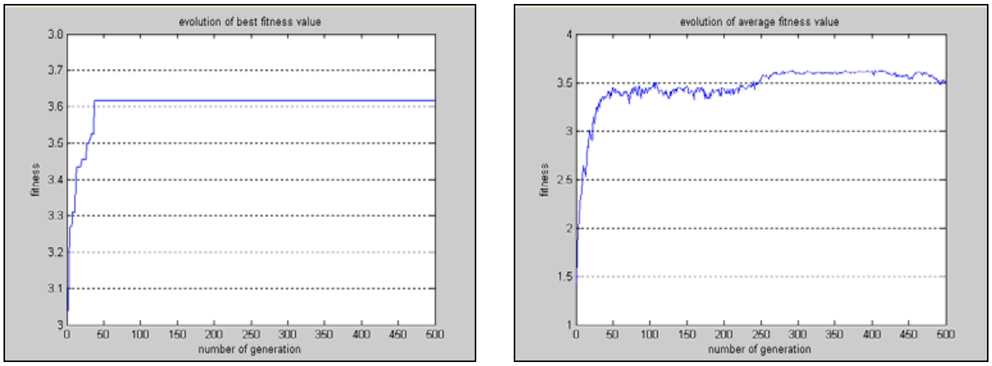


Fig.13. Flow chart for geno-fuzzy control system

We used the GAOT toolbox for Matlab. GAOT is a Genetic Algorithm Optimization Toolbox (GAOT) used for optimizing the fuzzy system. [20] In the evolution process we used the following parameters: Population size 50 individuals; crossover rate 80%; mutation rate 5%; number of generations 500.
Fig 14.A and 15.B show respectively the best chromosome fitness in each generation and the average of all the chromosomes in each generation.



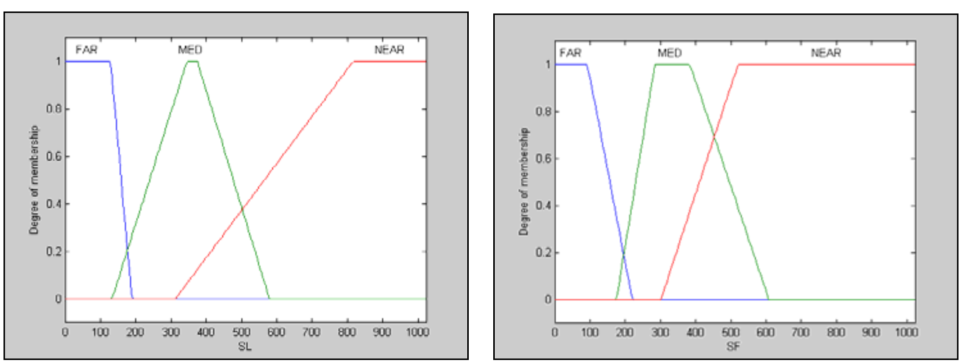
(A) (B)

Fig.14. A: Evolution of best chromosome, B: Evolution of average fitness chromosome

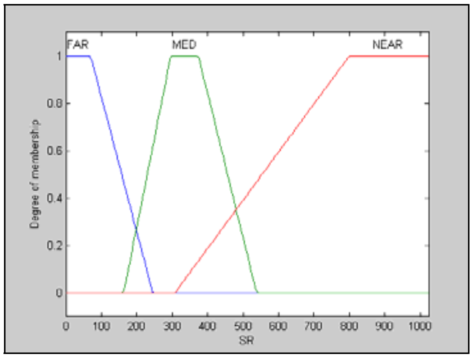
Figs. 15 (A, B, C, D, and E) show the optimal membership functions resulted after the evolutionary process. The figures are respectively for S_{left}, S_{front} and S_{right} sensor inputs, and for the LMS and RMS output membership functions. The optimal chromosome for the rule base output is as follows:

4 5	4 4	4 4	3 5	5 4	5 4	3 5	3 2	5 2
R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9
2 5	2 4	1 4	5 1	1 4	1 4	1 4	4 5	5 5
R_{10}	R_{11}	R_{12}	R_{13}	R_{14}	R_{15}	R_{16}	R_{17}	R_{18}

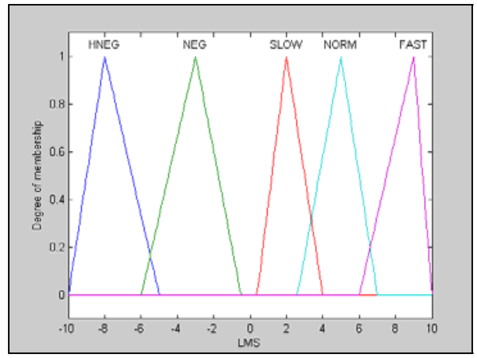
Where: 5, 4, 3, 2 and 1 represent respectively (HNEG, NEG, SLOW, NORM and FAST) that are the linguistic terms of the output fuzzy system.



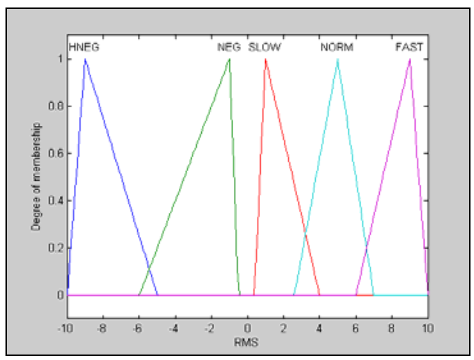
(A) (B)



(C)



(D)



(E)

Fig.15. A: The optimal S_{left} input membership function, B: The optimal S_{front} input membership function, C: The optimal S_{right} input membership function, D: The optimal RMS output membership function, E: The optimal LMS output membership function

5.2 Comparison of results

We designed a fuzzy system to control the robot along a given track. The result was a poor performance of the robot, as are shown in Fig. 16(A and B), as the robot needs 105 seconds to get to the target, as shown in Fig. 16(C) and exceeded the track limits. The fuzzy system was modified and was tested in the Kiks simulation environment. Fig. 16(D) shows the robot’s trajectory which improved as the robot needs now 55 seconds to get to the target. The results of the geno-fuzzy control systems are the best as presented in Fig. 16(E). The robot moved smoothly along its track and it needs 39 second to get to the target. Thus the geno-fuzzy system improves the path following behavior with a very short time to reach the target.

