A Genetic learning Motion planning of an Autonomous Mobile Robots

Hachour Ouarda

Abstract— This paper describes how soft computing technology as Genetic Algorithms (GAs) can be applied for path planning of an Autonomous Mobile Robot (AMR). GAs are search algorithms based on the mechanics of natural genetics. They combine survival of the littlest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. The proposed GA approach has an advantage of adaptivity such that the GA works perfectly even if an environment is unknown. . These environments were randomly generated . While randomized, GAs are no simple random walk. They efficiently exploit historical information to speculate on new search points (sub path positions) with expected improved performance. We measure the number of generations of candidates. The coding of GA is to affect label 0 for free cell and 1 for hazardous cell. This way of work is very useful later if the substring is inherited to new generations by genetic operators. The objective is to find a feasible and flexible path from initial area source to destination target area, flexible because the user can change the position of obstacles it has no effect since the environment is unknown. This robust method can deal a wide number of environments and gives to our robot the autonomous decision of how to avoid obstacles and how to attend the target. More, the path planning procedure covers the environments structure and the propagate distances through free space from the source position. For any starting point within the environment representing the initial position of the mobile robot, the shortest path to the goal is traced. The results gotten of the GA on randomly generated terrains are very satisfactory and promising.

***Keywords*—** Genetic Algorithm (GA), Motion Planning, Autonomy requirements, Autonomous Mobile Robot (AMR), Intelligence.

I. INTRODUCTION

Autonomous mobile robots which work without human operators are required in robotic fields. In order to achieve tasks, autonomous robots have to be intelligent and should decide their own action. When the autonomous robot decides its action, it is necessary to plan optimally depending on their tasks. More, it is necessary to plan a collision free path minimizing a cost such as time, energy and distance.

When an autonomous robot moves from a source position to a target position, it must find a feasible connection between the source and the target. In other word: It is necessary to plan an optimal or feasible path avoiding obstacles in its way and answer to some criterion of autonomy requirements such as: thermal, energy, time, and safety for example. Therefore, the major main work for path planning for autonomous mobile robot is to search a collision free path. Many works on this topic have been carried out for the path planning of autonomous mobile robot [11 ,12].

To operate independently in unknown or partially known environments is the basic feature of an autonomous mobile robot. The autonomy implies that the robot is capable of reacting to static obstacles and unpredictable dynamic events that may impede the successful execution of a task. To achieve this level of robustness, methods need to be developed to provide solutions to localization, map building, planning and control. The development of such techniques for autonomous robot navigation is one of the major trends incurrent robotics research.

In all research developments, the robot has to find a collision-free trajectory between the starting configuration and the goal configuration in a static or dynamic environment containing some obstacles. To this end, the robot needs the capability to build a map of the environment, which is essentially a repetitive process of moving to a new position, sensing the environment, updating the map, and planning subsequent motion.

The planner's approximation of the configuration space is a direct function of the configuration space samples that the planner observes. Consequently, sampling is the search for the set of samples that provide enough information to construct a sufficient approximation of configuration space connectivity. For every configuration space, there are an optimal number of samples that must be selected to construct a sufficient approximation of configuration space connectivity.

Local planners impose an artificial potential field function on top of the configuration space. This potential field function is sloped so that its minimum is at the goal configuration. The artificial potential field is also influenced by configuration space obstacles. Configuration space obstacles have high artificial potentials that decline gradually with distance from the obstacle. At any instance, the robot calculates the derivative of the potential function and descends the maximal downward gradient in an effort to reach the minimum at the goal position. This calculation quickly determines the motion to take next.

Several others guided sampling strategies use information obtained from previous experience to guide their behaviour. Entropy-guided sampling adapts sampling to find configurations that offer maximal information gain given the current state of the planner. The measure of information gain can be mathematically proven to converge (in the limit) on the desired state. Reinforcement learning achieves a similar outcome at a higher level.

Reinforcement learners develop a policy for action given the state of the agent. Because this policy applies to many different agent states, an agent using such a policy is partially robust to stochastic execution of its actions. Regardless of the state in which the planner finds itself after taking some action, the policy provides a subsequent action to take. Many reinforcement learning algorithms also converges on the optimal policy (given some reward function) after enough experience.

Motion planning algorithms have also been applied to an array of problems beyond traditional robotics. The two main areas of application are computational biology and computer games and animation. Though not the main focus of this thesis, these non traditional applications of planning may benefit from many of the improvements in performance and reliability offered by the utility-guided framework.

Motion planning algorithms find collision-free paths for robots in obstructed configuration spaces. Because the size of configuration space is often quite large, and in many cases the task of planning constrained by time, the efficiency of this search is a significant concern, especially in real-world motion planning. To be useful for real world robotics, a motion planner must be able to compute a path quickly. The actual time required depends on the robots task and how quickly the environment changes.

To plan efficiently despite the proven computational intractability of motion planning, sampling-based methods compute an approximate implicit representation of configuration space connectivity. This representation is constructed by sampling and observing a subset of all points in a particular configuration space. The planner's approximation of the configuration space is a direct function of the configuration space samples that the planner observes. Consequently, sampling is the search for the set of samples that provide enough information to construct a sufficient approximation of configuration space connectivity.

For every configuration space, there is an optimal number of samples that must be selected to construct a sufficient approximation of configuration space connectivity.

Path planning in spatial representation often requires the integration of several approaches. This can provide efficient, accurate, and consist navigation of a mobile robot. It is sufficient for the robot to use a topological map that represents only the areas of navigation (free areas , occupied areas of obstacles). It is essential the robot has the ability to build and uses models of its environment that enable it to understand the environment's structure. This is necessary to understand orders, plan and execute paths [4].

