Improving the cooperation of fuzzy simplified memory A* search and particle swarm optimisation for path planning

Mehdi Neshat*

Department of Computer Science,
College of Software Engineering,
Shirvan Branch,
Islamic Azad University,
Shirvan, Iran
Email: neshat_mehdi@yahoo.com
*Corresponding author

Ali Akbar Pourahmad

Department of Information Science,
College of Library Science,
Shirvan Branch,
Islamic Azad University,
Shirvan, Iran
Email: pourahmad@iau-shirvan.ac.ir

Zahra Rohani

Department of Computer Science,
College of Hardware Engineering,
Shirvan Branch,
Islamic Azad University,
Shirvan, Iran
Email: rohani@iau-shirvan.ac.ir

Abstract: Problem solving is a very important subject in the world of AI. In fact, a problem can be considered one or more goals along with a set of available interactions for reaching those goals. One of the best ways of solving AI problems is to use search methods. The simplified memory bounded A* (SMA*) is one of the best methods of informed search. In this research, a hybrid method was proposed to increase the performance of SMA* search. The combining fuzzy logic with this search method and improving it with PSO algorithm brought satisfactory results. The use of fuzzy logic leads to increase the search flexibility especially when a robot dealing with lots of barriers and path changes. Furthermore, combining PSO saves the search from being trapped into local optimums and provides for search some correct and accurate suggestions. In the proposed algorithm, the results indicate that the cost of search and branching factor are decreased in comparison with other methods.

Keywords: informed search; fuzzy logic; particle swarm optimisation; simplified memory bounded A*; robot navigation.

Reference to this paper should be made as follows: Neshat, M., Pourahmad, A.A. and Rohani, Z. (2020) 'Improving the cooperation of fuzzy simplified memory A* search and particle swarm optimisation for path planning', *Int. J. Swarm Intelligence*, Vol. 5, No. 1, pp.1–21.

Biographical notes: Mehdi Neshat is a faculty member of Computer Science Department at the Islamic Azad University of Shirvan, Iran. His Master's degree was related to Artificial Intelligence from the Islamic Azad University of Mashhad in 2008. Designing intelligent systems, swarm optimisation methods and fuzzy logic are his interests.

Ali Akbar Pourahmad received his PhD in Information and Knowledge Science from the Science and Research Branch of Tehran, Iran, in 2004. He is an Associate Professor in the Center for Advanced Studies in Information Science and Technology, School of Science and the Islamic Azad University of Shirvan, Iran. His research interests include data mining, data processing, knowledge based systems and its application.

Zahra Rohani is a PhD student in Hardware Engineering in the Islamic Azad University of Kerman, Iran, in 2004. He is a faculty member of Computer Science Department at the Islamic Azad University of Shirvan, Iran. His research interests include designing smart circuit, advanced control, robotics and its application.

1 Introduction

In computer sciences and mathematics, a search algorithm is the one which receives a problem as input and returns one or more solutions to it after evaluating possible ones. The set of possible solutions to a problem is named the search space. The goal is always intended to be found in the least possible time and at the lowest cost. Generally, there are two main methods for searching the problem space. The first one is the data-driven search, and the second one is the goal-driven search. The first one always begins from the starting status and makes it possible to reach the goal through authorised operators. This method is also named forward chaining. However, in the goal-driven search or backward chaining method, the search begins from the goal status to reach the starting point. It should be noted that the method proposed in this research was of data-driven type. Many methods have been presented for the informed search of the goal with searching robot. Among them, A* can be referred to as one of the best methods (Stuart and Norvig, 2003).

However, this good method has some defects such as the exponential search time and too much memory which it requires to reach the goal. The combination of iterative deepening search method and A* search was suggested for the memory problem (IDA*). IDA* is appropriate for problems with stepwise cost, and it do not need to keep an ordered queue of nodes. However, like the iterative deepening search version with a steady cost, the number of repetitions would unfortunately increase the calculations and the search time. Managing the memory optimally, SMA* could resolve the problems of A* method to a great extent. Two functions of $h(n)$ and $g(n)$ play key roles in SMA* search. In many searching and routing problems, especially in the real world in which dynamics are high, it is very difficult or almost impossible to determine the accurate values of these two functions. The function $g(n)$ represents the cost of path between current node and the node *n*, while *h*(*n*) is the estimated cost of the cheapest path between the node n and the goal node. If the environment is definite, static, discrete and accessible, it will be easy to estimate these two functions. Otherwise, these values cannot be easily estimated. Benefiting from fuzzy logic in environments with uncertainty would be an optimised and appropriate solution. Fuzzy logic is an appropriate tool for modelling and inferring indefinite knowledge and transforming it into definite knowledge. Despite all the advantages of FSMA*, this method has problems dealing with local optimums. In various tests of this method, it was observed that a cell which was not optimised would be selected. This cell would also increase the search cost. One of the best swarm intelligence methods named particle swarm optimisation (PSO) was used to solve this problem because this method would represent a high convergence rate. It would also have less time complexity in comparison with other methods of swarm intelligence. The results obtained through the combination of PSO and FSMA* indicated that the proposed method had a better performance.