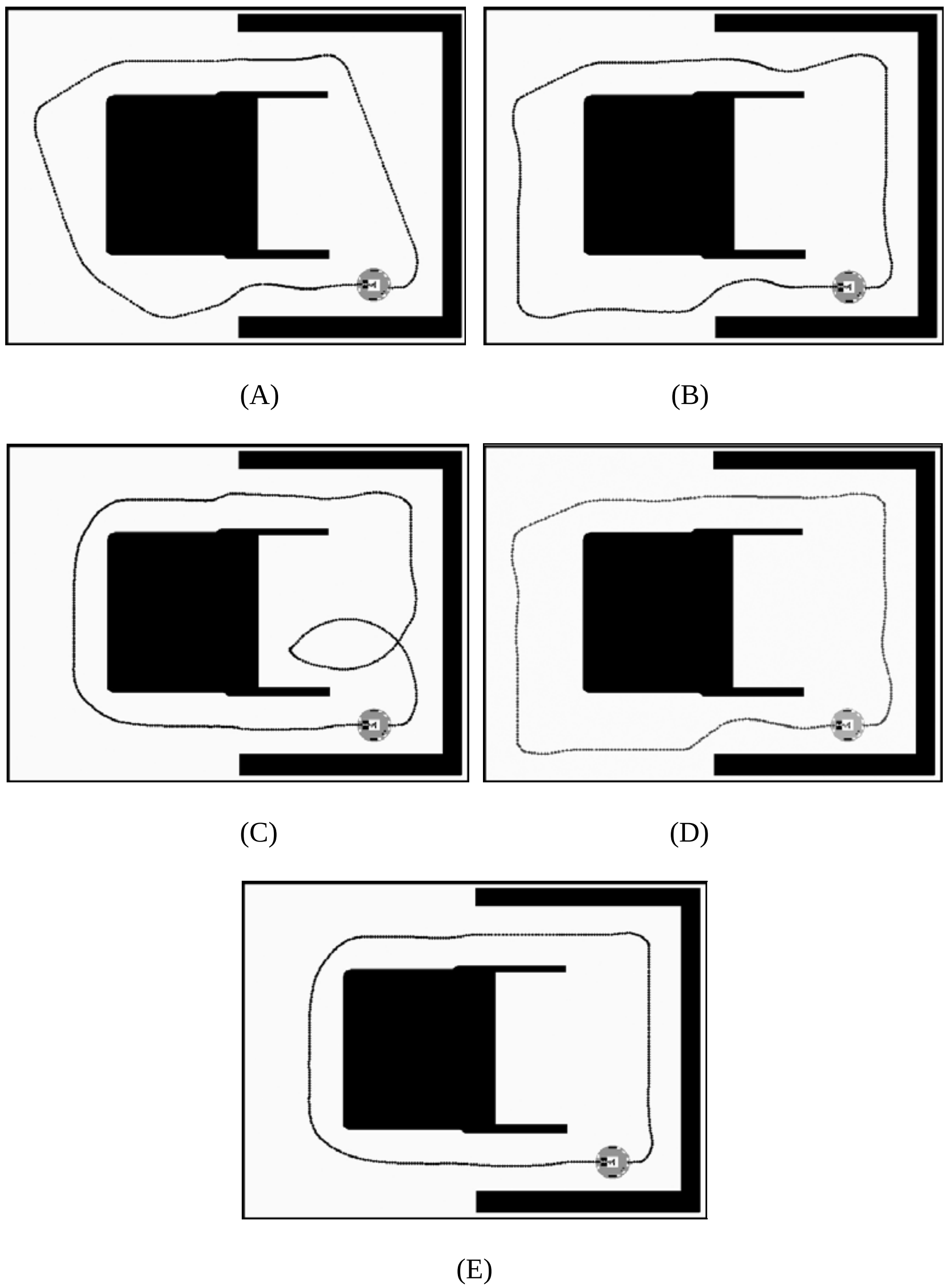


Fig.16**.** A and B: poor performance of the robot C: Robot with fuzzy system needs 105 seconds. D: Robot with modified fuzzy system needs 55 seconds, F: Optimal solution needs 39 seconds

6. Conclusions

In this paper we presented a hybrid geno-fuzzy control system for a mobile robot. Genetic algorithms are proven to be an efficient tool for designing an optimal fuzzy control system. The hybrid system optimized the membership function set and the output rule base on the fuzzy controller. The optimization improves the robot time performance and the robot behavior. The optimal solution has driven the robot on its track with a suitable speed. After the evolution process the robot walked fast along the corridor, its wall following ability improved significantly, while it managed to keep a suitable distance from the obstacles. Thus the optimal fuzzy system generated an optimal path towards the target.

