Mobile Robots Path Planning using Genetic Algorithms

Nouara Achour
LRPE Laboratory, Department of Automation
University of USTHB
Algiers, Algeria
nachour@usthb.dz

Mohamed Chaalal
LRPE Laboratory, Department of Automation
University of USTHB
Algiers, Algeria
mchaalal@usthb.dz

***Abstract*—In this article, we discuss path optimization to solve the problem of path planning for autonomous mobile robots. We consider the case of constrained environments where the robot is represented as a point. For that, we used an approach based on models of evolution; the genetic algorithms which are an interesting alternative to conventional methods of path planning. A population of paths is obtained firstly using a random distribution strategy. The performance of the proposed Genetic Algorithm based approach is tested on environments with increasing complexity. Through some results, we give a comparison between this strategy and a method based on Lazy A* search.**

Keywords-path planning; PRM; Genetic algorithms; robotics.

I. INTRODUCTION

The aim of motion planning is to find the allowable movements of a robot in a constrained environment (clearance of a robot from obstacles is low). In its simple version, it consists of finding a path free of collision from an initial configuration q_{init} to a final configuration q_{final}, it is a PSPACE-hard difficulty as shown by Reif [1].

This means that the complexity of the path planning problem increases exponentially with the dimension of the configuration space. Based on scientific research over the past twenty years, we found that there are two major families of algorithms that address this problem. One uses a deterministic approach while the second uses a probabilistic approach [2]. There are several deterministic search algorithms; we can cite Bellman, breadth first search, etc.

Recently, random sampling has emerged as a powerful technique for planning in large configuration spaces [3][4]. Random-sampling planners are classified into two categories: PRM (Probabilistic RoadMap) and RRT (Rapidly-Exploring Random Tree).

An other approach uses genetic algorithms which are Meta-heuristic search algorithms. Genetic algorithms (GA's) are search strategies based on models of evolution [5]. They have been shown to be able to solve hard problems in tractable time. Here, we need a solution space composed of a set of nodes randomly generated in C_{free} (free Configuration space).

When using GA's, we need a solution space composed of a set of nodes randomly generated in C_{free}. The algorithm execution is performed by the research of the best configurations between the initial and the final nodes with checking the optimization criterion which is here the distance.

The strength of this method is that it allows to explore and exploit the best solutions by two operators, which are selection and genetic reproduction.

Section II gives an overview of previous related works and exposes this work's motivations, Section III lays the mathematical description of the terms and concepts used in this article, Section IV describes briefly the PRM-based path planning, Section V details our approach in using genetic algorithms to plan optimized paths. In the last section, we report a series of actual runs.

II. RELATED WORK

GA's are considered as optimization algorithms for search in a space of potential solutions, so they are faced with the exploration-exploitation dilemma. The solution to the problem of planning by Genetic Algorithms is proposed for the first time by [5]. There are also other contributions by several researchers [6][7]. The common problem to all methods is how to choose the initial population. Most of these methods use a set of paths encoded in the chromosomes. The optimal path is calculated after several iterations. The necessary step in these algorithms is the determination of the fitness function (optimization criterion). [4] proposed to use a function of performance determined by the linear combination of distance, the smoothing angle and the robot position from the obstacles. Some papers have focused on dynamic environments [8] and others have tried to investigate in the synergism of respectively fuzzy logic - GA's [10] and neural networks-GA's [11].

In this article, we discuss path optimization to solve the problem of path planning for autonomous mobile robots. We have mainly focused on using genetic algorithms to calculate optimized paths, we have thus demonstrated their advantages and disadvantages, and for this we made a comparison with a widely used approach, PRM associated with A^* algorithm for finding optimal paths.

III. PRELIMINARIES

As the paper covers several notions relating to C-space (configuration space), we give some definitions for the most pertinent ones used here.

Definition 1: A **Workspace** $\mathcal{W}$ is a physical space represented by $\mathbb{R}^2$ for planar (2D) or $\mathbb{R}^3$ for 3D spaces.

Definition 2: An ***Obstacl***e $\mathcal{O}$ is a portion of $\mathcal{W}$ that is "permanently" occupied, represented by the *obstacle region* $\mathcal{O}, \mathcal{O} \subseteq \mathcal{W}$.

Definition 3: A robot $\mathcal{A}$ consists of one rigid body, or more "motion-constrained" rigid bodies, represented by the *robot region* $\mathcal{A}$ and is the set of all points in $\mathcal{W}$ that lie in $\mathcal{A}$, $\mathcal{A} \subseteq \mathcal{W}$.