2 Literature review

Hybrid fuzzy A* method was suggested for better robot planning, and some good results were obtained (Gerdelan and Reyes, 2006a). It was used in different problems (Gerdelan et al., 2006; Gerdelan and Reyes, 2006b). One use of fuzzy logic and A* search was in unsupervised underwater robot planning (Anwary, 2008). Other routing applications of fuzzy logic include the generalisation of ring-like routing through the shortest fuzzy path (Boulmakoul, 2004), routing based on several fuzzy factors (Soltani and Fernando, 2004), and the use of fuzzy operations in the navigation of robot in virtual environments (Jaafar and McKenzie et al., 2007). Moreover, fuzzy routing based on virtual potential field (Tanasie et al., 2007) and directing and routing the robot in indefinite environments using fuzzy mapping and networking were among the studies which have been conducted in this area (Karaman and Temelta, 2005).

Furthermore, different methods of swarm intelligence were widely used in path-planning in the recent decade. For instance, using PSO algorithm to optimise path-planning algorithm and clustering in wireless sensor networks indicate that the results were more optimal in comparison with other methods including LDC (Ataul et al., 2008), GLBCA (Low et al., 2008), and GA (Kuila et al., 2013; Kuila and Jana, 2014). Several review papers are published about the application of classic methods and swarm intelligence in path-planning networks, especially wireless sensor networks. They can provide us with a general view of the importance, range of applications, and brilliant results of such Swarm Intelligence methods in comparison with other methods of path-planning (Saleem et al., 2011; Kemal and Mohamed, 2005; Zungeru et al., 2012). Ant colony optimisation algorithm (ACO) was used in finding optimal path-planning in wireless sensor networks too, and the results were better than those of AGRA and M-IAR methods in terms of load balancing, adaptability, and scalability (Bi et al., 2010).

Another path-planning application of ACO was to combine it with QoS method. This method (ACO-QoSR) could decrease the computational complexity and energy consumption in routing (Cai et al., 2006; Camilo et al., 2006). Chen et al. (2007) introduced a meta-heuristic method named jumping ant routing algorithm (JARA) in order to decrease the search time and system overhead. This method resulted in a good performance. Improving ant colony optimisation through reinforced learning (RL) and using it for routing wireless sensor networks was introduced in 2007 (Ghasemaghaei et al., 2007). Inspired by the way of creating connections among specific mold cells which is named physarum polycephaum and creating very narrow pipelines through them, Li et al. (2011) introduced a mathematical model, which was completely innovative of its type for path-planning. Bee colony optimisation algorithm was also used to solve this problem (Saleem and Farooq, 2007). PSO algorithm of discrete type had good results in planning distributed routing (Chengming, 2011). In the recent decade, PSO algorithm was more commonly used to solve the problem of routing, and it had good results (Toofani, 2012; Sarangi1 and Thankchan, 2012).

According to the newest publications in this way, Bakdi et al. (2017) recommends a hybrid method which is a combination of genetic algorithm and adaptive fuzzy control. The method can trade-off a great balance between minimum path length, maximum safety and shortest trajectory runtime. Although, the method is faster than the traditional path planning algorithms, using the new version of genetic algorithm like a hybrid adaptive genetic algorithm (Yun et al., 2017) or utilising new swarm intelligence method such as swallow swarm optimisation (Neshat et al., 2013; Neshat and Sepidname, 2015) can be effective. Furthermore, a new hierarchical path planning method is published for mobile robots in jumbled environment which is benefited by a Pareto dominance PSO (Mac et al., 2017). However, it is better to apply different kinds of adaptive PSO to deal with trapping into local optimum including fuzzy adaptive informed PSO (Neshat, 2013) and predator PSO (Neshat et al., 2012).

