A Multi-Operator Based Simulated Annealing Approach For Robot Navigation in Uncertain Environments

Hui Miao Hui.Miao@microchip.com
Microchip Australia Design Centre
Microchip Technology Inc.
Brisbane, 4108, Australia

Abstract

Optimization methods such as simulated annealing (SA) and genetic algorithm (GA) are used for solving optimization problems. However, the computational processing time is crucial for the real-time applications such as mobile robots. A multi-operator based SA approach incorporating with additional four mathematical operators that can find the optimal path for robots in dynamic environments is proposed in this paper. It requires less computation times while giving better trade-offs among simplicity, far-field accuracy, and computational cost. The contributions of the work include the implementing of the simulated annealing algorithm for robot path planning in dynamic environments, and the enhanced new path planner for improving the efficiency of the path planning algorithm. The simulation results are compared with the previous published classic SA approach and the GA approach. The multi-operator based SA (MSA) approach is demonstrated through case studies not only to be effective in obtaining the optimal solution but also to be more efficient in both off-line and on-line processing for robot dynamic path planning.

Keywords: Optimization, MSA, SA, GA, Dynamic Environments

1. INTRODUCTION

Mobile robots have been widely used in many industrial areas such as aerospace systems, nuclear applications, and mining equipment. How to find an absolute safe path in a dangerous environment for a mobile robot is one of the most important aspects in robot navigation. The main goal of the robot path planning is to search a safe path for a mobile robot, to make the robot move from the start point to the destination point without collision with obstacles. Also, the path is often required to be optimal in order to reduce processing times, communication delay, and energy consumption. According to [18], existing methods for robot path planning could be classified in different ways. Depending on the environment where the robot is located in, they can be categorised into the following two types:

1) Path planning in a static environment with static obstacles in the map; and
2) Path planning in a dynamic environment with both static and dynamic obstacles in the map.

Each of these two types could be further divided into two sub-groups depending on how much the robot knows about the entire information of the surrounding environment:

- Path planning in a clearly known environment, in which the robot already knows the location of the obstacles before it starts to move. Because the environment is fully known, the path for the robot could be the globally optimised result.
- Path planning in a partially known or uncertain environment, in which the robot probes the environment using sensors to acquire the information about the location, shape, and size of the obstacles, and then uses the information for local path planning.

This paper proposes a multi-operator simulated annealing (MSA) approach incorporating with multiple mathematical operators for robot path planning in dynamic environments. In this work, the MSA approach will be shown that the approach gives much improved performance than existing approaches for dynamic path planning results that have been presented in ICARCV'2008 conference [18].

2. RELATED WORKS AND MOTIVATIONS

2.1 Related Works on Dynamic Path Planning

Given the entire information of the environment in which a robot is, the globally optimal or near-optimal path could be found by using optimisation algorithms, e.g., the GA [1], [3], [20] and Fuzzy logic [2]. The A* algorithm [4] is developed to help a robot to find the optimal path in grid decomposed static maps. The A* algorithm uses the heuristic based Dijkstra algorithm to obtain the optimal result for the robot. The drawback of this method is the use of a uniformed grids representation, which demands allocation of large amounts of memory, even for those regions that may never be traversed or may not contain any obstacles, implying that the efficiency of the method may be low.

Evolved from the improved A* shortest-path algorithm, an improved algorithm is developed by Hu and Gu [5] to solve the problem of optimum route planning in vehicle navigation systems. It is based on the standard GA and the lambda-interchange local search method. However, in many practical applications such as those described in [6] and [7], it is difficult for a robot to get the full information of the surrounding environment at any one time because the status and the movement of the obstacles in the environment change all the time in the map. A one time global path planning made by the robot may become an infeasible solution due to the changes in the environment.

For dynamic environments with moving obstacles, limited work has been reported on optimal path planning for mobile robots. Chakravorty and Junkins [8] have introduced a methodology for intelligent path planning in uncertain environments with vision-like sensors. Recently, Wang, Sillitoe and Mulvaney [9] have presented a GA planner to determine optimal or near-optimal path solutions for mobile robots in dynamic environments. The GA based approach is shown to be a promising tool for the path planning problem of mobile robots in dynamic environments with moving obstacles.