Path planning is one of the key issues in mobile robot navigation. Path planning is traditionally divided into two categories: global path planning and local path planning. In global path planning, prior knowledge of the workspace is available. Local path planning methods use ultrasonic sensors, laser range finders, and on-board vision systems to perceive the environment to perform on-line planning.

Motion planning is one of the important tasks in intelligent control of an autonomous mobile robot. It is often decomposed into path planning and trajectory planning. Path planning is to generate a collision free path in an environment with obstacles and optimize it with respect to some criterion. Trajectory planning is to schedule the movement of a mobile robot along the planned path. A wide variety of approaches have been considered, but these can broadly be categorized into on-line and off-line techniques. However, few algorithms have been developed for on-line motion planning in a time-varying or unknown terrain. The problem of the path planning has been studied extensively over the last decades [3,13,14]. Most great research application efforts have been spent on path planning in *static* environments. That is, a path has to be found between two configurations for a movable object in an environment containing stationary obstacles whose geometry and coordinates are given. Whereas less attention has been given for *dynamic* environments. Besides stationary obstacles, dynamic environments contain moving obstacles with which collisions must be avoided as well. As an example, a mobile robot operating at a factory floor will have to navigate among humans or other robots, which can be considered as moving obstacles. In general, we can clarify the path planning problem is in its most general form a geometric problem which is based on the following steps:

A description of the geometry of the robot.

. A description of the geometry of the environment or *workspace* in which the robot moves or operates.

. A description of the degrees of freedom of the robot's motion.

. An initial and a target configuration in the environment, between which a path is to be planned for the robot

Using these informations, we can construct the *configuration space* of the robot, in terms of which the path planning problem is formulated generally. Previous research on the path planning can be classified as following one of two approaches: model-based and sensor –based. In general, the model-base approach considers obstacle avoidance globally it uses prior models to describe known obstacles completely in order to generate a collision free path. In contrast, the sensor-based approach aims to detect and avoid unknown obstacles.

To detect and to avoid known, partially known or unknown obstacles, we need the theory and practice of intelligence and robotic systems are currently the most strongly studied and promising areas in computer science and engineering which will certainly play a primary role in future. These theories and applications provide a source linking all fields in which intelligent control plays a dominant goal. Cognition, perception, action, and learning are essential components of such systems and their integration into real systems of different level of complexity (from micro-robots to robot societies) will help to clarify the true nature of robotic intelligence

Intelligent Autonomous systems IAS designers search to create dynamic systems to navigate and perform purposeful behaviours like human in real environments where conditions are laborious. However, the environment complexity is a specific problem to solve since the environments can be imprecise, vast, dynamical, and partially or not structured.

Then, IAS must then be able to understand the structure of these environments. To reach the target without collisions, IAS must be endowed with recognition, learning, decision-making, and actions capabilities. IAS have many possible applications in a large variety of domains, from spatial exploration to handling material, and from industrial tasks to the handicapped helps. In fact, recognition, learning, decision-making, and action constitute principle problems of the obstacles avoidance of IAS. Three levels are required to recognition namely: inaccurate data processing (issued from sensors), construction of knowledge base, and establishments of an environment map. To solve these problems and remedy in sufficiency of classical approaches related to real-time, autonomy, and intelligence, current approaches are based on hybrid intelligent systems.

Path planning plays an important role in various fields of application and research, among which are CAD-design, computer games and virtual environments, molecular biology, and robotics. In its most general form, we can say that the main work of this level is to plan a feasible path for some moving mass between a start position and a goal position in some environment. A more challenging path planning problem occurs when the set of all possible states is not discrete as in the case of a grid, but continuous. To clarify more the idea, an industrial manipulator robot that has to move in a three-dimensional environment while avoiding collisions with itself and obstacles in the environment. The challenge in these cases is to discretize the problem in a sensible way, such that it becomes tractable.

Motion planning will frequently refer to motions of a robot in a 2D or 3D world that contains obstacles. The robot could model an actual robot, or any other collection of moving bodies, such as humans or flexible molecules. A motion plan involves determining what motions are appropriate for the robot so that it reaches a goal state without colliding into obstacles. When the autonomous robot decides its action, it is necessary to plan optimally depending on their tasks especially if it is a 3D environments complexity. To plan 3D collision free path is to find the capability to operate independently in unknown or partially known 3D environments complexity. The autonomy implies that the robot is capable of reacting to static 3D obstacles and unpredictable dynamic 3D events that may impede the successful execution of a task. To achieve this level of robustness, methods need to be developed to provide solutions to localization, map building, planning and control [1, 2, 3, 4, 5]. The development of such techniques for autonomous robot navigation is one of the major trends in current robotics research.

Many works on this topic have been carried out for the path planning of autonomous mobile robot. Because perfect information concerning the moving obstacles in the environment may not be available, it is important that partial information is adequately coped with. There are a number of existing methods for dealing with this scenario. In particular, we can estimate future trajectories of the obstacles based on current behaviour, or we can assume worst-case trajectories. Whichever of these we choose; we end up with some trajectory or set of trajectories that we can represent as 3D objects in the configuration-time space.