R E F E R E N C E S

[1] *Jonathan Baber, Julian Kolodko, Tony Noel Michael Parent, and Liubo Vlacic*, cooperative autonomous driving intelligent vehicle sharing city roads, IEEE robotics & automation magazine 1070-9332/05/$20.00 ©2005

[2] *Fakhreddine O. Karray and Clarence De Silva,* Soft Computing and Intelligent Systems Design: Theory, Tools and Applications, Addison Wesley 2004 ISBN: 0321116178

[3] *Abduladheem A. Ali, Easa A. Abd,* Genetic-fuzzy inverse controller for a robot arm suitable for on line applications, transactions on engineering, computing and technology V9 November 2005

[4] *R. A. Brooks*, A robust layered control system for a mobile robot, IEEE Journal of Robotics

[5] *Stefano Nolfi, Dario Floreano*, evolutionary robotics: the biology intelligence and technology of self organizing machines (intelligent robotics and autonomous agents), 2000 ISBN: 0-262-14070-5

[6] *A. Abraham,* Hybrid Intelligent Systems: Evolving Intelligence in Hierarchical Layers, StudFuzz 173, 159–179 (2005) © Springer-Verlag Berlin Heidelberg 2005

[7] *Frank Hoffmann,* Evolutionary algorithms for fuzzy control system design, Proceedings of the IEEE, 2001, to appear in special issue on Industrial Innovation using Soft Computing.

[8] *Frank Hoffmann*, Soft Computing Techniques for the Design of Mobile Robot Behaviours, © Elsevier Science Inc., 1994, NY 10010 0020-0255/94/$6.00

[9] Michel Pasquier, Li How Lim, "fuzzy world 2-reactive beheaviours and environment exploration strategies" The fifth international conference on control, automation, robotics and vision (ICARCV 98)

[10] *Michael A. Lee and Hideyuki Takagi,* Integrating design stages of fuzzy systems using genetic algorithms This paper appeared in the Proc. of the 2nd Int'l Conf. on Fuzzy Systems (FUZZ-IEEE'93), Vol.1, pp. 612-617 (1993)

[11] *Randy L. Haupt, Sue Ellen Haupt*, Practical Genetic Algorithms, John Wiley & sons, inc., publication second edition 2004 ISBN: 0471455652

[12] *Simon X. Yang, Xiaochuan Wang, Guoyin Wang Max Q.-H. Meng* ,An Embedded Genetic Fuzzy Motion Controller for a Mobile Robot, Copyright @ 2005 IFAC

[13] *Yuchao Zhou*, an Area Exploration Strategy Evolved by Genetic Algorithm, Master Thesis, of Science Athens, Georgia 2005.

[14] *Franz Rothlauf,* Representations for Genetic and Evolutionary Algorithms, © Springer-Verlag Berlin Heidelberg 2 edition (March 3, 2006) ISBN: 354025059X

[15] *E. Tunstel, T. Lippincott, and M. Jamshidi*, Behavior hierarchy for autonomous mobile robots: Fuzzy-behavior modulation and evolution, Intl. Journal of Intelligent Automation & Soft Computing, vol. 3, no. 1, pp. 37–50, 1997

[16] *Hartmut surmann, michail maniadakis*, learning feed-forward and recurrent fuzzy systems: a genetic approach, journal of system architecture 47 (2001) 649-662

[17] *Alan C. Schultz and John J. Grefenstette,* using a genetic algorithm to learn behaviors for autonomous vehicles, In Proceedings of the AIAA Guidance, Navigation and Control Conference, Hilton Head, SC, August 10-12, 1992.

[18] *Frank Hoffmann, Gerd Pfister,* A New Learning Method for the Design of Hierarchical Fuzzy Controllers Using Messy Genetic Algorithms, 1995.
http://citeseer.ist.psu.edu/cache/papers/cs/1463/http:zSzzSzHTTP.CS.Berkeley.EDUzSz~fhoffmanzSzifsa95.pdf/hoffmann95new.pdf

[19] Khepera "User manual" Version 5.02 K-Team Lausanne, 12 March 1999

[20] http://www.ie.ncsu.edu/mirage/GAToolBox/gaot

[21] http://www.tstorm.se/projects/kiks/

Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 48-3 (2015) 778–783

Multi-strategy spatial data acquisition missions using genetic algorithms

F. Halal, M. B. Zaremba

Département d'informatique et d'ingénierie
Université du Québec en Outaouais
Gatineau, QC, Canada

Abstract: Monitoring of biological and chemical pollutants in large bodies of water requires the acquisition of a large number of in-situ measurement by a mobile sensor platform. Critical to this problem is an efficient path planning method, easily adaptable to different control strategies that ensure the collection of data of the greatest value. This paper proposes a hybrid Genetic Algorithm (GA) based deliberative path planning algorithm using waypoints for a ship navigation trajectory. The algorithm combines many sampling strategies for navigation and collection of samples from different water pollutant classes to fit a suitable objective function. Multi point crossover is employed to produce a global path consisting of good samples segments from many strategies paths. The system has been developed and tested for inland water monitoring and planning the ship trajectory using moderate resolution satellite data.

Keywords: hybrid genetic algorithm, path planning, monitoring system, remote sensing, navigation control, heuristic search.

1. INTRODUCTION

Monitoring of environmental phenomena in inland waters requires measuring physical processes across the entire spatial domain (Singh et al, 2009). Remote detection techniques provide significant advantages in terms of spatial and temporal coverage and cost-efficiency. The remote sensing data have to be augmented and updated by the in situ measurements, due to the need for precise local measurements for the calibration of satellite imagery in varying water conditions The environment maps of large areas are often obtained through processing of multi-spectral satellite imagery, which can subsequently be used to obtain water pollutant models by applying such measures as Total Suspended Sediments (TSS) (Koponen et al., 2005) and maximum chlorophyll index (MCI) (Gower et al., 2008).