Zhang, L¨ u, and Song [10] have developed an artificial potential field algorithm for dynamic path planning for soccer robots in dynamic environments where both the target and obstacles are moving. The D* method [11], the dynamic A*, is a typical method for path planning in dynamic and unknown environments. It plans optimal traverses in real-time by incrementally repairing the paths to the robot's state when new environment information becomes known to the robot, making it possible to reduce the computational cost significantly. When the robot gathers new information about the environment, it re-plans new paths based on the new information.

To further enhance the performance of the D* algorithm, improvements have been made to the D* algorithm. Representative works include the framed-quadatree D* method [12], the field D* mehtod [13], and others such as [14], [15]. The framed-quadatree D* method uses the quadatree structure to represent the dynamic environments. In order to minimise the search space, different dimensions of grids are used in the quadatree structure and border cells are added for connecting the grids. The field D* method [13] employs an interpolation-based planning and re-planning algorithm to generate smooth paths through non-uniform cost grids. It uses linear interpolation during the planning to calculate accurate path cost estimates for arbitrary positions within each grid cell and to produce paths with a continuous range of headings. It can produce a smooth optimal path for a robot to overcome the sub-optimal problem appearing in other non-uniformed-girds methods. Willms and Yang [16] proposed a dynamic system for real-time robot path planning. Recently, they further developed a grid-based algorithm for real-time robot path planning via a distance- propagating dynamic system [17].

2.2 **Motivations of This Work**

SA based method [18] is published previously proving that SA method can offer better performance on both path length and processing time than the GA method [9]. The performance of the previous proposed SA [18] still deteriorates significantly as the problem size increases. We believe that the performance of the SA approach in [18] can be further improved because of the operator that generates new solutions is relatively simple in [18]. Only two operators have been used, which switching or deleting some bits of the result to generate a new solution. This means that the possibility of jumping out of the local minimum is small. Therefore, more mathematical operators (switching, deleting, mutating and repairing) are implemented with the existing SA approach to improve the efficiency of the SA approach.

In this work, an SA approach incorporating multiple mathematical operators is developed for robot path planning in dynamic environments. It will be shown that the approach gives much improved performance than existing approaches for dynamic path planning. Some preliminary results of this work have been presented in ICARCV'2008 conference [18].

3. Multi-Operator Simulated Annealing Approach

3.1 Dynamic Environments

As in previous work [18]. The obstacles in the environments are represented by bounded polygons. Thus, the movement and trajectory of a dynamic obstacle are constituted by a series of polygons with their positions being updated along with the time. The vertices of the obstacles form the search space of our path planning algorithm. Following assumptions are made in this work: The movement and trajectory of moving obstacles in the environment are unknown to the robot; The motion parameters, such as speed and direction, of the dynamic obstacles can be sensed by the robot if the obstacle is in the range of the robot sensors; The robot could change its moving direction at any time when necessary; As in [9], all obstacles in the map are enlarged by a fixed value so that the robot could approach the obstacles without collision; and the dimension of the robot is neglected, and consequently the robot is regarded as a single point.

FIGURE 1 shows a dynamic environment, where the black polygons represent the static obstacles and the hollow polygons are moving obstacles. All the obstacles are enlarged by some values through creating additional margins. The vertices of the enlarged polygons are the search space for robot path planning.

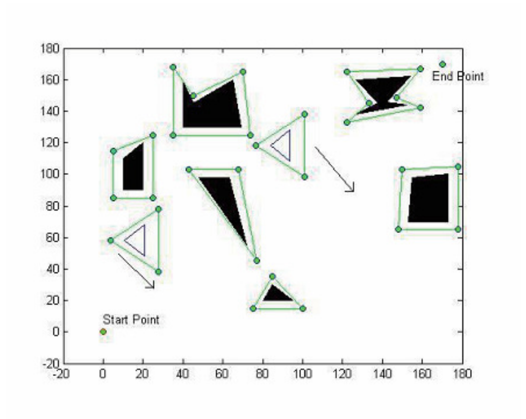


FIGURE 1: A Dynamic Environment with Static and Moving Obstacles

3.2 Architecture of the Multi-Operator SA Approach

The SA algorithm begins with the off-line path computation, in which the locations of the vertices of the static obstacles are fully known to the robot. Once the off-line computation is complete, the robot can start to travel through the stationary obstacles. When a moving obstacle enters the detection range of the robot sensors, the obstacle together with its moving speed and direction will be detected. Then, the robot calculates the possibility of clashing of the robot with the moving obstacle. The calculation result determines what the robot does next. If the moving obstacle will not hit the robot, the robot will keep travelling using the current path. Otherwise, if the robot will likely collide with the obstacle with the current movement, the SA algorithm will be triggered to find an alternative path from the current location of the robot to the destination.

