Number 2 Volume 11 June 2005 Journal of Engineering

PLANNING THE OPTIMUM PATH FOR A MOBILE ROBOT USING GENETIC ALGORITHM

Dr. Mayada F. Abdul-Halim
Wathiq Nagah Abdullah
University of Baghdad
College of Science
Department of Computer Science

ABSTRACT

One aspect of interest in robotics is planning the optimum path for a mobile robot or the optimum trajectory for link movements of a stationary robot in order to increase their efficiency. The objective of this paper is to identify the sequence of steps and processes needed for construction off-line path planning system using genetic algorithm (as we coined GPPS). In off-line path planning, the robot is given a map with the location of all obstacles in a given world. The goal is to construct the shortest possible path between a pre-defined start and goal positions and then follow this path without running into the obstacles. In addition to the three basic genetic operators, a new operator is proposed here which is coined as repair operator. Repair operator eliminates infeasible path segments and removes path points from nearby obstacles. However, the shortest possible path resulted from applying genetic operators and repair operator may contain overlapping and redundant segments. Hence, to eliminate these drawbacks, a new operator is proposed which is coined as enhancement operator. Eighty experiments are tested on GPPS with different cases. These cases are taken from different perspectives: number and distribution of obstacles, size of obstacles, and number of experiments per a workspace. All experiments with these different cases give, as possible, an acceptable feasible path.

الخلاصة

إحــدى المظاهر في ما يتعلق بأهمية الروبوت هي تخطيط أفضل طريق للروبوت المتحرك أو أفضل مسار مــن الحــركات المرتـــبطة ببعضها للروبوت الثابت وذلك لزيادة كفاءته. الهدف من هذا البحث هو تعريف مجموعة الخطوات و العمليات اللازمة لبناء نظام تخطيط طريق لخريطة معرفة مسبقا باستخدام الخوارزمية الجينــية (والذي اسميناه GPPS). في النظام الجيني المستخدم لتخطيط طريق تعطى للروبوت خريطة تحوي مواقــع كل الحواجز الموجودة في المساحة التي سيتحرك فيها الروبوت. هدفنا هو تكوين أقصر مسار ممكن - مــن نقطـــة الـــبداية إلــى نقطة النهاية المعطاة- باستخدام الخوارزمية الجينية و أتباع هذا الطريق بدون الاصـــطدام بالحواجـــز. إضــافة للعمليات الثلاث الأساسية في الخوارزمية الجينية، استخدمت عملية جديدة اسميناها (عملية الإصلاح). عملية الإصلاح تقوم بحذف أجزاء الطريق غير الملائمة و نقاط الطريق القريبة مـــن الحواجـــز. ومن هنا فإن اقصر طريق ممكن – الناتج من إجراء العمليات الجينية و عملية الإصلاح- ممكـــن أن يحـــوي علـــى أجزاء طريق متداخلة و متكررة. ولإزالة هذه العيوب اقترحنا عملية جديدة تدعى (عملية التحسين). ثمانون تجربة أجريت لاختبار (تخطيط الطريق باستخدام الخوارزمية الجينية) لحالات

مختلفة من التنفيذ. تلك الحالات أخذت من عدة أوجه مختلفة: عدد و توزيع الحواجز، حجم الحواجز، و عدد التجارب في مجال العمل الواحد. كل التجارب مع هذه الحالات المختلفة تعطي – قدر الإمكان طريق ملائم مقبول.

KEY WORDS
Genetic algorithms, path planning, off-line path planning, obstacle avoidance, Path traversal optimization, Time traversal optimization.

INTRODUCTION

In the last twenty years, mobile robots have become a subject of significant interest because they rise a multitude of challenging problems on one hand and because they are open to a broad range of possible applications on the other hand. The problem of optimal-path planning for autonomous mobile agents in two dimensions are concerned with large-scale or "high-level" path planning in which the start and goal points are specified, and the size and dimensions of the agent are negligible in comparison to the geometry of the path. And consistent with the term "large-scale", reasonably assuming complete knowledge of the features of the two-dimensional space including not only the locations and dimensions of obstacles for traversal, but traversal costs per unit distance for homogeneous regions within the two-dimension space or "terrain". The primary intended application of this research is off-road navigation by autonomous vehicles, that it also applies to planning military operations and constructing permanent linear features such as pipelines.

Finding an optimal path in two-dimensional workspace with obstacles is a well-known combinatorial optimization problem. Combinatorial optimization problems have been traditionally approached using exact techniques. Finding the optimal solution is ensured with these techniques but, unfortunately, they are seriously limited in their application due to the so–called combinatorial explosion.