In this paper we present a GA method developed to optimize path planning and navigation based on waypoints. Due to a large size of Lake Winnipeg – the site that relates to our experiments – combined with the requirement to acquire different samples following different acquisition strategies - conventional path planning does not give satisfactory results (Ragavan et al., 2011). The problem can be reduced to a simpler form by applying path planning waypoints. Waypoints are usually abstract points (Taha et al., 2009) used to help to define local paths through which a mobile platform can navigate, reach its region-of-interest destination, and collect the water pollutant samples (Park et al. 2013).

The path finding problem has typically been defined in terms of a Travelling Salesman Problem (TSP) and the Vehicle Routing Problem (VRP), and solved using evolutionary algorithms employed in with variants, such as ant colony (Colmenares et al., 2014), heuristic search D* algorithm (Luo et al., 2013) and genetic algorithms (Yoshikawa et al., 2009).

Genetic Algorithms have been frequently used for their flexibility and high quality of the search results (Samadi et al., 2013) in NP-hard problems. They can solve the problem without any advance knowledge about the environment, and are largely unconstrained by the limitations of the classical search methods (Rothlauf, 2006). By mimicking natural evolution processes they have the ability to adaptively search large spaces in near-optimal ways. In practical terms, GA methods are easy to interface with exciting simulation models.

Still, GAs come with some shortcoming First, the initial random search generates many infeasible and useless paths. Second, genetic operators, relying on heuristic knowledge, are not sufficient. Third, new offsprings may contain infeasible paths, (Yun et al. 2011). In order to overcome these disadvantages, a hybrid genetic algorithm was proposed to improve the genetic algorithm performance, (Yun et al. 2010). The initial populations, used to create the optimal solution, are generated based on multiple constraints. Since a high number of constraints are involved in the optimization process, many search strategies are applied to form the initial population and maintain population diversity of the genetic algorithm (Xiao-ting et al., 2013). Weak initial population leads to bad and unfeasible solutions.

The objective of this work is to find the optimal path planning for the sample acquisition platform in order to maximize the total quantity and quality of water samples. In general, each problem to be solved requires a unique fitness function that represents a performance criterion used in the evaluation of the performance of all chromosomes in a population. Many factors, such as travelling distance, time window and the sample values (weight) are involved in the optimization process. Thus multi-objective functions are used to find the suitable solution which fits the overall goal.

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.06.177

The Multi Point Crossover (MPC) was the method applied in this paper in order to deal with multiple sampling areas. Depending on the multi objective function and the initial population, the MPC operator works to build the final solution which consists of valuable segments, local paths, and many search strategies. The mutation operator helps the population to avoid local minima. The evolution process optimizes the path planning by designing new chromosomes which consist of best value samples from many global paths.

The structure of the paper is as follows. The second section reviews multi-spectral satellite sensors applied to water pollutant classification, and presents data acquisition mission strategies. A discussion of the proposed GA architecture is presented in Section 3. The initial population and path planning generators are presented in the fourth section. The optimization process is discussed in Section 5. The experiment setup and the obtained results are given in the Section 6.

2. ACQUISITION MISSION PROCESS

2.1 Pollutant sample acquisition

Medium resolution satellite imaging instruments, such as NASA's MODIS (Moderate Resolution Imaging Spectroradiometer) or MERIS (Medium Resolution Imaging Spectrometer), are typically used to monitor inland waters. MODIS has 36 spectral bands with center wavelengths ranging from 0.412 µm to 14.235 µm. MERIS has 15 spectral bands, optimized for chlorophyll detection.

Water wave reflection can be exploited to determine the concentration of the chlorophyll pigment and TSS. The classification of the water spectral characteristics is performed through the analysis of the shape of specific regions of the spectral curve.

In our tests, MERIS reflection shape features for Lake Winnipeg were divided into two dominated water classes. The first one relates to the level of chlorophyll-a concentration which has a peak at band 9 (705 nm), as shown in the bottom-left portion of Fig 1, and the second one represents the TSS concentration which has a flat portion of the curve from bands 5 to 9 (560nm - 705 nm), as shown in the bottom-right graph in Fig 1. It is representative of the water with high suspended matter concentration and high chlorophyll-a concentration.

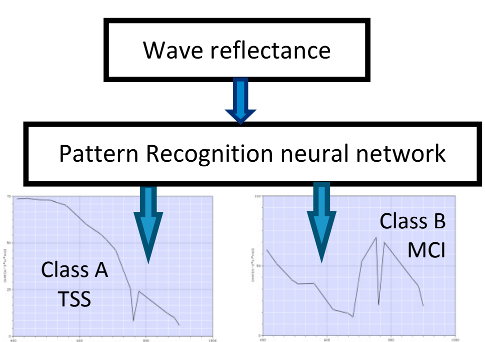


Fig. 1 Pattern recognition neural network classifies the pollutant water.

Each class has a specific spectral signature that reflects it spectral characteristics. A pattern recognition neural network was employed to classify the water into two types A and B, i.e., TSS and chl-a.