As shown in FIGURE 2, a mathematical model is established for calculating the possibility of clashing of the with a moving obstacle. It is seen from FIGURE. 2 that the first crossing point of the current robot path and the predicted trajectory of the moving obstacle can be calculated. Then, the time t required for the robot to travel from its current location to the first cross point can be derived; and the location and consequently the corresponding exclusion area of the moving obstacle can also be estimated after the obstacle moves forward for the same amount of time t. If the robot path segment from the first crossing point to either of the two directions of the path crosses the edges of the moving obstacle odd times, then a collision will likely occur between the robot and the moving obstacle; otherwise, a collision will unlikely happen.

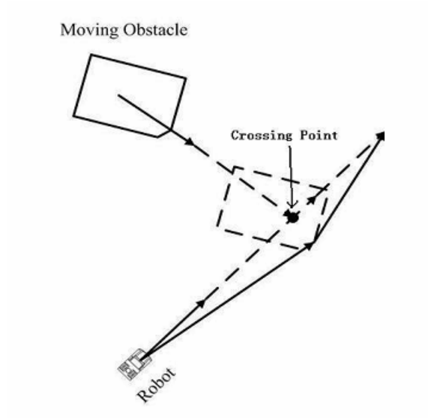


FIGURE 2: Calculation of the collision possibility (dotted lines: the original trajectories of the robot and dynamic obstacle; solid line: alternative robot path to avoid collision with the obstacle).

3.3 MSA Architecture

As in [18], a feasible path solution is expressed by a series of vertices linking the start point through to the end point. Each vertex of the obstacles has its series number; and thus a path is represented by a sequence of vertex numbers. Therefore, a feasible path solution X is described as:

$$X = \{V_{start}, V_{start+1}, V_{start+2}, \cdots, V_{end-1}, V_{end}\} \quad (1)$$

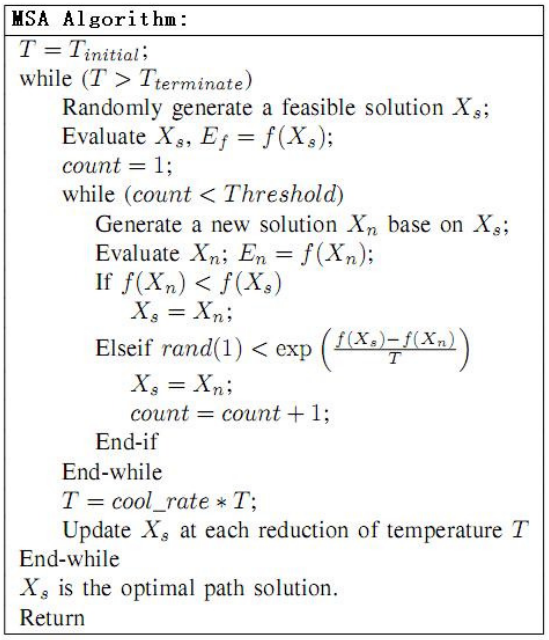
Where Vi means the ith vertex.

Traditionally, the length of the path, E_f, is used as a criterion to quantify the quality of the path solution derived from a path planning algorithm: the shorter the path, the better the solution. The evaluation function E_f is given by:

$$E_f = \sum_{i=start}^{i=end-1} D(V_i, V_{i+1}) \quad (2)$$

Where $D(V_i, V_{i+1})$ is the direct distance from V_i to V_{i+1}.

FIGURE 3 shows the top-level algorithm structure and pseudo-code of the multi-operator SA for dynamic path planning.



```
MSA Algorithm:
$T = T_{initial}$;
while ($T > T_{terminate}$)
    Randomly generate a feasible solution $X_s$;
    Evaluate $X_s$, $E_f = f(X_s)$;
    $count = 1$;
    while ($count < Threshold$)
        Generate a new solution $X_n$ base on $X_s$;
        Evaluate $X_n$; $E_n = f(X_n)$;
        If $f(X_n) < f(X_s)$
            $X_s = X_n$;
        Elseif $rand(1) < \exp\left(\frac{f(X_s) - f(X_n)}{T}\right)$
            $X_s = X_n$;
            $count = count + 1$;
        End-if
    End-while
    $T = cool\_rate * T$;
    Update $X_s$ at each reduction of temperature $T$
End-while
$X_s$ is the optimal path solution.
Return
```

FIGURE 3: Pseudo-code of the MSA Algorithm

3.4 Random Multi-Operator Path Planner

The MSA procedure is demonstrated in FIGURE 3. Different from the simple path planners that were previously used in [18], more complicated random path planners are developed in this work. Deleting, switching, mutation and repairing operators are used in the planner; and the planner randomly chooses one operator to generate a new path Xn from the initial path Xs. FIGURE 4 shows that how the operators randomly generate a new path Xn from the initial path Xs.