This paper deals with the intelligent path planning of AMR in an unknown environment, by applying the principles of the genetic algorithms. The aim of this paper is to develop an AMR for the AMR stationary obstacle avoidance to provide them more autonomy and intelligence .This technology GA based on intelligent computing as becoming useful as alternate approach may be able to replace the classical approaches such as: recognition, learning, decision-making, and action (the principle obstacle avoidance problems). GA has been theoretically and empirically proven to provide robust search capabilities in complex spaces offering a valid approach to problems requiring efficient and effective searching the proposed GA approach can be realized in efficient manner and has proved to be superior to combinatorial optimization techniques, due to the problem complexity.

Recently, applications of genetic algorithms to path planning or trajectory planning have been recognized. GA is a search strategy using a mechanism analogous to evolution of life in nature. The GAs, which are evolutionary have recently emerged from study of the evolution mechanisms and are searching strategies suitable for finding the globally optimal solution in a large parameter space. They are based on learning mechanisms. It has widely been recognized that GA works even for complex problems such that traditional algorithms cannot find a satisfactory solution within a reasonable amount of time.

GAs are search algorithms based on the mechanics of natural genetics. A genetic algorithm for an optimization problem consists of two major components. First, GA maintains a population of individual corresponds to a candidate solution and the population is a collection of such potential solutions. In general an individual GA is commonly represented by a binary string the mapping between solutions and binary strings is called a “coding”. The aim of this work is to propose a simple and efficient navigation approach for autonomous mobile robot. Note that, the algorithm described here is just to find a feasible and flexible path from initial area source to destination target area, flexible because the user can change the position of obstacles it has no effect since the environment is unknown. This robust method can deal a wide number of environments and gives to our robot the autonomous decision of how to avoid obstacles and how to attend the target. More, the path planning procedure covers the environments structure and the propagate distances through free space from the source position. For any starting point within the environment representing the initial position of the mobile robot, the shortest path to the goal is traced; the following sections clarify the principle of work.

II. Motion Planning

A. Necessity of intelligent autonomous Robot

A robot is a "device" that responds to sensory input by running a program automatically without human intervention. Typically, a robot is endowed with some artificial intelligence so that it can react to different situations it may encounter. The robot is referred to be all bodies that are modeled geometrically and are controllable via a motion plan. A robotic vehicle is an intelligent mobile machine capable of autonomous operations in structured and unstructured environment. It must be capable of sensing thinking and acting. The mobile robot is an appropriate tool for investigating optional artificial intelligence problems relating to world understanding and taking a suitable action, such as , planning missions, avoiding obstacles, and fusing data from many sources.

Industrial robots used for manipulations of goods; typically consist of one or two arms and a controller. The term controller is used in at least two different ways. In this context, we mean the computer system used to control the robot, often called a *robot work-station* controller. The controller may be programmed to operate the robot in a number of way; thus distinguishing it from hard automation. The controller is also responsible for the monitoring of auxiliary sensor that detect the presence, distance, velocity, shape, weight, or other properties of objects. Robots may be equipped with vision systems, depending on the application for which they are used. Most often, industrial robot are stationary, and work is transported to them by conveyer or robot carts, which are often called autonomous guided vehicles (AGV). Autonomous guided vehicles are becoming increasingly used in industry for materials transport. Most frequently, these vehicles use a sensor to follow a wire in the factory floor. Some systems employ an arm mounted on an AGV.

Robot programmability provides major advantages over hard automation. If there are to be many models or options on a product, programmability allows the variations to be handled easily. If product models change frequently; as in the automotive industry, it is generally far less costly to reprogram a robot than to rework hard automation. A robot workstation may be programmed to perform several tasks in succession rather than just a single step on a line. This makes it easy to accommodate fluctuations in product volume by adding or removing workstations. Also; because robots may be reprogrammed to do different tasks; it is often possible to amortize their first cost over several products. Robots can also perform many applications that are poorly suited to human abilities. These include manipulation of small and a large object like electronic parts and turbine blades, respectively. Another of these applications is work in unusual environments like clean rooms, furnaces, high-radiation areas, and space. Japan has led the world in the use of robots in manufacturing. The two sectors making heaviest use of robots are the automotive and electronics industries. Interest in legged locomotion has been stimulated by application in traversing rough terrain and in unmanned exploration of unknown environment. Aside from electronic motivation, there are many unanswered scientific.

Several autonomy requirements must be satisfied to well perform the tasks of AMR such as: Thermal, Energy, Communication Management, Mechanical design, etc. The development of such techniques for autonomous robot navigation is one of the major trends in current robotics research. Though there are many skills that are necessary for a robot to achieve in general. The autonomy is one of the most significant problems that a remains largely unsolved is the dexterous physical manipulation of objects in the world. Without the ability to interact with physical objects in a variety of sophisticated ways, robots can never be truly autonomous. Additionally, there are many practical robotic tasks, from elder care to building construction that require physical manipulation of the world. Human beings manipulate their world in a myriad of sophisticated ways on a daily basis. On any given day, entering the computer science building and attending to tasks at my desk requires a large number of sophisticated manipulations of physical objects.

B. Navigation

Navigation is the ability to move and on being self-sufficient. The AMR must be able to achieve these tasks: to avoid obstacles, and to make one way towards their target. In fact, recognition, learning, decision-making, and action constitute principal problem of the navigation. One of the specific characteristic of mobile robot is the complexity of their environment. Therefore, one of the critical problems for the mobile robots is path planning, which is still an open one to be studying extensively. Accordingly, one of the key issues in the design of an autonomous robot is navigation, for which, the navigation planning is one of the most vital aspect of an autonomous robot.