3 Materials and methods

3.1 *Fuzzy simplified memory bounded A**

In addition to all of its advantages, SMA* search has some disadvantages, too. The performance of this search highly depends on the defining strategy of two functions *g*(*n*) and *h*(*n*), in so far as the optimality and even completion of this method is questioned if these functions are not defined and selected correctly. In fact, it is not very hard to define these functions if the environment is static, deterministic with low complexity. An instance can be determining the optimal distance between two cities in a country. However, it is so complex to define the optimal path for a mine-detecting robot in an old field covered with different natural barriers such as uneven coating plants, pits, streams, pools of water, rocks and so on.

The membership function *g*(*n*) is expressed in a fuzzy manner as follows:

$$\tilde{g}(n_i) = \begin{cases} \mu_g(n_1)/n_1 + \mu_g(n_2)/n_2 + ... + \mu_g(n_e)/n_e \\ \qquad = \sum_{i=1}^{n} \mu_g(n_i)/n_i \\ \end{cases} \tag{1}$$

n_e is the target node. Triangular and trapezoidal membership functions were used to express fuzzy function $\tilde{g}(n)$. Figure 1 indicates a view of this function: Verbal expressions (*low, average, high, very high cost*) are used.

Figure 1 Fuzzy membership function $\tilde{g}(n)$ (see online version for colours)

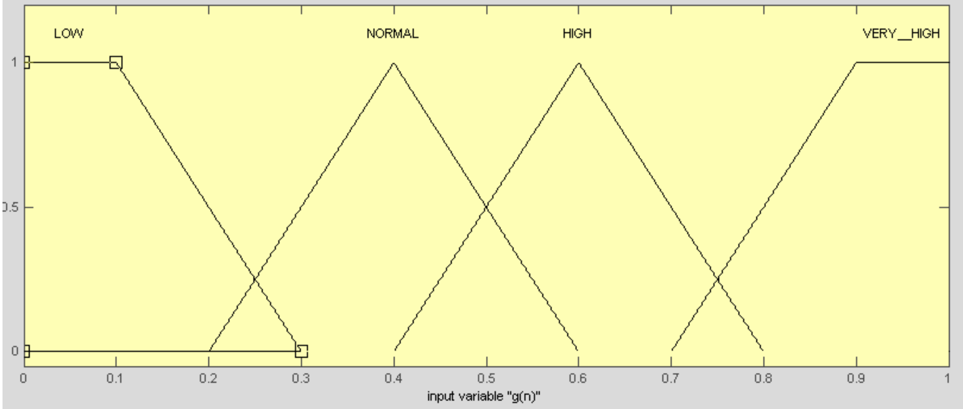
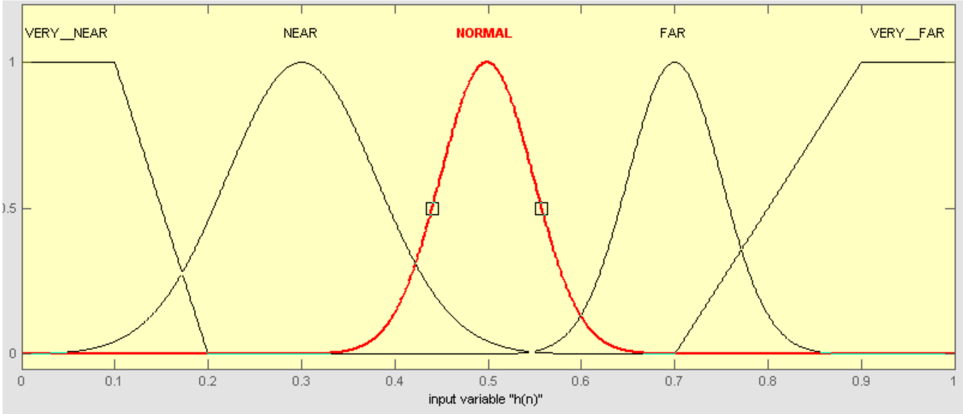


Figure 2 Fuzzy membership function $\tilde{h}(n)$ (see online version for colours)



Triangular and trapezoidal fuzzy membership functions were used to make search fuzzy calculations simpler and faster. Fuzzification equation (2) of $\tilde{g}(n)$ function were expressed as follows. In different environments, they are multiplied by ranges of cost and heuristic fuzzy functions as a coefficient with respect to the route cost values and heuristic function value.