The principle objective of this paper is to identify the sequences of steps and type of processes necessary to implement a path planning system using genetic algorithm that can generate optimal or near-optimal path according to a given workspace that does not make any prior assumptions about feasible knot points. *Genetic algorithm* is a class of stochastic search strategy that can be used to optimize the combinatorial explosion problem of path planning in robot navigation workspace. The rest of this paper is organized as follows: In the next section, the off-line path planning problem is given. Describing the design of the genetic-based path planning system (is coined GPPS) including the chromosome representation, the GA process and repair operator for auto-tuning the generated paths are presented in section 3. Finally, some concluding remarks are given in section 4.

OFF-LINE PATH PLANNING PROBLEM

The problem of path planning is to plot a continuous set of points connecting the initial position of the robot to its desired position. The requirement that the path planning must meet is finding quickly an optimal path due to some imposed criteria for the solution of the problem (which are referred to as path metrics). For example, path length, evaluation of clearance between the robot and obstacles, and others (Podsedkowski 1999)(Mchenry 1998).

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, stationary and the robot can track perfectly (Konar 2000). The main feature of the off-line approach is that a complete sequence of steps that solve a problem or reach a goal is determined before taking any action. Recovering from error and dealing with unexpected event is usually left as a subsequent execution stage (Kortenkamp, Bonasso, and Murphy, 1997).

Global planning includes three classical types of problems such as:

1- Obstacle avoidance

2- Path traversal optimization
3- Time traversal optimization.

The " Obstacle avoidance " type of problems deals with identification of obstacle free trajectories between the starting and the goal point. While the " Path traversal optimization " problem is concerned with identification of the paths having shortest distance between the starting and the goal point. Finally, the " Time traversal optimization " problem deals with searching a path between the starting and the goal point that requires the minimum time of traversal (Konar 2000).

GENETIC-BASED PATH PLANNING ALGORITHM

Since it was introduced by Holland in 1975, Genetic Algorithm (GA) approach has attracted the attention of researchers and been used to solve many difficult combinatorial optimization problems. The GA is an iterative procedure that maintains a number of candidate solutions, called population, over many simulated generations. Each chromosome is represented by a number of strings and undergoes genetic operation such as crossover, mutation and selection for improving the quality of the solution. At each iteration, called generation, each chromosome is evaluated and recombined with others on basis of its overall quality or fitness value in solving the problem. Recently, to improve the performance of the GA method, many variations of GA method are developed such as the concept of parallel GA (Cantu-Paz 1999).

This paper, presents a genetic-based path planning system (GPPS). The design of the genetic-based path planning system- GPPS- adopted in this paper is based on off-line path planning. **Fig. (1)** illustrates the general steps of the GPPS. The detailed description of the GPPS steps are presented in what follow.

```
Begin
    Get the map of the workspace.
    Get start and goal points.
    Form chromosomes by random generation of point
        coordinates until the population is fully filled with
        individuals.
    For all chromosomes in population, find fitness value.
    Repeat
        Repeat
            Select according to fitness two chromosomes from population.
            Apply crossover between selected chromosomes according to the
                crossover probability to generate a new offspring.
            Apply mutation on each point in the new offspring
                according to the mutation probability.
            Repair the new offspring.
            For all chromosomes in population, find fitness value.
        Until new population is generated
    Until specified number of generation is fulfilled or convergence
        state is reached.
    Enhance the resulted shortest path.
End.
```

Fig. (1) The GPPS algorithm

REPRESENTATION AND INITIALIZATION

In the genetic-based path planning, a path for a mobile robot is encoded based on an order of way points. Each robot has a starting point and a goal point in the workspace under the assumption that a robot passes each point at most once in a path.

It is important to define that the genetic algorithm generates a path consists of one or more straight-line segments (segment is a line between any two successive points), with the starting point, goal point, and (possibly) the end points of the segments that connect the starting point with goal point (Michalewicz, 1999). **Fig. (2)** depicts the chromosome representation of a possible path.