Definition 4: A robot configuration q is a set of parameters that completely specify the position of robot $\mathcal{A}$ with respect to a fixed frame F, $\mathcal{A}(q)$ is the region of $\mathcal{W}$ occupied by $\mathcal{A}$ at configuration q. Each parameter of q corresponds to one degree of freedom of the robot.

Definition 5: The configuration space $\mathcal{C}$ for a robot $\mathcal{A}$ is the set of all configurations of robot $\mathcal{A}$ in $\mathcal{W}$ or:

$$\mathcal{C} = \{\forall\, q \mid \mathcal{A}(q) \subseteq \mathcal{W}\} \tag{1}$$

Definition 6: The configuration space obstacles $\mathcal{CO}$ is the mapping of the obstacles in the workspace to the configuration space, it is the set of all configuration of robot $\mathcal{A}$ at which the robot region is in contact or overlaps with obstacles regions or:

$$\mathcal{CO} = \{all\ q, i \mid \mathcal{A}(q) \cap \mathcal{O}_i \neq \emptyset\} \tag{2}$$

Definition 7: Free configuration space $\mathcal{C}_{free}$ is the set of configurations at which the robot is free from collision with the Workspace obstacles, or simply:

$$\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{CO} \tag{3}$$

Definition 8: A local-path lp is a continuous function of a parameter s which takes values in the interval : [0, 1] to $\mathcal{C}$ or: $lp: [0, 1] \rightarrow \mathcal{C} \mid lp(s)=q(s)$.
Furthermore, if the local-path is defined by its end configuration $lp_{(q0,q1)}$, then $lp(0)= q_0$, $lp(1)= q_1$.

Definition 9: The general definition of a path p is the same as that of a local-path, and when it is additionally defined by its start and end configurations $p_{(qs,\, qe)}$, then $p(0)= q_s$, $lp(1)= q_e$.

Given the discrete search used here, p will be mostly defined as an ordered set of k local-paths, or $p_{(lp1,\ lp2,..\ lpk)}$ where; $lp_i(1)= lp_{i+1}(0)$, $i=1,..k-1$, this results in a continuous path. A *sub-path*$_{(q1,q2)}$ of a path p is a continuous function from $[s_1, s_2] \rightarrow \mathcal{C}$ where $p(s_1) = q_1, p(s_2) = q_2$ and $\forall s \in [s_1, s_2]$: *sub-path*$_{(q1,q2)}$ $(s) = p(s)$. $0 \leq s_1 < s_2 \leq 1$.

IV. PLANNING WITH PRM

Probabilistic algorithms are based on the use of randomness for the construction of a graph capturing, in condensed form, the connectivity of the free space C_{free}, thereby precluding any explicit representation of the configuration space C. The basic idea of planners based on random sampling is to exploit the outstanding performance of collision detection algorithms that verify whether a given configuration is free or not. It should be noted that the completeness of these algorithms is rather low.

The principle used to sample the space C$_{free}$ is that of uniform random sampling, which can blanket the entire free region.

```
Simple Query PRM Algorithm (N, B,M,$\mathcal{V}$'s, $q_{ini}$ , $q_{fin}$)
1. Sample N configurations
2. R(V,E)←For every node n, assign the closest B visible
   nodes as visible from n.
3. Add $q_{ini}$ and $q_{fin}$ to R and query it using Lazy A* for
   path p, if successful return p.
4. Sample and connect M new nodes to R in "difficult"
   regions.
5. Query R using Lazy A* for path p, if successful return
   p else return Failure.
```

We may note that in step 1 and 2 we only construct a roadmap without actually checking the nodes in V nor the edges in E for collision. This will be performed during the query phase only when necessary by using a Lazy A*[9] with weighed L-infinity norm $d(q', q'')_{wL\infty}$ (the weights being ν_i's) as the cost of edge *e(q',q'')*. the difficult regions in step 4 are those collision free nodes that fail more often to connect to their neighbouring nodes compared to others, this could potentially indicate that these nodes are located in narrow regions [9][12] (Figure 1).