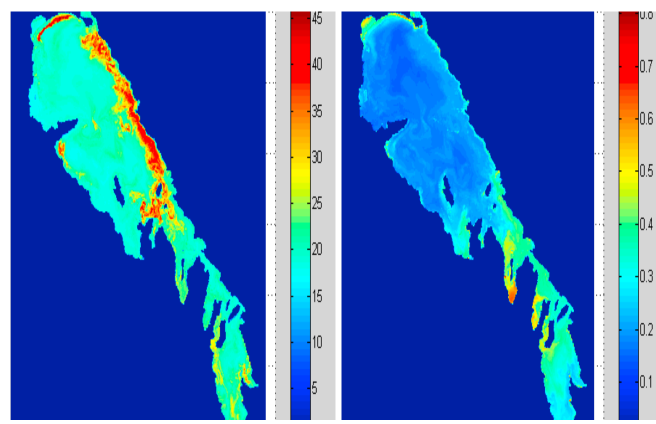
For each water class, a separate model was obtained and used for the assessment of the water pollutant. Equation (1) has been used to measure TSS:

$$TSS = 53.7\left(L_{709} / L_{560} + L_{665}\right) - 17.0 \tag{1}$$

where L_{709}, L_{560} and L_{665} denote the wavelength of 709 nm, 560 nm and 665 nm respectively. Equation (2) represents a MCI calculation based on (Gower et al., 2008).

$$MCI = L_{709} - L_{681} - 0.389\,(L_{753} - L_{681}) \tag{2}$$

where L_{xxx} is the radiance value of the band at wavelength xxx. The factor 0.389 is calculated as the wavelength ratio (709–681) / (753–681). Fig 2a represent Lake Winnipeg MCI map and Fig 2b shows the TSS map. The maps were taken and processed using VISAT-beam.



a) b)

Fig. 2. Lake Winnipeg maps: a) MCI, b) TSS.

2.2 Data acquisition mission

The acquisition mission can vary depending on the water pollutant samples distribution, their collection cost and the objective function which evaluates the samples value (weight) and the environment conditions. The goal is maximizing the quantity and the quality of the collected water pollutant samples during the mission. The strategies can be classified in two conflicting categories (Halal et al., 2010):

(1) Sampling the maxima of the water pollutant concentration distribution.

(2) Sampling which imposes time constraints on the acquisition mission.

In the first group, the sampling strategies include uniform coverage of high-concentration areas, sampling at local concentration maxima, and sampling along maximum gradient lines. Time windows can be imposed for the chlorophyll concentration sampling. Samples can be kept in the dark on ice only for a limited time without any degradation. The time ranges from hours to days depending on the sample sensitivity. Holding time is an important consideration because time-sensitive samples may need to be filtered in the field and placed on dry ice (Berkman et al., 2007). In the second group, one strategy considers the fact that specific samples should be collected while neglecting the others samples within a certain time window. Those samples can be treated as more valuable than the others. The path planning maximizes the value of the collected samples along its trajectory during the task. Depending on the time and the distance constraints, many pollutant patches could be neglected. The third strategy represents a hybrid between the two conflicting strategies, where time constraints are imposed on certain patches such as the chlorophyll concentration, and no constraints on other patches. A certain number of samples has to be collected in a specific patch before heading to another patch. This strategy has the advantages of both of previous ones. The path planning widely navigates collecting the valuable samples from local maxima in a way that fits the acquisition constraints.

3. HYBRID GENETIC ALGORITHM

In this paper hybrid genetic algorithms of optimum path planning for water sample collection are proposed. Two methods are introduced to improve the genetic algorithm performance: the obstacle avoidance algorithm and the pruning algorithm collaborate to produce feasible initial population. Infeasible paths are deleted during the evolution of the genetic algorithm, which improves the path planning efficiency.

3.1 Genetic Algorithm architecture

The basic operation of the proposed genetic algorithm can be summarized as follows (Fig. 3):

Step 1: Random waypoints

The random waypoints approach is applied in the deliberative navigation using the three strategies to generate many global paths. The global path consists of several local paths, which are the arcs between two waypoints (samples) that have a direct connection between them. One direction is considered for local paths pointing toward the target.

Step 2: Remove Redundant Point Algorithm: An algorithm screens unnecessary waypoints to generate a free collision path.

Step 3: Pruning algorithm

No-sample waypoints are removed. Each waypoint can be a sample collation station, this algorithm screens out, with the waypoints which don't contain valuable samples in their neighbour.

Step 4: Initial feasible population: The collision free path is stored by adding the chromosome to the initial pool population. The unfeasible solution will be deleted; Step 5: Definition of the objective function; Step 6: Multi-point crossover operation; Step 7: Mutation operation; Step 8: Termination: If the solution meets the termination criteria, the evolution will be stopped. Otherwise, the evolution will continue until the maximum generation number is reached.

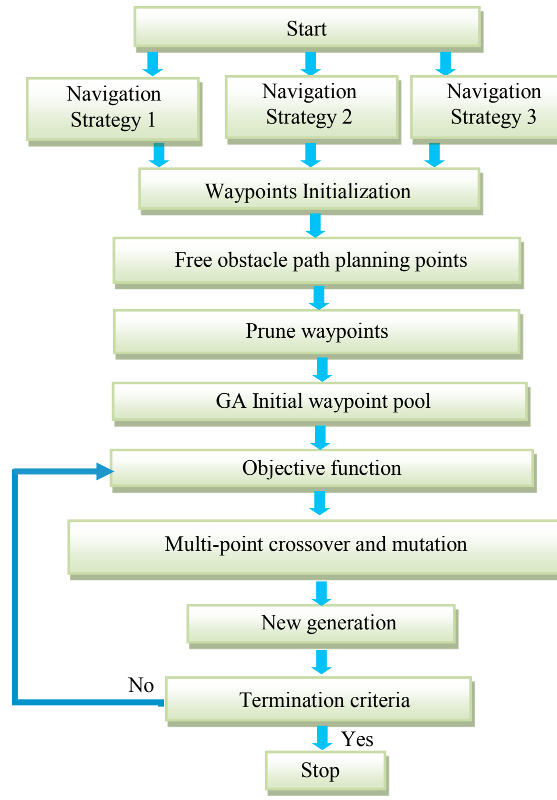


Fig. 3. Genetic Algorithm path planning flow chart.