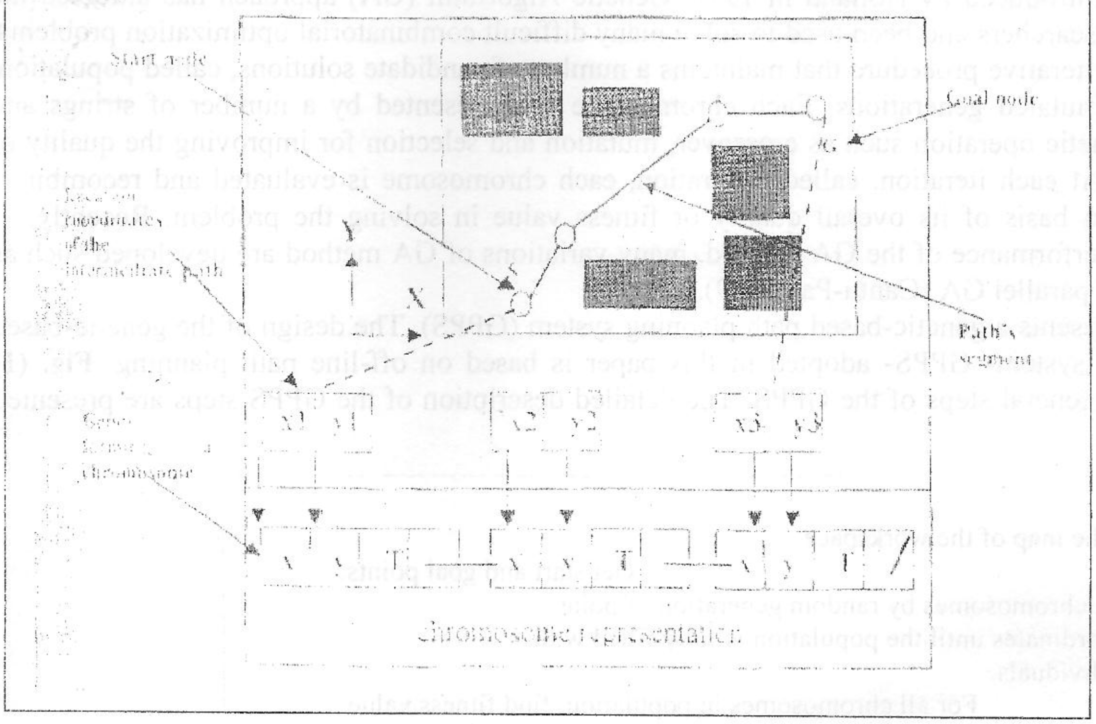

Fig. (2) GA chromosome representation of a path

Each point in the path chromosome consists of four fields. The first two fields x_i, y_i, $1 \le i \le n$, denote the coordinates of that point along the path and the third field b_i is a Boolean variable denotes whether the point is on an obstacle (i.e., whether the given point is feasible or not). If the point is feasible, its b_i value is set to TRUE, otherwise it is set to FALSE. The final field denotes a pointer pointing to the next point in the path to be visited (Konar 2000).

The length of the path chromosome, the number of the nodes represented in a path, is variable to deal with different situations of access to a goal point from a start point by different number of intermediate points. Intuitively, the length of a path chromosome should not be less than two. Hence, the length could be ranged from two to maximum value *Maxlength* that is determined previously at the initial genetic run.

An initial population of *pop_size* variable-length chromosomes is to be generated randomly. For each point of such a chromosome, the coordinates x and y are generated randomly (of course, the values of coordinates are restricted to be within the confine of the environment).

The value of the Boolean variable b_i, which represents the feasibility of its point, is determined. If the point is feasible, its b_i value is set to TRUE, otherwise it is set to FALSE. The method for checking the feasibility of a point is based on searching the dynamic array that contains the obstacles locations then check if the point does not fall on any of them. In this way, we will consider a discretized subset of all possible paths (with or without collision).

FITNESS EVALUATION

For each path chromosome in the current population, the fitness is computed. Because the problem deals with finding the shortest distance, it deals with paths that have minimum value. Calculation of the fitness involves finding the sum of the straight-line distance between any two successive points in the path chromosome. Before computing the fitness of a path, feasibility operation is checked. If the path points fall on obstacles, then the fitness value is divided by a large factor in order to give that infeasible path poor chance for selection in the next generation. The factor is selected proportionally to the size of the robot environment.

The formula of fitness function is:

$$fitness = \frac{1}{(\sum_{i=0}^{n-1} disance(point_i, point_{i+1})) * pen} \qquad (1)$$

Where n is the total number of path's points, while i is an index of the point in the path. The function *distance* returns the Euclidian distance between $point_i$ and. $point_{i+1}$. Finally, the penalty variable pen is one for all feasible solutions and ten for violator solutions (*i.e.* infeasible paths). The high penalty on violators (*i.e.* ten) increases the probability that violators will procreate. In GPPS, infeasible paths can be utilized as feasible paths using a new operator described next.

CHEKING THE FEASIBILITY OF A PATH

The process of checking the feasibility of a path done by evaluating the feasibility of each segment (a line between any two successive points) that constitutes the path. This requires an algorithm that takes the path points from starting point to its end point pixel by pixel. Digital Differential Analyzer (DDA) (Salmon and Slater, 1987) algorithm can be modified to check the path points if there is a point lies on an obstacle or not. We modify DDA algorithm so that instead of drawing a pixel, it will check the feasibility of that pixel.