V. PATH PLANNING USING GENETIC ALGORITHMS

The algorithm is divided into two phases; first, we generate a set of configurations in free space and then in the second phase we use genetic algorithms as a metaheuristic search procedure to find the optimal path that leads robot from q_{ini} and q_{fin}. In order to sample the space C$_{free}$, we use the uniform random sampling, which can blanket the entire free region. Each time we generate a configuration, we check whether it is in collision or not. In our case, we generate an initial population of segments with a performance function based on the minimum distance (Figure 2).

To find the optimal path by the approach of genetic algorithms, we try to find the best segments of this path, such that the sum of their metric distance is minimal. The coding of these solutions is as follows: each chromosome contains two nodes, an initial and final node. These nodes are those generated in the first phase, so they are free of collision.

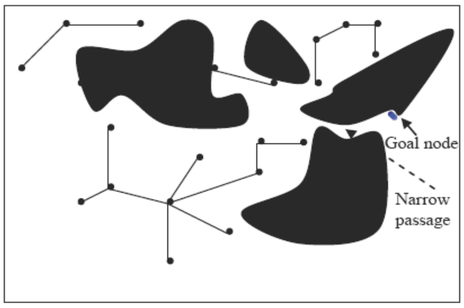

Figure 1. Roadmap built by PRM in a C-space with narrow regions.

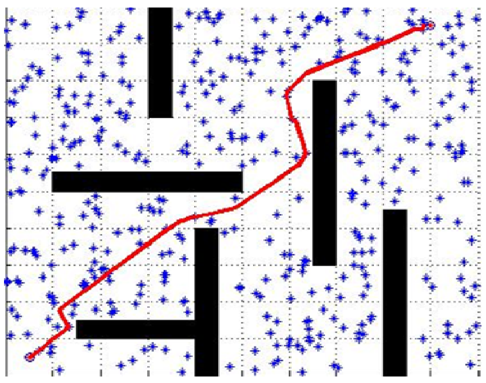

Figure 2. The blue stars represent all the configurations (nodes) generated randomly and the best path (in red) passing through the best nodes.

The algorithm runs in a finite number of iterations. At each iteration, we choose a new population that depends on the initial node of each segment (which represents the final node of the best segment in the previous iteration). The algorithm stops when we reach the goal configuration.
In order to be eligible in the initial population, a segment should not collide with obstacles.

Solutions encoding is performed by real numbers to store the topological structure of the solution space to the genotype space. With binary coding, there is a risk of losing information and the concept of geometric path.

After the initialization of the solution space, the second operation *crossover* is performed. This operation can produce new solutions from the initial population, which increases the chance of finding the best segment. The mathematical formulation of the operation of crossing depends on the nature of representation used to encode the solutions, for example, the crossing at one point can be applied to individuals with binary representation or real numbers. It consists of choosing a random point for the two chromosomes to exchange genetic material between two people around this point.

The crossing is performed only for the final node of each chromosome (Figure 3), and each time we check whether the node is in collision or not and visible to the initial node to avoid collision with obstacles.

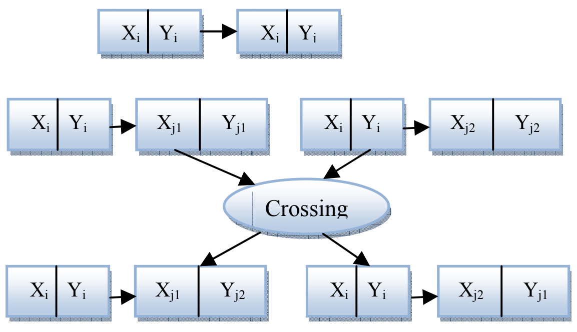

Figure 3. Crossing is performed only for the final node of each chromosome.

The probability of crossing and mutation are respectively equal to 1 and 0.

The fitness function is as follows:

$$f(s) = \left(\frac{1}{d_i} e^{\frac{-\theta_i}{d_i}}\right) - e^{\frac{-d_i}{\theta_i}} \tag{4}$$

$$d_i = \|n_i - n_{best}\|^2 + \left\|n_{fin} - n_i\right\|^2 \tag{5}$$

$$\theta_i = \tan^{-1}\frac{y_{n_{fin}} - y_{n_i}}{x_{n_{fin}} - x_{n_i}} \tag{6}$$

n_i and n_{best} being respectively the final and initial node of each segment, n_{fin} is the final node of the global path.

After evaluating each chromosome by the performance function, the best segment is the one which minimizes the distance of the path passing through these intermediate nodes. The best chromosome is determined after each generation, the final node of this chromosome becomes the starting point of the next segment.