Similar to the procedure for initial path selection, the feasibility of each path segment generated by the operators is tested. This is to ensure that the path segment does not intersect with any edges in the map. When the length of the path is chosen as the evaluation criterion, randomly deleting vertices could help improve the performance of the path solution. Therefore, the possibility of choosing the deleting operator is set to be higher than other operator. As will be seen later in case studies, the possibility of choosing the deleting operator is set to be 0.70 in this work.

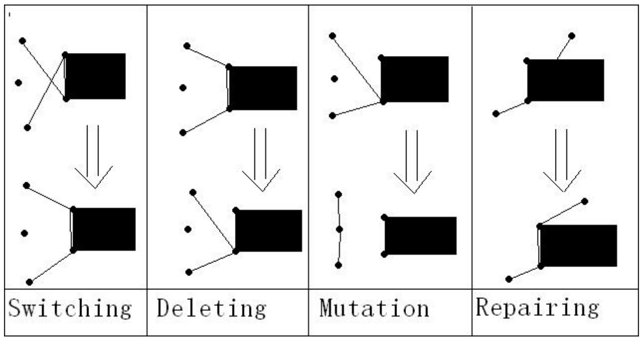


FIGURE 4: Multi-Operator Planner

After a new path solution is generated by using the operators, it is evaluated using the evaluation criterion, i.e., the length of the path. It is accepted if it is better than the previous one. It may also be accepted in a certain probability defined by the current temperature even if it is not better than the previous one.

3.5 Online Path Planning

While a robot uses the route generated by off-line planning to travel through static obstacles, the on-line path planner is triggered automatically to calculate an alternative path when a dynamic obstacle is detected. As no particular brand or configuration of the sensors is specified, the sensing range of the robot sensors is set to be a fixed value. If the distance between the robot and every vertex of the moving obstacle is shorter than the fixed value, the moving obstacle enters the sensing range of the robot and thus can be detected by the robot sensors.

It has assumed that robot sensors can monitor the shape and trajectory of a moving obstacle as well as the moving speed and direction. After acquiring the moving information of the moving obstacle, the robot could infer the possibility of collision with the moving obstacle. When it is inferred that the robot will collide with the moving obstacle, the SA based dynamic path planning algorithm will be triggered for finding an alternative path for the robot to travel from its current location to the destination. The search space, the current status of the robot, and location and moving information of the moving obstacle will be updated to enable the dynamic path planning.

4. Experiment Results

4.1 Simulation Environments and Parameters

Our case studies are carried in Matlab [19] under Windows XP on a computer with 2.8GHz Pentium Core 2 Duo CPU and 2GB memory. Four dynamic environments are designed to test the performance of the dynamic path planning approaches. They contain both static and dynamic obstacles, as shown in Table 1. For each of these four environments, the number of the vertices of the static obstacles is also tabulated in TABLE 1; it is used for offline path planning before the robot starts to travel.

Environment	Static Obstacles	Dynamic Obstacles	No. of Static Vertices
1	3	2	10
2	6	2	25
3	9	4	53
4	14	6	82

TABLE 1: Four Environments

The dynamic obstacles in all the environments have random shapes. The first two environments simulate simple scenarios where the dynamic obstacles appear simultaneously and simply move forward in the same direction. With more static and dynamic obstacles, the last two environments are more complicated scenarios where the dynamic objects each appears at a random time and moves either forward or backward. Each environment was tested in fifty times. The termination conditions of the approaches are tabulated in TABLE 2.

Initial Temperature	Termination Temperature	Cooling Rate	Deleting Operator Rate	Other Operator Rate
9999	5555	0.97	70%	30%

TABLE 2: Control Parameters for MSA

4.2 Simulation Results for Case One

Environment One contains two dynamic obstacles which will appear simultaneously as well as three static obstacles with ten vertices. It is depicted in FIGURE 5, where the black and fully filled blocks represent static obstacles; the hollow triangles show the trajectories of the dynamic obstacles. The sequence of the points in the figure is the travel trajectory of the robot from the start to the end points. The arrows indicate the moving directions of the dynamic obstacles.