4. PATH PLANNING AND INITIAL POPULATION

4.1 Waypoint initialisation

The population is represented by many ordered set of the waypoints. Each feasible set is considered to be an individual in the population. Each waypoint represents a location in the environment, and is characterized by an identifier - (x,y) coordinates. The initial genotype can be represented by a cell array, given the fact that each pair of cells represents the length and the heading angle between two sequential waypoints. In this work, a heuristic approach is proposed to find the optimal path. The search for the optimal path starts by randomly planning local path segments, which vary in length and heading angle. Path planning generator works as follows: 1) The first waypoint in the path is the starting point with the initial angle as zero. 2) While the path planning doesn't get the desired target, generate a random number of l, path length, between $[l_{\min}, l_{\max}]$, and a random β, heading angle between $[\beta_{\min}, \beta_{\max}]$ obtaining the next waypoints (Xiao-ting et al, 2013). A maximum number of waypoints is defined for each search strategy. 3) The waypoint represents a water sample candidate. Thus, many strategies can be applied to water pollutant patches by adjusting L and β. Each path planning strategy handles different number of samples (waypoint) depending on the search path. Figure 4 illustrates a path planning generator.

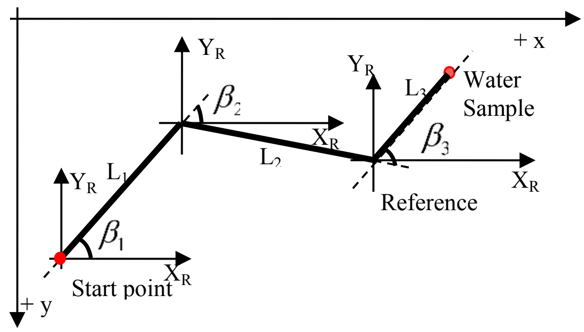


Fig. 4. Path planning generator.

Genotype

The chromosomes are encoded in an integer string. Each gene consists of two variables, the local path length and the heading angle. The path planning waypoints in the environment are represented in a long chromosome. The encoding technique uses the previous chromosome with the start point to identify the waypoints with (x,y) coordinates as shown in Fig. 5.

a)

Heading angle	travel distance	Heading angle	travel distance	……………	Heading angle	travel distance

b)

Starting point	Node 1	Node 2	..	..	..	…..	Node n	Destination point

Fig. 5. a) Local paths polar coordinate Chromosome, b) Waypoint chromosome.

4.2 Redundant waypoints removal

The redundant waypoints removal algorithm is designed for distinguishing whether the path is feasible or not. Fig. 6 illustrates this algorithm.

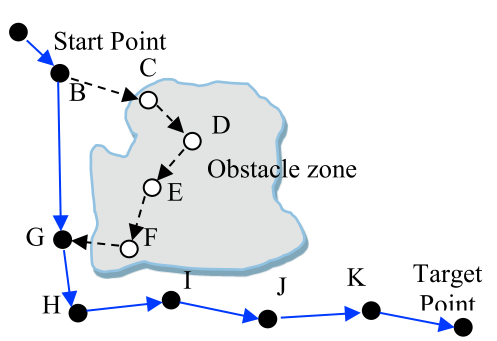


Fig. 6. Obstacle avoidance algorithm.

While the feasible path will be saved in the GA initial population, the infeasible path will be deleted. The basic idea of this algorithm is to verify and delete any local path hitting an obstacle area. As shown in Fig. 6, a new path BG is free obstacle; a deletion operation is applied to C, D, E and F waypoint. The algorithm deletes, for example, any waypoint that fall in the land area.

5. OPTIMIZATION

5.1 Fitness function

Each optimisation process needs an objective function with multiple constraint factors. The fitness function is a particular type of the objective function that quantifies the optimality of a solution and evaluates the suitability of a solution with respect to the overall goal. The main issue here is to define an appropriate fitness function that serves as an adequate representative of the optimization process.

The region of interest approach was used to identify the study zones and their boundaries, (Park et al., 2013). This approach prunes the search area by selecting many subsets of samples within the dataset representing the analysed area.

The proposed fitness function F consists of 3 components:

$$F = SV + ROI_A + DIS \qquad (3)$$

SV - data set value, which determines the sample value; ROI_A - the region of interest award, controls the sample quality and helps the platform reach its destination;

DIS - distance factor, which has an important role in improving the path length and in teaching the platform how to reach the target point at the short path.

These components are calculated and summed with each collected sample. Figure 7 summarizes the fitness function components.

$$SV = \left[\frac{\sum_{i=1}^{t} samples\ value}{\text{specific number of collected sample}} \right]$$

$$ROI_A = \left[\frac{\sum_{i=1}^{t} ROI\_samples}{\text{specific number of collected sample}} \right]$$

ROI = 1 if TSS value > 0.5
ROI= 0 if TSS value < 0.3

$$DIS = \left[1 - \frac{\sum_{i=1}^{t} \sqrt{(y_t - y_{t-1})^2 + (x_t - x_{t-1})^2}}{\text{maximum allowable distance}} \right]$$

Fig. 7. Fitness function components.

5.2 *Multi-point crossover*

Multi-point crossover was used to increase the number of swapped string segments and to reduce the size of each exchanged segment. Multi-point crossover operates in the global path planning phase. Selection of the waypoints consists of the following three steps:

Step 1: Select the best fitness chromosome (Parent 1) and another (Parent 2) from the feasible pool as shown in Fig 8. Step 2: Use a bounded range to select the nodes as multi crossover points. Due to the difference in the chromosome length, the crossover point should be applied to the short chromosome. Step 3: Swap the contents between two sequential crossover points of two parent individuals. The waypoints between $node_1$ and $node_2$ have to be swapped. The next segments to be exchanged are between $node_3$ and $node_4$, and so on. Figure 9 explains the crossover procedure.