GENETIC ALGORITHM PARAMETERS

To create a sequence of populations that hopefully will contain more and more good paths to our path planning as time goes on, the main genetic operators (selection, crossover, and mutation) are applied iteratively on each generated population. The following paragraphs discuss the main genetic operators that are used by GPPS.

As a reproduction, a binary tournament selection and elitist strategy are used. In binary tournament selection, two path individuals with their respective fitness values are paired randomly from old population to partake in a competition. The competition is held in a way so that diploid individual with higher fitness value wins the competition and advance a copy of it into the mating pool while the lesser fitness diploid individual is killed.

The elitist strategy is used by GPPS so that the best path in the current generation will be automatically survived to the next generation (Goldberg, 1989). Elitism makes the GA retain the best path at each generation. The best path can be lost if it is not selected to produce or if it is destroyed by crossover or mutation.

The crossover operator employs one-point and two-point crossovers, which are similar to the classical crossover widely used in genetic algorithms with crossover rate P_c. One point crossover used in the early generations of the genetic run to supply the population with more disruption paths. It recombines "good" parts of the paths present in both parents to produce hopefully better paths represented by the offspring (Michalewicz, 1999)(Gallardo, et. al. 1998). Two selected path chromosomes are cut in random positions and glued together: the first part of the first path with the second part of the second path, and the first part of the second path chromosome with the second part of the first path chromosome.

Fig. (3) and (4) show respectively examples of one point and two point crossovers applied on two selected path chromosomes. The crossover point(s) randomly occur at the third node in case of one-point crossover and at the second and the fourth nodes in case of two-point crossover.