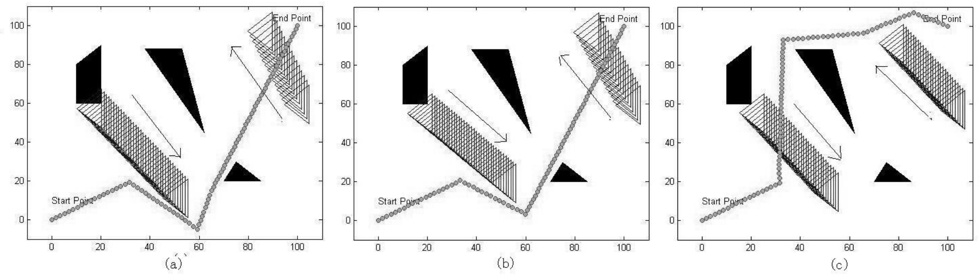


FIGURE 5: Environment One (left to right: MSA, SA, GA)

All four approaches can re-plan the path successfully when moving obstacles are detected and thus no collision has occurred in FIGURE 5. TABLE 3 lists the simulation results of off-line processing time, on-line processing time, and path length. The results show that all approaches have similar path length. For offline and online processing times, MSA approach have the minimum processing time, the normal SA approach outperforms the GA method.

4.3 Simulation Results for Case Two

Compared with Environment One, Environment Two also contains two dynamic obstacles which will appear simultaneously. The total number of the vertices of the static obstacles is 25, compared to 10 in Environment One. FIGURE 6 shows Environment Two and its simulation results for the MSA, normal SA, GA approaches.

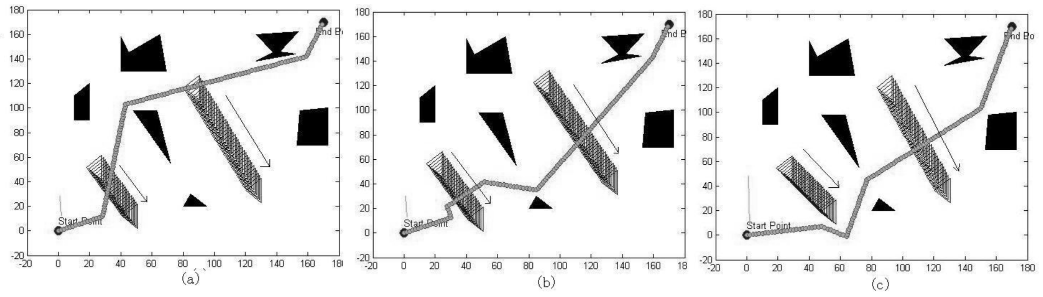


FIGURE 6: Environment Two (left to right: MSA, SA, GA)

TABLE 3 gives some quantitative performance results of the four approaches in off-line processing time, on-line processing times, and path length for Environment Two. It is seen from this table that the performance of the path length can be considered to be comparable for all the approaches. However, for both off-line and on-line processing times, MSA has the best performance, the normal SA is superior to the GA method.

4.4 Simulation Results for Case Three

Environment Three is more complicated than Environment Two. Additional four static and two dynamic obstacles are present in the environment. There are nine static obstacles and four dynamic obstacles altogether; and the number of the vertices of the static obstacles reaches 53. Unlike what we have simulated in the last two environments, the dynamic obstacles in Environment Three do not appear simultaneously. Furthermore, the trajectory of one dynamic obstacle is not a strait line, i.e., the dynamic obstacle changes its direction during movement.

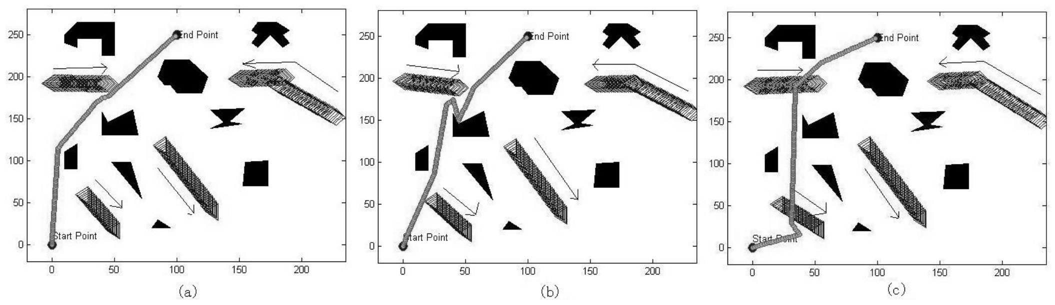


FIGURE 7: Environment Three (left to right: MSA, SA, GA)