Navigation is the science (or art) of directing the course of a mobile robot as the robot traverses the environment. Inherent in any navigation scheme is the desire to reach a destination without getting lost or crashing into any objects. The goal of the navigation system of mobile robots is to move the robot to a named place in a known, unknown, or partially known environment.

The goal of the navigation process of mobile robots is to move the robot to a named place in a known, unknown or partially known environment. In most practical situations, the mobile robot can not take the most direct path from the start to the goal point. So , path planning techniques must be used in this situation, and the simplified kinds of planning mission involve going from the start point to the goal point while minimizing some cost such as time spent, chance of detection, or fuel consumption.

Several models have been applied for environment where the principle of navigation is applied to do path planning. For example, a grid model has been adopted by many researchers, where the robot environment is dividing into many line squares and indicated to the presence of an object or not in each square. On line encountered unknown obstacle are modelled by piece of “wall”, where each piece of “wall” is a straight-line and represented by the list of its two end points. This representation is consistent with the representation of known objects, while it also accommodates the fact the only partial information about an unknown obstacle can be obtained from sensing at a particular location.

In the figure 1 we present one example of navigation approach using a square cellule grid for the movement. One of the key issues in the design of an autonomous robot is navigation, for which, the navigation planning is one of the most vital aspect of an autonomous robot. Therefore, the space and how it is represented play a primary role in any problem solution in the domain of mobile robots because it is essential that the mobile robot has the ability to build and use models of its environment that enable it to understand the scene. The figure 2 illustrates one model of navigation where the polygonal model is used for the navigation.

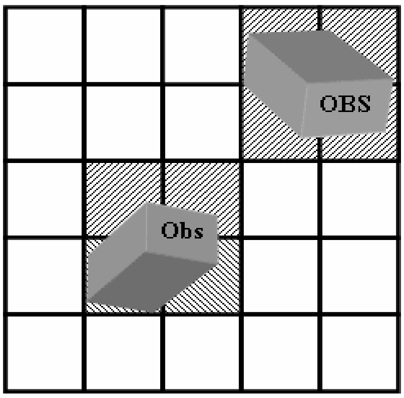

Fig. 1 an example of a square cellule grid 3D navigation

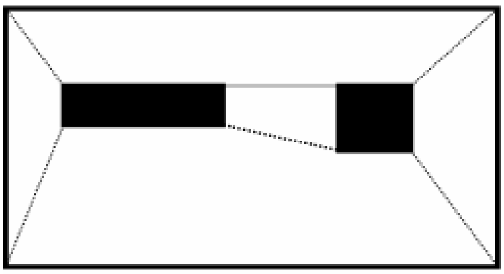

Fig. 2 an example of the polygonal model

A cellular model, in other hand, has been developed by many researchers where the world of navigation is decomposed into cellular areas, some of which include obstacles. More, the skeleton models for map representation in buildings have been used to understand the environment’s structure, avoid obstacles and to find a suitable path of navigation. These researches have been developed in order to find an efficient automated path strategy for mobile robots to work within the described environment where the robot moves. Often, a path is planned off-line for the robot to follow, which can lead the robot to its destination assuming that the environment is perfectly known and stationary and the robot can rack perfectly. Early path planners were such off-line planners or were only suitable for such off-line planning. However, the limitations of off-line planning led researchers to study on-line planning, which relies on knowledge acquired from sensing the local environment to handle unknown obstacles as the robot traverses the environment.

The major task for path-planning for single mobile robot is to search a collision –free path. The work in path planning has led into issues of map representation for a real world. Therefore, this problem considered as one of challenges in the field of mobile robots because of its direct effect for having a simple and computationally efficient path planning strategy. For path planning areas, it is sufficient for the robot to use a topological map that represents only the different areas without details such as office rooms. The possibility to use topological maps with different abstraction levels helps to save processing time. The static aspect of topological maps enables rather the creation of paths without information that is relevant at runtime. The created schedule, which is based on a topological map, holds nothing about objects which occupy the path. In that case it is not possible to perform the schedule. To get further actual information, the schedule should be enriched by the use of more up-to date plans like egocentric maps.

Topological maps can be used to solve abstract tasks, for example, to go and retrieve objects whose positions are not exactly known because the locations of the objects are often changed. Topological maps are graphs whose nodes represent static objects like rooms, and doors for example. The edges between the nodes are part’s relationships between the objects. In most practical situations, the mobile robot can not take the most direct path from start to the goal point. So, path finding techniques must be used in these situations, and the simplest kinds of planning mission involve going from the start point to the goal point while minimizing some cost such as time spent, chance of detection, etc. When the robot actually starts to travel along a planned path, it may find that there are obstacles along the path, hence the robot must avoid these obstacles and plans a new path to achieve the task of navigation. Several approaches for path planning exist for mobile robots, whose suitability depends on a particular problem in an application. For example, behavior-based reactive methods are good choice for robust collision avoidance [6, 7, 8, , 9, 10]. Path planning in spatial representation often requires the integration of several approaches. This can provide efficient, accurate, and consist navigation of a mobile robot. It is sufficient for the robot to use a topological map that represents only the areas of navigation (free areas, occupied areas of obstacles). It is essential the robot has the ability to build and uses models of its environment that enable it to understand the environment's structure. This is necessary to understand orders, plan and execute paths.

C. The proposed Genetic Motion planning

GAs are search algorithms based on the mechanics of natural genetics. A genetic algorithm for an optimization problem consists of two major components. First, GA maintains a population of individual corresponds to a candidate solution and the population is a collection of such potential solutions. In GA, an individual is commonly represented by a binary string the mapping between solutions and binary strings is called a "coding". GA has been theoretically and empirically proven to provide robust search capabilities in complex spaces offering a valid approach to problems requiring efficient and effective searching [5].