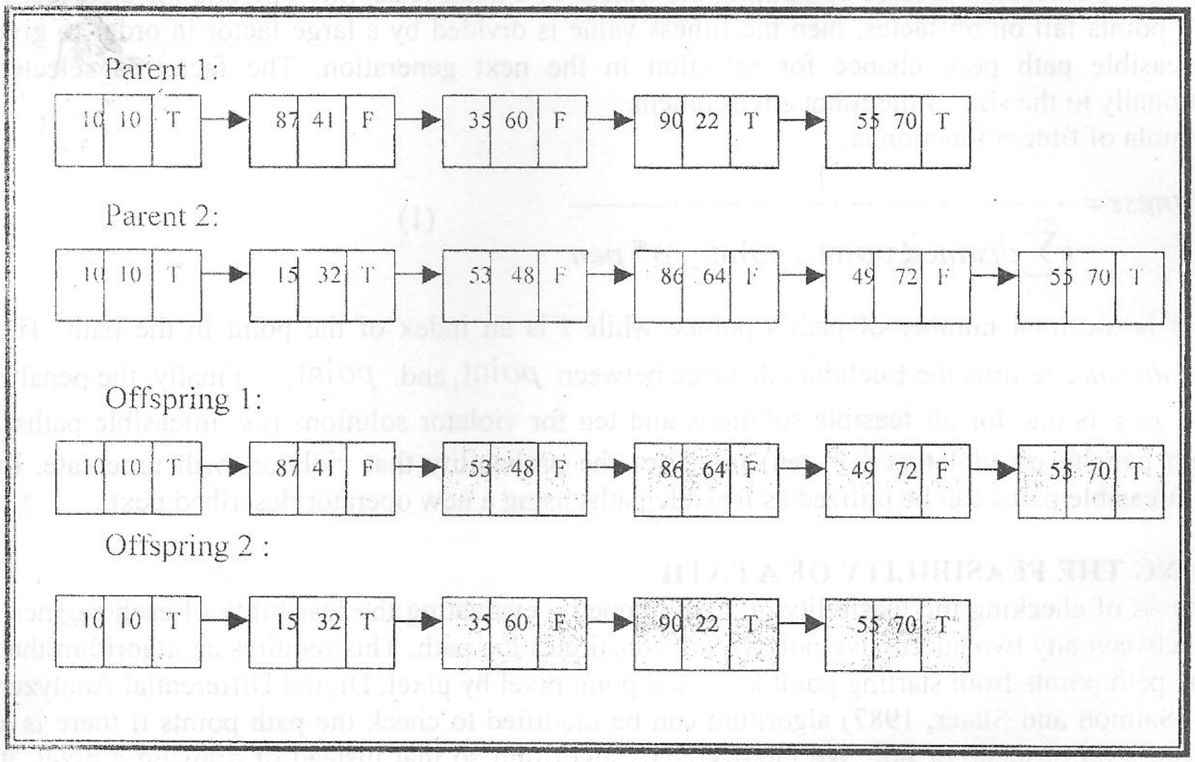

Fig. (3) Example of one point crossover

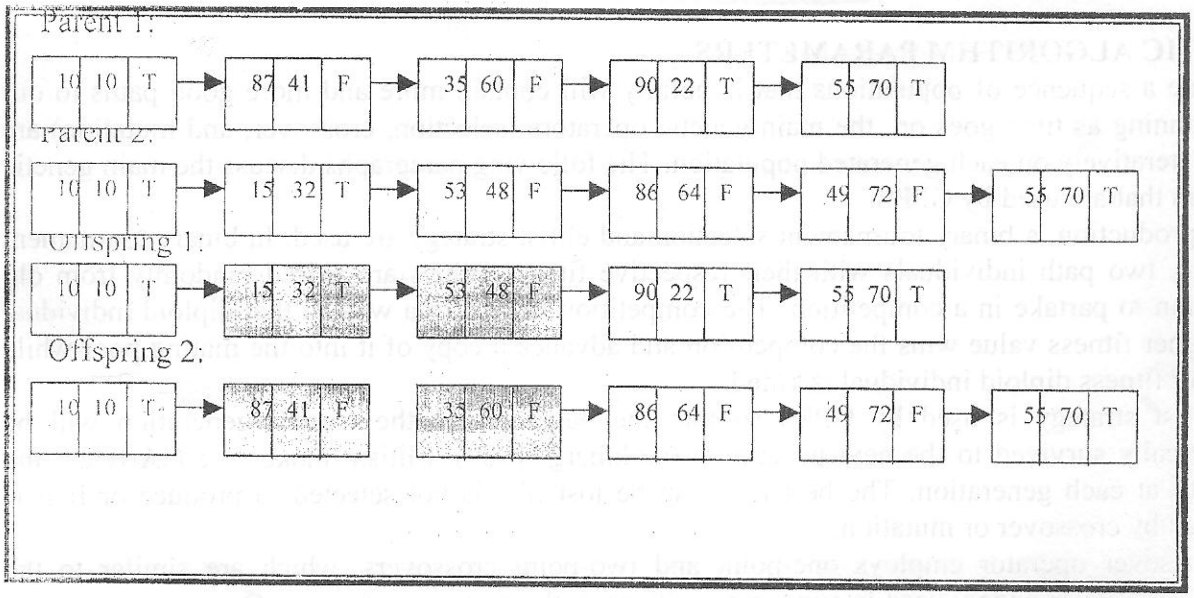

Fig. (4) Example of two point crossover

Next, mutation operation is applied with small probability p_m. The mutation is used to alter path points such that a constant value (d), generated at random, is added to or subtracted from the selected path's point. **Fig. (5)** shows how mutation operator is applied to a path example. In this figure, the mutated node is the third node and the value of d is four. The x coordinate value of the third node is replaced with $x+d$ value and the y coordinate value is replaced with y-d.

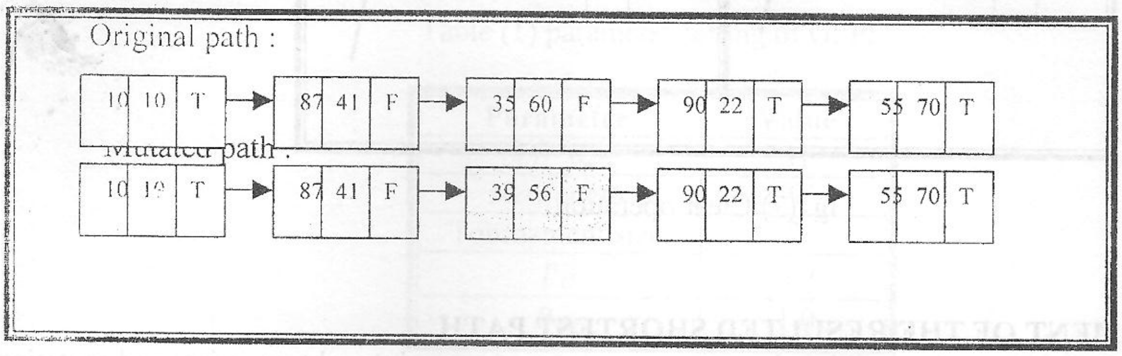

Fig. (5) Mutation operation

REPAIR OPERATOR

In addition to the previous operators, we proposed a new operator, which is coined as repair operator. This operator performs two operations: discard and clear. First discard operation eliminates each infeasible segment found in the chromosome path. Discarding infeasible segment is accomplished via moving end points of such segment outside the obstacle boundaries. **Fig. (6)** depicts an example.