FIGURE 7 shows Environment Three and its simulation results for the MSA, normal SA, GA approaches. No collision has occurred in all four approaches, implying that all approaches can re-plan the path successfully when moving obstacles are detected. TABLE 3 gives some quantitative simulation results for Environment Three. It is seen from this table that in all performance criteria (off-line processing time, on-line processing time, and path length), the normal SA is significantly better than the GA method; and the MSA performs much better than the normal SA.

4.5 Simulation Results for Case Four

Environment Four is the most complicated scenario in our simulation studies. There are fourteen static obstacles and six dynamic obstacles altogether in the environment. The number of the vertices of the static obstacles is 82. The dynamic obstacles appear randomly at different times and move in different directions. Also, the dynamic obstacles can change their moving directions during movement.

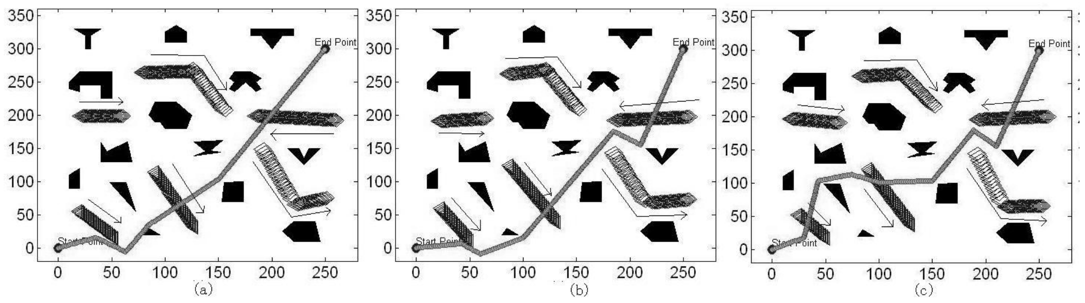


FIGURE 8: Environment Four (left to right: MSA, SA, GA)

FIGURE 8 shows Environment Four and its simulation results for the MSA, normal SA, GA approaches. Again, the robot does not collide with any obstacles in all the approaches, implying that all approaches work well in re-planning of the robot path. TABLE 3 summarizes some quantitative results for Environment Four. It is seen from these results that among the three approaches, the MSA performs the best and the GA method gives the worst performance on the processing time, and the MSA approach gives the optimal path length result.

Environment/Performance	**MSA**	**Normal SA**	**GA**
Environment One			
Offline Processing Time	0.1421	0.1497	0.9147
Online Processing Time	0.1445	0.1514	1.1247
Path Length	145.78	145.78	145.78
Environment Two			
Offline Processing Time	0.2281	0.3817	1.7912
Online Processing Time	0.2411	0.4015	2.0713
Path Length	246.248	256.76	261.46
Environment Three			
Offline Processing Time	0.3418	0.9012	3.3452
Online Processing Time	0.3519	1.1075	3.9716
Path Length	280.85	290.36	305.43
Environment Four			
Offline Processing Time	0.3918	1.4098	4.1214
Online Processing Time	0.4125	1.6987	4.3123
Path Length	410.24	443.67	460.67

TABLE 3: Summary of Performance Results

5. Experiment Results Evaluation

5.1 Path Length Evaluation

FIGURE 9 graphically compares the path length performance of the three approaches for all four environments. Taken from the quantitative simulation results shown in TABLE 3, the values of the path length in the figure are median values obtained in offline path planning. It is seen from FIGURE 9 that the path length performance of all three approaches deteriorates when the environment becomes more complicated; while the MSA approach performs the best in all cases.