The best node is the one which has the maximal fitness function. In our case, we have not considered the mutation operation since its random effect may produce undesirable results, there is a risk of losing the best individuals.

```
Genetic Algorithm (N, B,M,f, $q_{ini}$ , $q_{fin}$)
1. Encoding solutions into vectors $(x_i, y_i, x_j, y_j)$
2. Crossing (selection of crossover point randomly)
3. Evaluate all chromosomes by the fitness function $f$
4. Select the best segment whose $f$ is minimal $(x_i, y_i, x_{best}, y_{best})$
5. Add $(x_{best}, y_{best})$ to p
6. $x_i = x_{best}$, $y_i = y_{best}$ until $x_i = x_{fin}$, $y_i = y_{fin}$
```

VI. RESULTS

The two approaches of path planning PRM/Lazy A* and Genetic Algorithms were implemented in Matlab and have been tested on environments with simple structure then with more complexity.

In our case the mobile robot $\mathcal{A}$ navigates in a 2D environment consisting of walls and polygonal obstacles, it has three dof (two degrees for translation and one degree for rotation).

For both approaches, we considered the same initial and final configurations, namely: $q_{init} = [x_{init}, y_{init}, \theta_{init}]^t = [3, 2, \pi]^t$, and $q_{fin} = [x_{fin}, y_{fin}, \theta_{fin}]^t = [18, 19, \pi]^t$. The number of samples (M) and neighbors (B) are set respectively at 500 and 5.

Figures 4 (a, c, e, g) and (b, d, f, h) show the results obtained respectively for the first approach (PRM/Lazy A*) and second approach (GA’s).

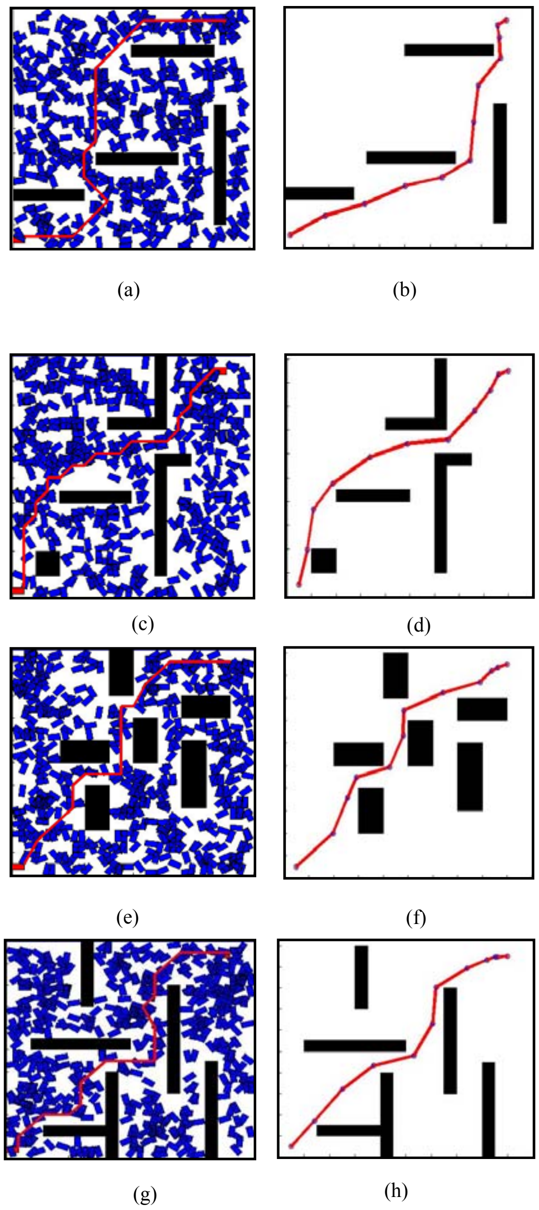

Figure 4. From top to bottom, the two paths obtained (on the right PRM/Lazy A* and GA's on the left), for environments 1, 2, 3 and 4.

Table I gives the computation time of a path by the two approaches in the four environments.

TABLE I.

	PRM/Lazy A*			GA's		
	M	N	T	M	N	T
Envt 1	500	500	118.90	500	34	20.02
Envt 2	500	1015	122.63	500	38	23.85
Envt 3	500	1634	168.92	500	41	21.96
Envt 4	500	2043	168.36	500	70	24.02

VII. DISCUSSION

In the typical results shown in Figure 2, we find that the final path is always optimized (in length) when using genetic algorithms. The time required to find the optimal path is about 24 seconds with a number of samples equal to 500.