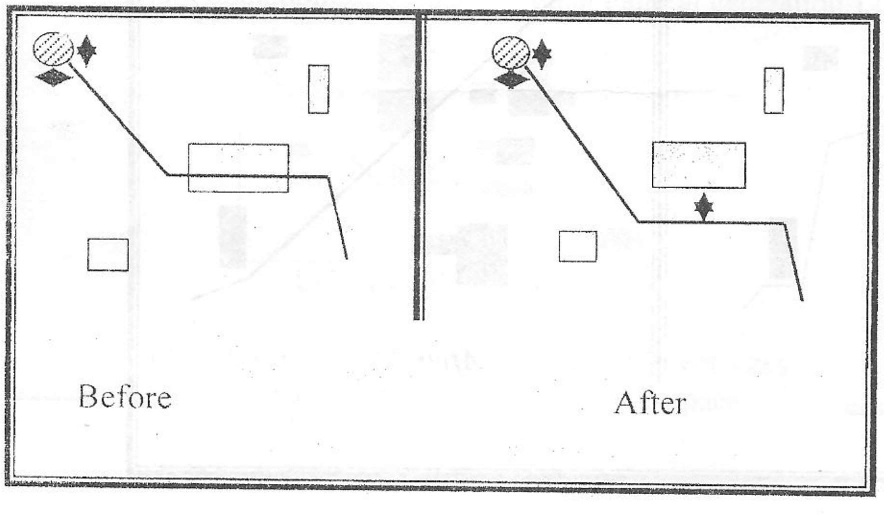

Fig. (6) Example of discard operation

There must be sufficient clearance between the obstacles for the easy movement of the robot through the path. Hence, the clear operation checks the clearance of a path points from nearby obstacles. The clear process is shown below. Consider **Fig. (7)** where the second path segment is near an obstacle.

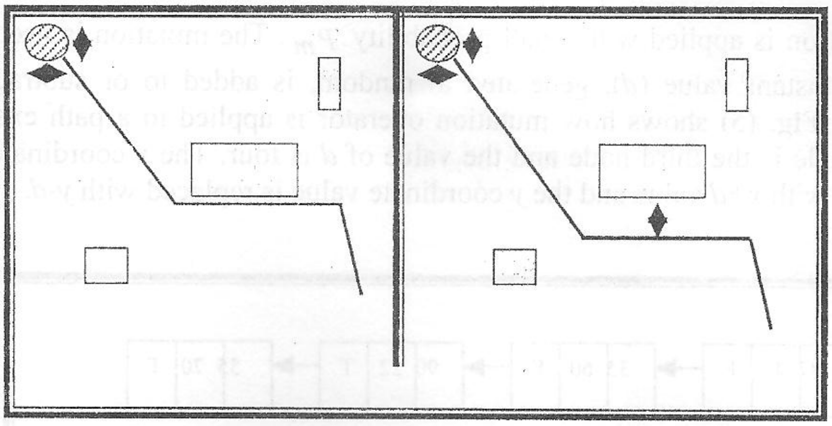

Fig. (7) Clear operation

ENHANCEMENT OF THE RESULTED SHORTEST PATH

The path generated from the traditional crossover operators (one-point and two-point) and mutation may contain overlapping and redundant segments. Hence, we propose an enhancement operator that eliminates any overlapping and redundant segments. This operation is accomplished by checking each two adjacent path segments. If the segment that connect the start point of the first path segment with the end point of the second path segment is found to be a feasible one, then eliminate these two adjacent segments and replace this new segment instead of them. **Fig. (8)** gives an example of enhancement operation.