Before the GA search starts, candidates of solution are represented as binary bit strings and are prepared. This is called a population. A candidate is called a chromosome (in our case: the path is a "chromosome" and positions are the "genes"). Also, an evolution function, called fitness function, needs to be defined for a problem to be solved in order to evaluate chromosome. As fitness function, we should define distance for each chromosome to give an evaluation function. This evaluation is the goal of the GA search and goes as follows: two (02) chromosomes are chosen randomly from populations are mated and they go through operations like the crossover to yield better chromosomes for next generations. This is repeated until about "**k**" populations with new chromosomes. To determine execution of the GA, we must specify a stopping criterion, in our case; it could be determined, by a probabilistic function: as we have four chromosomes and we choose randomly two chromosomes, to combine and to compare one path with itself. The crossover is the comparison operator (see Fig.3). Therefore, after several generations of GA search (The problem of mutation, see Fig.4), relatively low fitness of chromosomes remain in a population and some of them are chosen as the solution of the problem (the most preferable path).

GAs are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the littlest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. An occasional new part is tried for good measure avoiding local minima. While randomized, GAs are no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved performance. In the figure 5 we present genetic algorithm chart.

D. Crossover and Mutation process

We have designed an operator that swaps two substrings of two genitors. However two children paths are generated which inherit some of the properties (substrings) of their two parents. Let two paths A_1 and A_2. Two points $S_1 \in A_1$ and $S_2 \in A_2$ are randomly selected, from which two new paths A_3 and A_4 from start to goat are generated by using the linking and concatenation operators (see the figure 6).We note that: the path is chromosome and the positions are genes. In theory any path of the S_{AB} can be to obtained by the crossover operator. However, due to the chosen probabilistic distribution in the calculus. The children have a great probability of lying in the region bounded by the parents that demonstrates the need of mutations to explore new regions. The selection is done according to a probability proportional to the performance (« fitness » in the classical literature). The selection is given by:

$$p = 2^{n-m}$$

Where **n** is the code number of paths designed to be candidates of selection of two (02) paths, **m** is the code number of all paths. The number two (02) because we have two candidates (two paths) each time to be treated. At the beginning, a population of paths is created by the mutation operator between A1 = A and A2= B . The size ***n*** of the population is a critical parameter of al the genetic algorithms : if the number of individuals is small, the region of the terrain explored at the beginning of the search are limited, and then the population iteratively generated for optimizing a performance index may tend to include paths neighboring a local minimum. On the other hand, a large population allows the generation of many individuals covering most of the terrain, and has a good chance to find all the optimal and near – optimal solution, but the population will also include less interesting solutions and furthermore the computing time may be high. The better chromosome which has less cost of path (the shortest path) yields after progenitor (several mutations and generation).The criteria of progenitor is stopped after (2^{n-m})of generations is equal to "**k**" . The fitness function for each path is the number of pixels belonging to this path. (In the literature the fitness function is the performance that evaluates and gives a meaning of each chromosome). For improving iteratively the performances of the individuals in the population, the best individuals are preferred to serve as parents in the next crossover operations.

E. Path planning

Assume that path planning is considered in a square terrain and a path between two locations is approximated with a sequence of adjacent cells in the grid corresponding to the terrain. The problem is to navigate from one cell to another without collisions. Three statements are assigned to the cell grid : *free, occupied,* or *unknown* otherwise. A cell is *free* if it is known to contain no obstacles, *occupied* if it is known to contain one or more obstacles. All other cells are marked *unknown*. In the grid, any cell that can be seen by these three states and ensure the visibility constraint in space navigation.

We denote that the configuration grid is a representation of the configuration space. In the configuration grid starting from any location to attend another one, cells are thus belonging to reachable or unreachable path. Note that the set of reachable cells is a subset of the set of free configuration cells, the set of unreachable cell is a subset of the set of occupied configuration cells. By selecting a goal that lies within reachable space, we ensure that it will not be in collision and it exists some "feasible path" such that the goal is reached in the environment. Having determined the reachability space, the algorithm works and operates on the reachability grid. This one specifies at the end the target area.

A grid of (x,y) dimension of free path is denoted by "***X***," an occupy grid of (x',y') is denoted by "***Y*** " . An obstacle is collection of hazardous cells in the "***Y***" ***grid*** **.**A path from start cell "C" to destination cell "D" that the detected color of "***X***" does not interest any detected color "***Y***". the path is said to be monotone of free cells "***X***" with respect to i-coordinates if no lines parallel to k-axis cross the j-axis (see figure 7).

The proposed algorithm here relies on number of cells and iterates, as follows :

Algorithm Path Planning
Begin
1) x by y grid, start cell a in the grid.;
2) Select all free cells area (without label of hazardous cell);
3) Select all hazardous occupied cells area and unknown cells;
4) ***IF*** the collection of free cells is continuous ***THEN***
BEGIN
WHILE (hazardous cells) are not done
BEGIN
A path from "one cell" to " another cell" is the collection of all free neighboring cells, Detect all neighboring on the same destination until the target is reached
ELSE
Change the direction and continue on another free continuous collection of cells;
END;
END;
END.

The robot starts from any position then it must move by sensing and avoiding the obstacles. The trajectory is designed in form of a grid-map, when it moves it must verify the adjacent case by avoiding the obstacle that can meet to reach the target. We use an algorithm containing the information about the target position, and the robot will move accordingly.