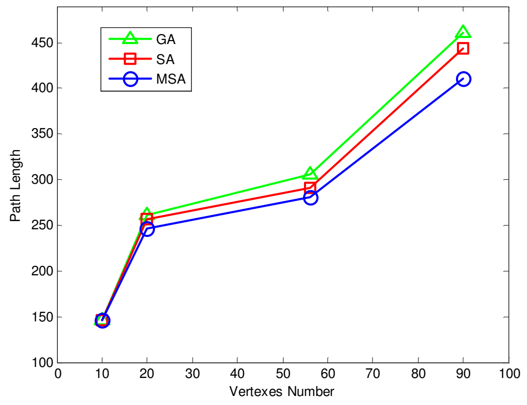


FIGURE 9: Path Length Evaluation

In the simplest environment, i.e., Environment One, all three approaches give the same path length. However, for Environments Two through to Four, the MSA approach improves the path length performance over the normal SA and GA approaches. For example, for Environment Four, the path length of the MSA is 10.9% shorter than that of the GA approach.

5.2 Offline Processing Time Evaluation

FIGURE 10 graphically demonstrates the offline processing time performance of the three approaches for all four environments. The offline planning is conducted based on the static obstacles in the environments. It is seen from FIGURE 10 that the MSA approach is significantly superior to the other two approaches. This is further verified by the quantitative comparison results in TABLE 3. For example, for Environment Four, the MSA improves the offline processing time performance by 72.5% and 90.9% over the normal SA and GA approaches, respectively.

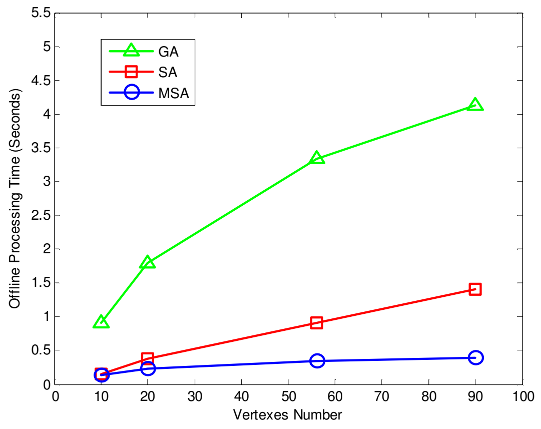


FIGURE 10: Offline Processing Time Evaluation

5.3 Online Processing Time Evaluation

Once a dynamic obstacle is detected, the on-line planner will calculate if a collision is likely to happen. If no collision is likely to occur, the robot keeps traveling along its current path; otherwise, the online planner will re-plan an alternative path for the robot. FIGURE 11 compares the processing time of the online path planning of the three approaches in all four environments. It clearly shows the superiority of the MSA approach to the other two approaches. As an example, in Environment Four with total about 90 vertices, the MSA approach consumes 81.5% less time to re-plan the path than the normal SA approach, and 92.1% less time than the GA approach.

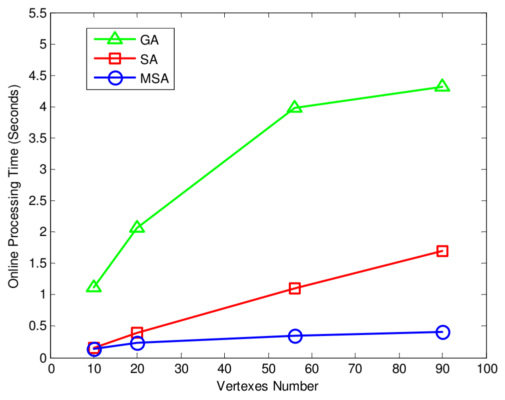


FIGURE 11: Online Processing Time Evaluation

6. CONSLUSION

A Multi-Operator SA (MSA) approach has been proposed for robot path planning in dynamic environments with both static and dynamic obstacles. The contributions of the work include the implementing of the simulated annealing algorithm for robot path planning in dynamic environments, and the enhanced new path planner for improving the efficiency of the path-planning algorithm. Comprehensive case studies and statistical analysis have been carried out to demonstrate the proposed approach in four dynamic environments with different complexities. The MSA has been shown to be capable of giving an optimal or near-optimal path solution in various dynamic environments, and to consume much less processing time than the standard SA with two operators. Unlike the popular A* or D* based approaches, it uses the vertices of the obstacles as the search space. Compare to the previous published SA method; the proposed MSA approach introduces two more additional mathematical operators to ensure the quality of the path solutions in the evolutionary computation. With comparisons with the normal SA and GA, the MSA approaches has been shown through case studies for four dynamic environments to be effective in getting quality path solution and computationally efficient in deriving the path solution. As a result of the significant improvement in the computational efficiency, real-time and on-line applications of the developed approach in dynamic path planning become possible.