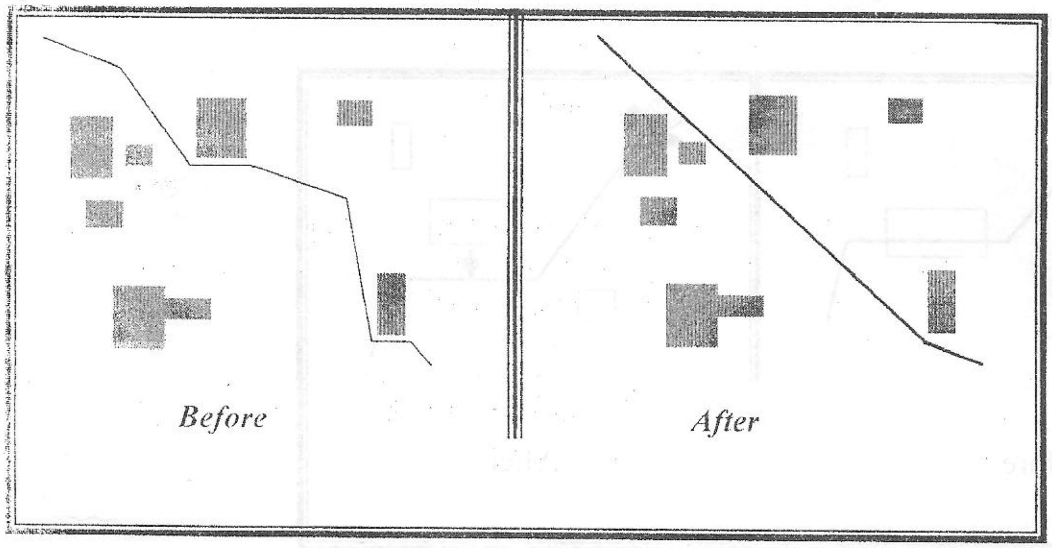

Fig. (8) The enhancement of a resulted path

TERMINATION CRITERIA

In most genetic-based path planning system the algorithm, terminate either when a feasible path is found or some fixed number of generations have elapsed. The convergence criteria and pre-specified number of generations are used in GPPS as termination criterion.

RESULTS AND CONCLUSIONS

GPPS was run on different cases. These cases are taken from different perspectives: number and distribution of obstacles, size of obstacles, and number of experiments per a workspace. All experiments with these different cases give, as possible, an acceptable feasible path. Eighty experiments are tested on GPPS. As a sample, **Figs. (9, 10), and (11)** give three different cases of these experiments with start point at (20,20) and the goal point at (580,420). **Table (1)** illustrates the parameter setting used by GPPS. For more results see (Abdullah 2004).

Table (1) parameter setting of GPPS

Parameter	Value
pop size	31
Maxlength	9
Tournament size	2
P_c	0.9
P_m	1.0

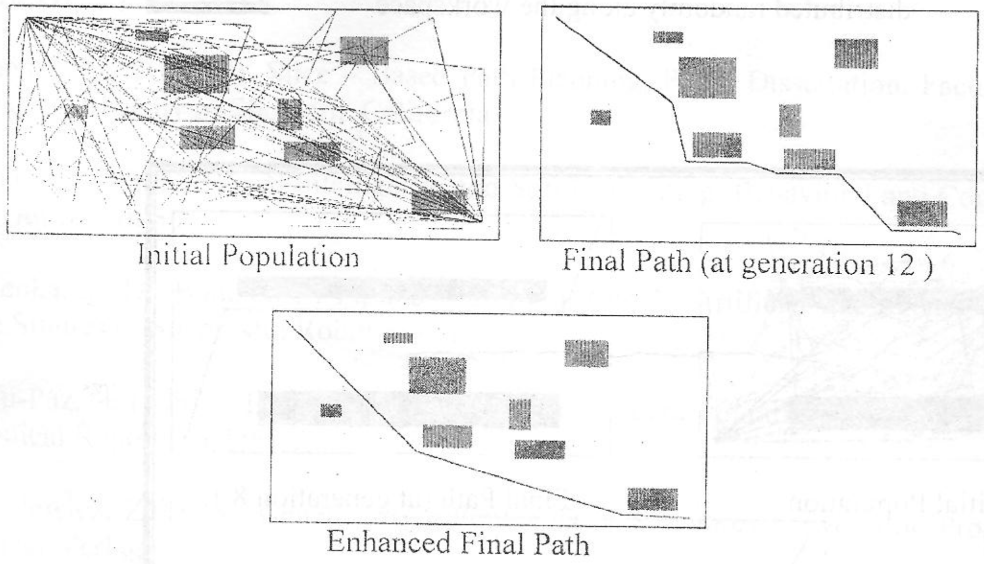

Fig. (9) Case1: eight obstacles with different sizes distributed randomly along the workspace

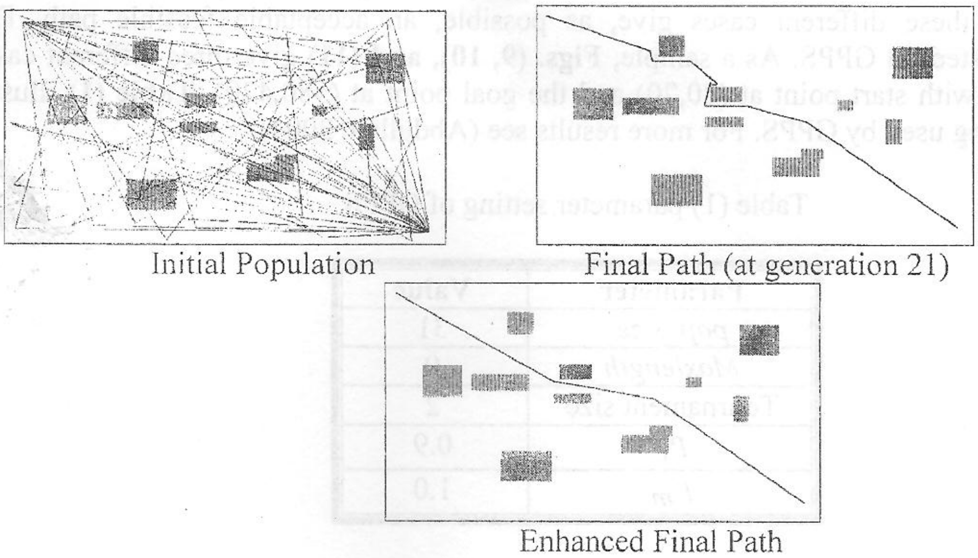

Fig. (10) Case2: eleven obstacles with different sizes distributed randomly along the workspace