To determine the nature of space of navigation, and as we have illustrated before, cells are marked as free or occupied; otherwise unknown. We can therefore divide our search area into free and occupied area. Note that all free space cells represent the walkable space and unwalkable in occupied space. Each free cell is able of lying all the neighbor free cell within a certain distance "d". This distance "d" is usually set to a value greater than or equal to the size of cell. Note that the set of free cells is a subset of the of free cells, which is in region of of free occupancy cells. Thus, by selecting a goal that lies within free space, we ensure that the free sub-path will not be in collision with the environment, and that there exists some sub-paths to get the target. We use probabilistic concept to select the sub-paths to get the target, given by:

a1: randomly probability of initial source position..
a2: randomly probability in the admissible region of free space area (walkable area).
a3: randomly probability in the non admissible region of hazardous space area (unwalkable area).
a4: randomly probability of destination location in free space area (walkable area).

Every $\mathbf{a}_i$ belonging to occupied area (unwalkable area) is removed. Note that these probabilities $\mathbf{a}_i$ are done in order to trace without collision the free trajectory and not to be in unwalkable area stopped with inside obstacles.

Note, we determine the free resultant cells within free space to get a feasible path during navigation. For unwalkable space (occupied space) we just develop a procedure of avoiding danger. The figure 8 shows an example of walkable or unwalkable space.

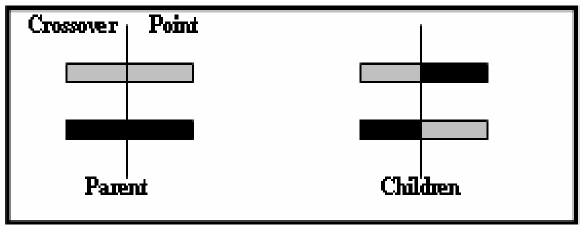

Fig. 3 an example of Crossover on singlepoint

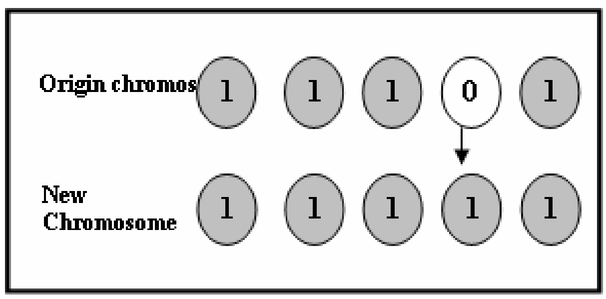

Fig. 4 an example of Mutation in the second bit

For unwalkable space, we compute the total size of free cells around danger (obstacle area). This total may be at least greater or equal than to the length of architecture of robot. This is ensure the safety to our robot to not be in collision with the obstacle, and that the path P has enough security SE to attend it target where it is given by P±SE (S is size of security). In principle, we generate a plan for reaching safety area for every neighboring danger area. The safety distance is generated to construct the safety area building to the navigation process, to be near without collision within this one.

F. Path planning based on GA

First, we need to choose a coding which maps a path from start cell "C" to destination cell "D" into a binary string. We can represent an arbitrary path by using a binary string of "variable length". To simplify the problem, we assume that a path from C to D is either i-monotone or j-monotone (but not necessarily both). Obviously, not all paths are monotone. The path can be represented by a column-wise (or row-wise) sequence of (x-1,y-1) pairs of direction and distance such that each pair corresponds to each column (or each row, respectively) . Thus, the path can encode into a binary whose length is proportional to j and fixed. The first bit B1 indicates that path is x-monotone if B1 =1, and is y-monotone if B1=0. A block of (n+1) bits represents distance and direction on each column or row, where (n+1) are pairs of distance and direction. In case of B1 =1 : the first 2 bits of a block denote the direction ;e.g., 00(vertical), 01 (diagonal), 10 (horizontal). In the case of B1=0; we denote 00 (vertical), 10 (diagonal) and 01 (horizontal) (see figure 9) the other bits of the block denote the distance is denoted by 0 if free, 1 if it is belonging to obstacle area. the population size is computed by $2^{(n+1)}$ bits. The likelihood of optimality which is the estimated probability of finding an optimal solution within g generations computed by : ($2^{(n)}/\ 2^{(g+1)}$). The mutation rate is a=0,1 crossover rate b=0,9 and win rate w=(1-a)/(1-b).this implies that GA can find an optimal solution with high probability if it is executed respectively and to a realized and b realized.