7. REFERENCES

[1] P. S. Y. Wang, J. Mulvaney, "*Genetic-based mobile robot path planning using vertex heuristics*," in Proceedings of the Conference on Cybernetics and Intelligent Systems, vol. 1, Bangkok, Thailand, June 7–9, 2006, pp. 1 – 6.

[2] Ahmed Mustafa, Aisha-Hassan A, *"Adaptive Emotional Personality Model based on Fuzzy Logic Interpretation of Five Factor Theory,"* International Journal of Computer Science and Security, vol. 3, no. 3, pp. 210–215, Sept. 2009.

[3] Dzulkifli Mohamad, *"Multi Local Feature Selection Using Genetic Algorithm For Face Identification,"* International Journal of Computer Science and Security, vol. 1, no. 2, pp. 1–10, Sept. 2007.

[4] J. N. Russell, "*Artificial Intelligence: A Modern Approach*." Berkeley, CA, USA: Prentice Hall, 2003.

[5] L. Hu and Z. Q. Gu, "*Research and realization of optimum route planning in vehicle navigation systems based on a hybrid genetic algorithm,*" Proceedings of the Institution of Mechanical Engineers Part D – Journal of Automobile Engineering, vol. 222, no. D5, pp. 757–763, May 2008.

[6] J. Ayers, "*Underwater walking,*" Arthropod Structure and Development, vol. 33, no. 3, pp. 347–360, July 2004.

[7] B. Williams and I. Mahon, "*Design of an unmanned underwater vehicle for reef surveying*," in Proceedings of the IFAC 3rd Symposium on Mechatronic Systems. Manly NSW, Australia: IEEE, Sept. 15, 2004.

[8] S. Chakravorty and J. L. Junkins, "*Motion planning in uncertain environments with vision-like sensors,*" Automatica, vol. 43, no. 12, pp. 2104–2111, Dec. 2007.

[9] Y. Wang, P. W. Sillitoe, and J. Mulvaney, "*Mobile robot path planning in dynamic environments,*" in Proceedings of the International Conference on Robotics and Automation, vol. 1. Roma: IEEE, Apr. 10–14, 2007, pp. 71–76.

[10] P.-Y. Zhang, T.-S. L¨ u, and L.-B. Song, "*Soccer robot path planning based on the artificial potential field approach with simulated annealing,*" Robotica, vol. 22, no. 5, pp. 563–566, Aug. 2004.

[11] A. Stentz, “*Optimal and efficient path planning for partially-known environments,*” in Proceedings of the IEEE International Conference on Robotics and Automation, vol. 4, San Diego, CA, USA, May 8–13, 1994, pp. 3310–3317.

[12] A. Yahia, A. Stentz, S. Singh, and B. Brummit, “*Framed-quadatree path planning for mobile robots operating in sparse environments,*” in Proceedings of the IEEE Conference on Robotics and Automation, vol. 1. Leuven, Belgium: IEEE, May 16–20, 1998, pp. 650–655.

[13] D. Ferguson and A. Stentz, “*Field D*: An interpolation-based path planner and replanner,*” in Proceedings of International Symposium on Robotics Research, San Francisco, CA, USA, Oct. 12, 2005, pp. 239–253.

[14] A. Stentz, “*The focussed D* algorithm for real-time replanning,*” In Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, Quebec, Canada, pp. 1652–1659, Aug. 20–25, 1995.

[15] A. Yahja, S. Singh, and A. Stentz, “*An efficient online path planner for outdoor mobile robots operating in vast environments,*” Robotics and Autonomous Systems, vol. 32, pp. 129–143, 2000.

[16] A. R. Willms and S. X. Yang, “*An efficient dynamic system for real-time robot path planning,*” IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 36, no. 4, pp. 755–766, 2006.

[17] A. R. Willms and S. X. Yang, “*Real-time robot path planning via a distance-propagating dynamic system with obstacle clearance,"* IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 38, no. 3, pp. 884–893, June 2008.

[18] H. Miao and Y.-C. Tian, “*Robot path planning in dynamic environments using a simulated annealing based approach,*” in Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision - ICARCV’2008, Hanoi, Vietnam, Dec. 17–20, 2008, pp. 1253–1258.

[19] Mathworks, “Matlab,” http://www.mathworks.com, retrived on 18 Feb 2009.

[20] Sufal Das, Banani Saha, *“Data Quality Mining using Genetic Algorithm”*, International Journal of Computer Science and Security, vol. 3, no. 2, pp. 105-112. May 2009.