If we increase the complexity of the environment, we obtain a path generally better optimized in distance and also in computation time. The number of iterations (70) is significantly lower than the one obtained from PRM (2043). Even if we increase the number of samples, the optimal path keeps the same performance.

The GA's approach does not require the construction of the graph of visibility, it minimizes the computation time of the path, it is very interesting for solving NP-hard problems.

Despite the good results, there remains a problem when meeting environments with narrow regions (Figure 4(c, d)) which makes very difficult the search for configurations that optimize the way, the algorithm falls into a local minimum. In this case, a probable solution is to generate additional configurations in these regions. But this is not always the right solution, because if we increase the number of configurations, there is a risk of not finding the way, despite the existence of feasible configurations in C_{free}. The failure to find a solution path is often due to the inefficiency of collision detection. By making several attempts, we can reach an acceptable solution. For example, for the environment 4, the path is found after 5 tests and 70 iterations.

VIII. CONCLUSION AND FUTURE WORKS

This paper investigates the application of GA's for solving the problem of path planning for mobile robots. We studied two approaches based on randomized algorithms, PRM and GA's. For the first approach, the results showed that we can find feasible paths for several types of environments, however, this does not qualify as 'robust' this approach, since we encounter cases where we can not find solutions, despite that there are configurations eligible to generate a path. In the second approach, which is based on genetic algorithms, a population of paths is obtained firstly using a random distribution strategy. The performance of the proposed Genetic Algorithm based approach is tested on environments with increasing complexity. The results obtained by this approach show the effectiveness of GA's, these algorithms can find the optimal path in a very short time and has the capacity to enrich the configuration space by a different set of eligible movements by using the crossover operator and selection.

As future works, we would like to extend this approach to multiple cooperating robots and mobile manipulators.

IX. REFERENCES

[1] J.H. Reif "Complexity of the mover's problem and generalizations," Proc. IEEE Transactions on Robotics and Automation, pp. 421–427, 1979.
[2] J. C. Latombe, Robot Motion Planning, Norwell, MA: Kluwer, 1991.
[3] E. Kavraki, P,Svestka, J-C. Latombe, and M.H. Overmars "Probabilistic Roadmap for path planning in high-dimensional

configuration spaces". IEEE Trans, Robot & Autom, 12(4),pp. 556–580, June 1996.

[4] L. Kavraki, M. Kolountzakis, and J.-C. Latombe. "Analysis of probabilistic roadmaps for path planning". In IEEE Trans. Robot. & Autom., volume 14, pp. 166–171, 1998.

[5] J.H. Holland, Adaptation in natural and artificial systems, Ann Arbor: University of Michigan Press, 1975.

[6] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. 1st ed. MA: Addison-Wesley, 1989.

[7] C. E. Thomas, M. A. C. Pacheco, M.M., and B.R.Vellasco, "Mobile Robot Path Planning Using Genetic Algorithms,". In foundations and tools for neural modeling, vol. 1606/1999, Springer Berlin/ Heidelberg, pp. 671–679, 1999.

[8] S. Yang , H. Cheng, and Fang Wang, "Genetic Algorithms With Immigrants and Memory Schemes for Dynamic Shortest Path Routing," IEEE Trans on Syst man, and cyb, vol. 40, n°1, pp. 52–63, 2010.

[9] R. Guernane, and N. Achour, "An Algorithm for Generating Safe and Execution-Optimized Paths" in Proceedings of the Int. Conference on Autonomic and Autonomous Systems, pp. 16–21, Spain, 2009.

[10] C. Fayad and P. Webb, "Development of a hybrid crisp-fuzzy logic algorithm optimised by genetic algorithms for path-planning of an autonomous mobile robot" in Journal of Intelligent & Fuzzy Systems, pp. 15-26, 2006.

[11] N. Noguchi and H. Terao "Path planning of an agricultural mobile robot by neural network and genetic algorithm" in Journal Computers and electronics in agriculture, 1997, vol. 18, n^{o}2-3, pp. 187–204.

[12] R. Bohlin, and L. E Kavraki, "Path Planning Using Lazy PRM" in Proceedings of the IEEE Int. Conference on Robot & Autom, pp. 521–528, IEEE Press, San Fransisco, 2000.