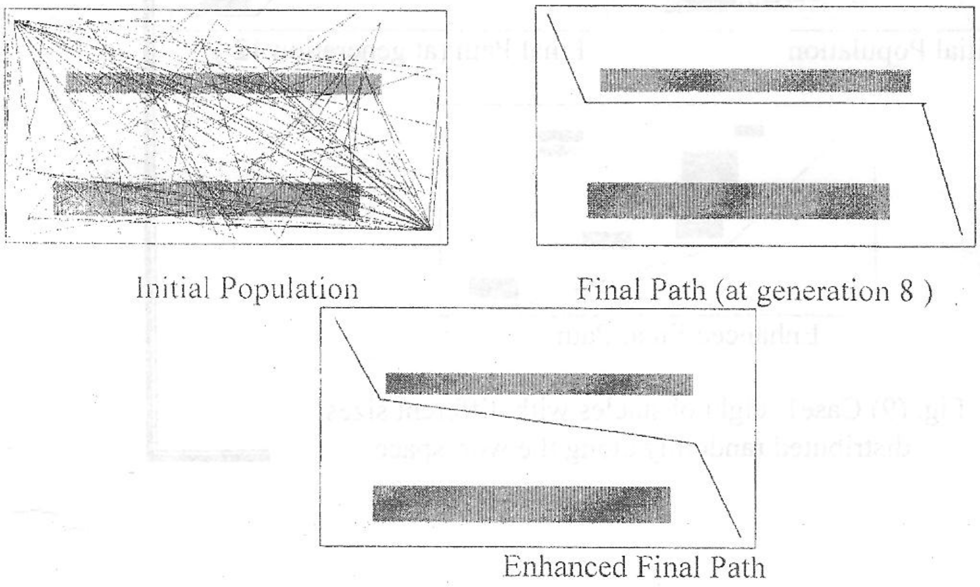

Fig. (11) Case3: two obstacles with bigsizes distributed randomly along the workspace

From the privious results we can conclude the following:

1- GPPS operates in the entire free space and does not make any prior assumptions about feasible knot points of a path.
2- GPPS introduces adaptive frequencies of the genetic crossover operator, as opposed to the constant genetic crossover used by the currently genetic path planning system. GPPS considers two forms of crossover, one-point and two-point. Although these two forms are very commonly used, they represent extremes. One-point crossover is the most disruptive of population, while two-point is the least disruptive.
3- The proposed repair operator aids the GPPS by providing paths that are clear and feasible.
4- The enhancement operation is applied for each resulted path from each run to improve the quality of the resulted path.
5- In GPPS, the resulted path from past run can be added to the initial population of the next run to take the advantage of the past experiences on that workspace.
6- The navigation process in GPPS terminates after a pre-specified number of generations or after stagnate state is reached. In these two cases, the GPPS find a feasible path within certain time period (i.e., within specified number of generations of the GPPS).

REFERENCES

Podsedkowski, L.(1999), Very Well Informed A* Searching Algorithm and its Application for Nonholonomic Mobile Robot Motion Planning. International Journal of Science & Technology, 10(2), 11(1,2), pp. 33-43.

Mchenry, M. C.(1998), Slice – Based Path Planning, Ph.D. Dissertation, Faculty of the Graduate School, University of Southern California.

Konar A.(2000), Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of The Human Brain.

Kortenkamp, D., Bonasso R. P., and Murphy, R.(1997), Artificial Intelligence and Mobile Robot: Case Studies of Successful Robot Systems.

Cantu-Paz, E.(1999), Designing Efficient and Accurat Parallel Genetic Algorithms, IlliGAL Technical Report No. 99017.

Michalewicz, Z.(1999), Genetic Algorithms + Data Structures = Evolution Programs. New York: Springer Verlag.

Salmon, R., Slater, M.(1987): Computer Graphics: Systems and Concepts. Reading, MA: Addison-Wesely.

Goldberg, D. E. (1989), Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley.

Gallardo, D., Colomina, O., Flrez, F., and Rizo, R.(1998), A Genetic Algorithm for Robust Motion Planning.

Abdullah, W. N. (2003), Genetic-based Path-Planning System. M.Sc. Dissertaition, Department of Computer Science, University of Baghdad.