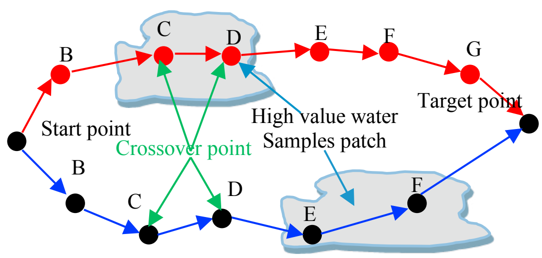


Fig. 8. Two chromosomes (Parents).

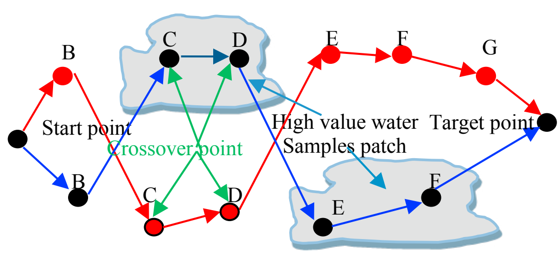


Fig. 9. Two offsprings from (Parents).

6. EXPERIMENTAL RESULTS

6.1 *Experimental setup*

Three classes of water pollutant patches are defined, which are as follows: medium and high concentration of TSS, and a medium concentration of chlorophyll-a. Medium TSS concentration zone (M_TSS): the patch axis start at longitude (97˚ 05' 55") W and (51˚ 57' 59") W, and finished at longitude (97˚ 14' 41") W and latitude (51˚ 42' 20") N. The width range is 5 km to 10 km. High TSS concentration zone (H_TSS): patch length 10 km, and the width varies from 3.7 km to 7 km. The search start point is longitude (97˚ 15' 57") W latitude (51˚ 27' 43") N, the target point is longitude (97˚ 11') W and latitude (51˚ 32' 38") N. Chlorophyll concentration zone (CHLO-A): the search starts at longitude (97˚ 13' 20") W and latitude (51˚ 13' 14") N. The target point is longitude (97˚ 10' 43") W and latitude (51˚ 58' 18") N. The patch axis passes through these two points. Figure 10 illustrates the water pollutant patches. The above points represent the patch entrance and the patch exit waypoints. Many strategies have been applied in order to maximize the data set value and to navigate an optimal patch subject to the imposed constrains.

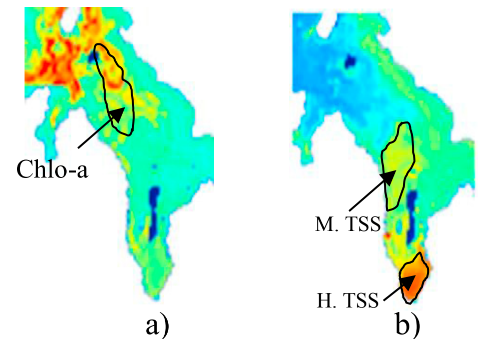


Fig. 10. Acquisition patches.

a) Chlorophyll-a patch, b) TSS patches

Three strategies were applied on each patch in order to generate the initial GA population. In the first strategy, the platform navigates perpendicularly to the patch axis, which means the heading angle varies in the range ± 90º. This strategy scans the patch searching for the local maxima. The path planning neglects the traveling distance and time. The second strategy initially navigates around the patch axis looking for the maxim value samples, minimizing the path travel distance. The heading angle ranges ± 35º. The third strategy represents the shortest distance path planning, from the start point to the target point, where the samples have the lowest value.

The GAOT Matlab toolbox was applied to optimise the final data acquisition platform path planning. The population size varies from one path to other, the crossover rate was 80%, the mutation rate was 5%, and the generation number 500.

6.2 *Test results*

Different results were obtained by changing the objective function factors. The multi-point crossover mixes up the paths and gets the optimal solution depending on the objective function factors. Figure 11a shows the optimal path planning for the navigation and collection of samples from three different H_TSS, M_TSS and chl-A patches, which has many segments belonging to different strategies.

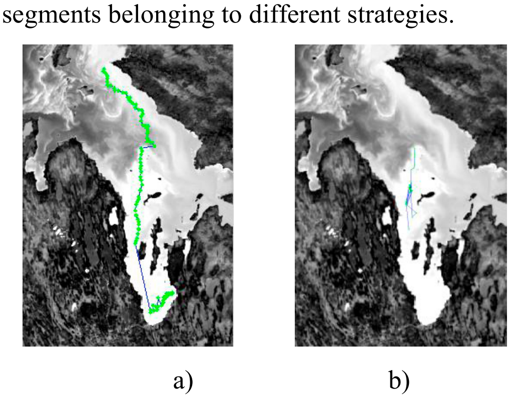


Fig. 11. Path planning optimisation process.

Multi-point crossover and multi-point mutation are applied. The trip length is 129680 m, and 94 samples are collected as follows: 27 H_TSS, 31 M_TSS and 36 Chl-A. Figure 11 b) shows the path planing for sampling M_TSS for 84469 m. Twenty-seven samples were colllected. A simple crossover and one waypoint mutation were applied.

The tests have demonstrated that waypoint navigation strategies help to find a suitable initial population which leads to an optimal solution or near optimal solution, depending on a suitable genetic algorithm operators and appropriate fitness function.