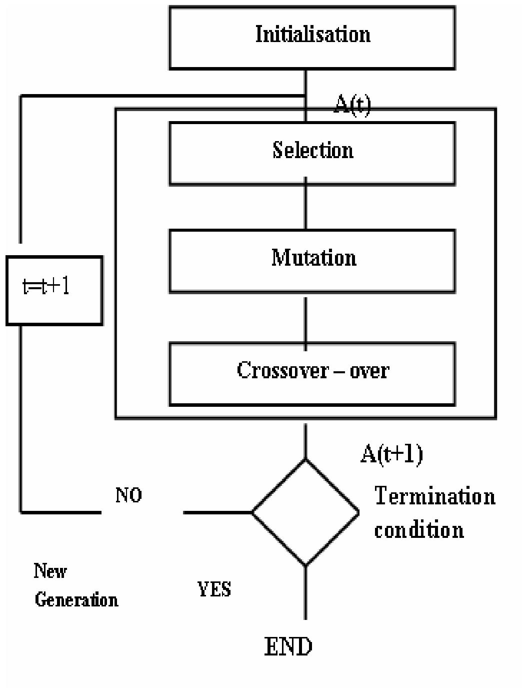

Fig. 5 Genetic Algorithm flowchart

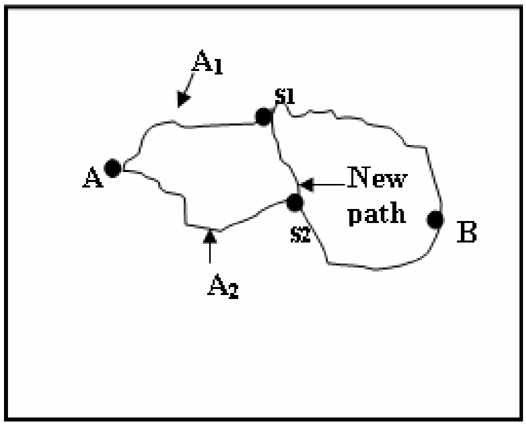

Fig. 6 crossover operation

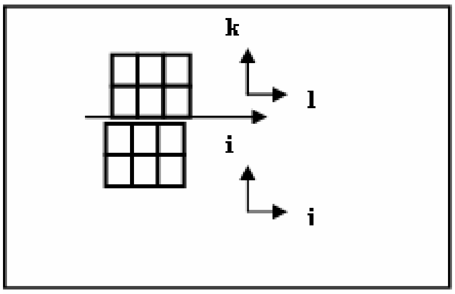

Fig. 7 an example of no intersecting in unknown environment

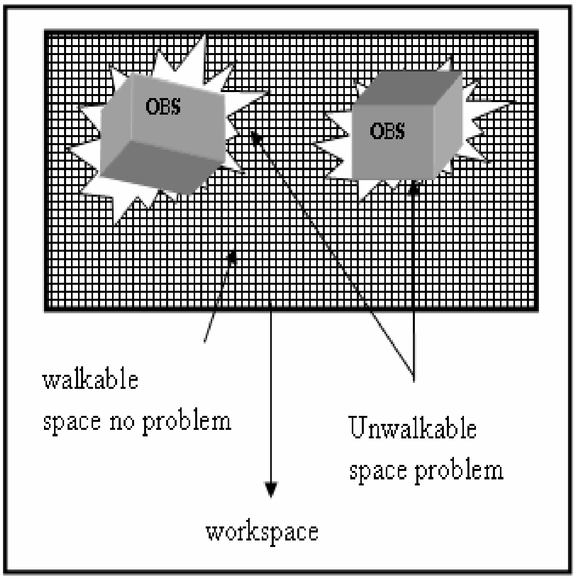

Fig. 8 an example of walkable space and walkable space

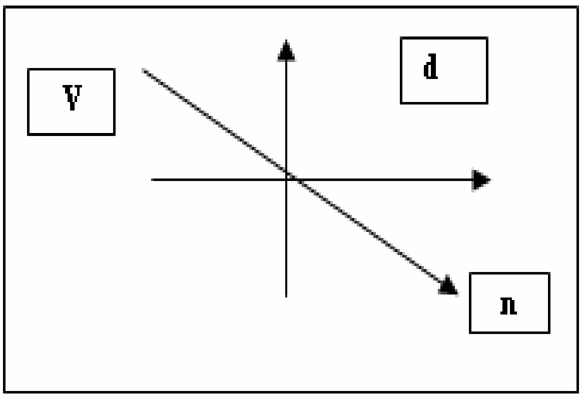

Fig. 9 an example of pairs of distance and direction

III. Simulation results

In order to evaluate, the average performance of our GA over various environments, we observed simulation of the GA for great number of environments. We can change the position of obstacles so we get other different environments. These environments were randomly generated .To find a new optimal path after insertion of deletion of an obstacle; we measure the number of generations of candidates. This way of work is very useful later if the substring is inherited to new generations by genetic operators. Every suboptimal is presented by its fitness function (number of detected pixels of free path) and is quite different from each other. Hence, a mobile robot detects unknown hazardous obstacle on the path. The generation of several paths gives at the end the best optimal path with low fitness function (e.g. the shortest path).

The simulation is done in different environments where the robot succeeds each time to reach its target without collisions. The walkable and unwalkable areas are detected in order to separate the free and occupied areas. The robot follows its destination for source to the target without collision in free area. Sensing, deciding, thinking and reacting; the robot across perfectly the connection between the source S and the target T searching its target as soon as possible the configuration walkable area build the feasible path .This connection can be a direct straight line, a set of sub paths which are straight or no (it depends on the configuration of electing and connecting between the source and the target).

The figure 11 and the figure 12 clarify more the principle and show how the robot succeeds to reach the goal without collisions according to the configuration of the selected environments. Taking a suitable action and reacting at the appropriate way, the robot finds its safe way without collisions in efficient manner.

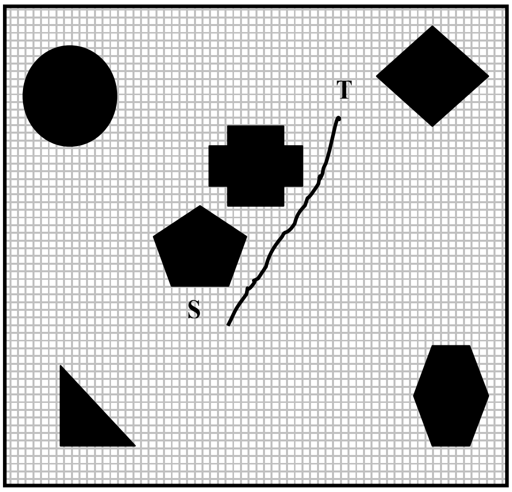

Fig. 10 an example of walkable space and walkable space

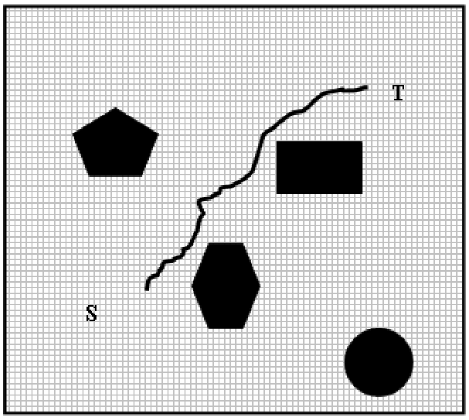

Fig. 11 an example of walkable space and walkable space

IV. CONCLUSION

The theory and practice of **AMR** are currently among the most intensively studied and promising areas in computer science and engineering which will certainly play a primary goal role in future. These theories and applications provide a source linking all fields in which intelligent control plays a dominant role. Cognition, perception, action, and learning are essential components of such-systems and their use is tending extensively towards challenging applications (service robots, micro-robots, bio-robots, guard robots, warehousing robots).

In this paper, we have presented a software implementation of navigation approach of an autonomous mobile robot in an unknown environment. The proposed approach can deal a wide number of environments. This system constitutes the knowledge bases of *GA approach* allowing recognizing situation of the target localization and obstacle avoidance, respectively. The proposed Algorithm has the advantage of being generic and can be changed at the user demand. Depending on the final performance requirements, *GA* can be implemented using software tools supported by the build map environment.

The robot starts from any position then it must move by sensing and avoiding the obstacles. The trajectory is designed in form of a grid-map, when it moves it must verify the adjacent case by avoiding the obstacle that can meet to reach the target. We use an algorithm containing the information about the target position, and the robot will move accordingly. . The proposed GA method can deal a wide number of environments and gives to our robot the autonomous decision of how to avoid obstacles and how to attend the target. More, the path planning procedure covers the environments structure and the propagate distances through free space from the source position. The results are very satisfactory to see the complexity of the principle and the extension versions of generation maps. This is certainly very useful and helpful for the application where the scene of the navigation is extended and randomly created "future extended environments".

REFERENCES

[1] D. Estrin, D. Culler, K. Pister, PERVASIVE Computing IEEE, 2002,pp. 59-69.

[2] T. Willeke, C. Kunz, I. Nourbakhsh, The Personal Rover Project : The comprehensive Design Of a domestic personal robot, Robotics and Autonomous Systems (4), Elsevier Science, 2003, pp.245-258.

[3] L. Moreno, E.A Puente, and M.A.Salichs, : World modeling and sensor data fusion in a non static environment : application to mobile robots, in *Proceeding International IFAC Conference Intelligent Components and Instruments for control Applications*, Malaga, Spain, 1992, pp.433-436.

[4] S. Florczyk, *Robot Vision Video-based Indoor Exploration with Autonomous and Mobile Robots*, WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, 2005.

[5] Gonzalez and R.Perez. : SLAVE : A genetic learning System Based on an Iterative Approach *IEEE, Transaction on Fuzzy systems*, Vol 7, N.2.

[6] O. Hachour and N. Mastorakis, IAV : A VHDL methodology for FPGA implementation, *WSEAS transaction on circuits and systems,* Issue5, Volume3,ISSN 1109-2734, pp.1091-1096.

[7] O. Hachour AND N. Mastorakis, FPGA implementation of navigation approach, *WSEAS international multiconference 4th WSEAS robotics, distance learning and intelligent communication systems (ICRODIC 2004),* in Rio de Janeiro Brazil, October 1-15 , 2004, pp2777.

[8] O. Hachour AND N. Mastorakis, Avoiding obstacles using FPGA –a new solution and application ,*5th WSEAS international conference on automation & information (ICAI 2004) , WSEAS transaction on systems* , issue9 ,volume 3 , Venice , italy15-17 , November 2004 , ISSN 1109-2777, pp2827-2834 .

[9] O. Hachour AND N. Mastorakis Behaviour of intelligent autonomous ROBOTIC IAR", *IASME transaction*, issue1, volume 1 ISSN 1790-031x WSEAS January 2004,pp 76-86.

[10] O. Hachour AND N. Mastorakism Intelligent Control and planning of IAR, 3rd WSEAS International Multiconfrence on System Science and

engineering, in Copacabana Rio De Janeiro, Brazil, October 12-15,2004.www.wseas.org.

[11] O.Hachour, The proposed Fuzzy Logic Navigation approach of Autonomous Mobile robots in unknown environments, International journal of mathematical models and methods in applied sciences, Issue 3, Volume 3, 2009 , pp 204-218.

[12] O.Hachour, the proposed hybrid intelligent system for path planning of Intelligent Autonomous Systems, International journal of mathematics and computers in simulation,Issue 3, Volume 3, 2009, Pages 133-145.

[13] O. Hachour, ,"path planning of Autonomous Mobile Robot", *International Journal of Systems Applications, Engineering & Development*, Issue4, vol.2, 2008, pp178-190.

[14] O.Hachour, "The Proposed Genetic FPGA Implementation For Path Planning of Autonomous Mobile Robot", *International Journal of Circuits , Systems and Signal Processing,* Issue 2, vol2 ,2008,pp151-167.