7. CONCLUSIONS

Genetic algorithms have demonstrated their usefulness in solving multi-objective path planning problems, where the mobile platform navigates an unknown environment. In this paper, a hybrid waypoint/GA algorithm was presented. The data acquisition platform path maximises the data set quantity and quality along a set of local paths, which are defined within local region-of-interest areas. The optimization process assists the planner in creating the optimal path by deleting unfeasible and obstacle collision waypoints. Each global path handles different number of water samples. Different acquisition strategies were investigated and tested for acquiring water pollutant samples in inland waters.

ACKNOWLEDGEMENTS

This work was supported by funding from ISTP and NSERC. The authors wish to acknowledge the support of Dr. Greg McCullough from CEOS, who provided in-situ spectral and chlorophyll-a concentration data.

REFERENCES

Colmenares, G. Halal, F. and Zaremba, M. B. (2014). Ant Colony Optimization for Data Acquisition Mission Planning. *Management and Production Engineering Review*, 5(2), 3-.11

Cui, S., and Dong, J., (2013), Detecting Robots Path Planning Based on Improved Genetic Algorithm. *Third International Conference on Instrumentation, Measurement, Computer, Communication and Control*.

Ergezer, H., Leblebicioglu, K. (2013). Path Planning for UAVs for Maximum Information Collection. *IEEE Transactions on Aerospace and Electronic Systems*. 49(1), 502-520.

Gower, J., King, S., Borstad, G., and Brown, L. (2008). The Importance of a Band at 709 nm for Interpreting Water-Leaving Spectral Radiance. *Canadian J. Remote Sensing*, 34(3), 287-295.

Halal, F., and Zaremba. M., (2010). Hybrid Geno-Fuzzy Control for Deliberative-Reactive Navigation. *IFAC workshop on Intelligent Control Systems*, Sinaia, Romania, 45-51.

Halal, F., Pedrocca, P., Hirose, T., Cretu, A.-M., and Zaremba, M. (2014). Remote-Sensing Based Adaptive Path Planning for an Aquatic Platform to Monitor Water Quality. *IEEE Symp. Robotic and Sensors Environments*, Timisoara, Romania, 43- 48

Hambrook-Berkman J.A., and Canova M.G., (2007) *Algal Biomass Indicators. U.S. Geological Survey*, TWRI Book 9, Version 1.0.

Haupt, R. L., and Haupt, S. E. (2004). *Practical Genetic Algorithms*, John Wiley & Sons, ISBN: 0471455652

Koponen, S., Vepsäläinen, J., Pulliainen, J., Kallio, K., Pyhälahti, T., Lindfors, A., Rasmus, K., and Hallikainen, M. (2005). *Using MERIS data for the retrieval of CHLA, CDOM AND TSS values in the Gulf of Finland and Lake Lohjanjärvi*, MERIS-(A) ATSR Workshop. ESA, Frascati, Italy.

Luo, C., Yang, S. X., Krishnan, M., Paulik, M., and Chen, Y., (2013). A Hybrid System for Multi-goal Navigation and Map Building of an Autonomous Vehicle in Unknown Environments. *Proc. IEEE International Conference on Robotics and Biomimetics*, Shenzhen, China. 1228-1233.

Park, S., and Cho, J., (2013) ROI-based Visualization of Spatial Information for a Remote-controlled Robot. *10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Jeju, Korea, 234- 235

Ragavan, S.V., Ponnambalam, S.G., and Sumero, C. (2011).Waypoint-based Path Planner for Mobile Robot Navigation Using PSO and GA-AIS, 978-1-4244-9477-4/11©2011 IEEE 756- 760

Rothlauf. F., (2006). *Representations for Genetic and Evolutionary Algorithms*. Springer-Verlag, Berlin.

Samadi, M., and Othman, M. F., (2013). Global Path Planning for Autonomous Mobile Robot using Genetic Algorithm. *International Conference on Signal-Image Technology & Internet-Based Systems*.726-730.

Singh, A., Krause, A., Guestrin, C., and Kaiser. W.J., (2009). Efficient Informative Sensing using Multiple Robots. *Journal of Artificial Intelligence Research*, 34, 707-755

Su, J., and Li, J., (2013). Path Planning for Mobile Robots Based on Genetic Algorithms. *Ninth International Conference on Natural Computation (ICNC)*. 723-727.

Taha, M., Ibrahim, S., Ragavan, S. V., and Ponnambalam SG. (2009).Way Point Based Deliberative Path Planner for Navigation. *IEEE/ASME Int. Conference on Advanced Intelligent Mechatronics* Singapore. 881-886.

Xiao-ting, J., Hai-bin, X., Li, Z., and Sheng, D. J. (2013). Flight path planning based on an improved genetic algorithm. *Third International Conference on Intelligent System Design and Engineering Applications*, 775-778.

Yoshikawa, M., Terai, H. (2009). Car Navigation System based on Hybrid Genetic Algorithm. *World Congress on Computer Science and Information Engineering*, 62-65

Yun, S. C., Ganapathy, V., and Chong, L. O. (2010). Improved Genetic Algorithms based Optimum Path Planning for Mobile Robot, *11^{th} Int. Conf. Control, Automation, Robotics and Vision,* Singapore, 1565-1570.

Yun, S. C., Parasuraman S., and Ganapathy, V. (2011) Dynamic Path Planning Algorithm in Mobile Robot Navigation. *IEEE Symposium on Industrial Electronics and Applications*, Langkawi, Malaysia, 364-369

Zhou, Y., (2005), *an Area Exploration Strategy Evolved by Genetic Algorithm*, M.Sc. Thesis, Athens, Georgia.