$$\mu_{g(n)}low = \begin{cases} \overline{0} & \overline{x \le 0} \\ \overline{1} & 0 < x \le 0.1 \\ 0.3 - x \big/ 0.2 & 0.1 < x \le 0.3 \end{cases} \tag{2}$$

$$\mu_{g(n)}normal = \begin{cases} x - 0.2 \big/ 0.2 & 0.2 \le x < 0.4 \\ 0.6 - x \big/ 0.2 & 0.4 \le x \le 0.6 \end{cases} \quad \mu_{g(n)}high = \begin{cases} x - 0.4 \big/ 0.2 & 0.4 \le x < 0.6 \\ 0.8 - x \big/ 0.2 & 0.6 \le x \le 0.8 \end{cases}$$

As function $h(n)$ is an estimation, it can be combined well using fuzzy logic. Verbal variables should be used with a more suitable interval to improve the search accuracy. Verbal variables (*very near, near, average, far, very far*) were used, and membership functions were Gaussian and trapezoidal. Figure 2 indicates fuzzy function $\tilde{h}(n)$.

Fuzzification membership equation (3) for function $\tilde{h}(n)$ are as follows:

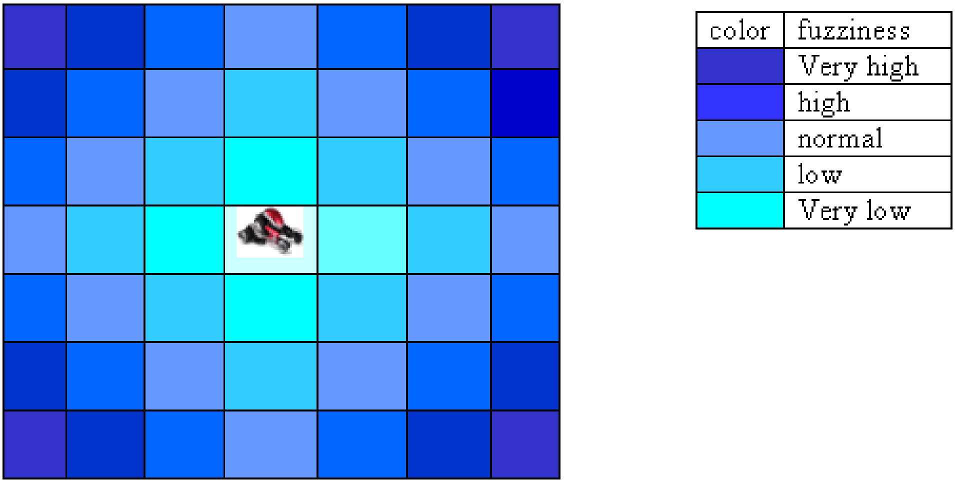
$$\mu_{h(n)} verynear = \begin{cases} 0 & x \leq 0 \\ 1 & 0 < x \leq 0.1 \\ 0.2 - x/0.1 & 0.1 < x \leq 0.2 \end{cases}$$

$$f(x;\sigma,c) = e^{\frac{-(x-c)}{2\sigma^2}} \rightarrow \mu_{h(n)} near = e^{\frac{-(x-0.3)}{2*(0.1)^2}}$$

$$\mu_{h(n)} normal = e^{\frac{-(x-0.5)}{2*(0.07)^2}} \qquad \mu_{h(n)} far = e^{\frac{-(x-0.7)}{2*(0.07)^2}} \tag{3}$$

This research proposes a new function called 'digress'. Function $\tilde{d}(n)$ is an unfavourable factor for search route of the environment. Instant route change and/or diagonal movement requires a higher cost. Such movements lengthen search time and slow the movement of an agent. Figure 3 indicates a sample of agent movement and cost unfavourability of environment search. This is a 7*7 frame with a fully adaptive behaviour. It assesses an environment's unfavourability rate constantly. By mapping suitable values, it may help a more informed search in indefinite and dynamic environments. In points with barriers, fitness value is set on the infinite mode. In this condition, a robot will never hit it.

Figure 3 Fuzzy values for the new unfavourability function (see online version for colours)



color	fuzziness
🟦	Very high
🟦	high
🟦	normal
🟦	low
🟦	Very low

The farther the distance is and the more divergent the route is from the robot route, the more the cost will be. The function is always unable to provide an accurate value to express unfavourability; therefore, it is expressed as a fuzzy function. Figure 4 indicates the membership functions of fuzzy function $\tilde{d}(n)$.

Figure 4 Fuzzy membership function $\tilde{d}(n)$ (see online version for colours)

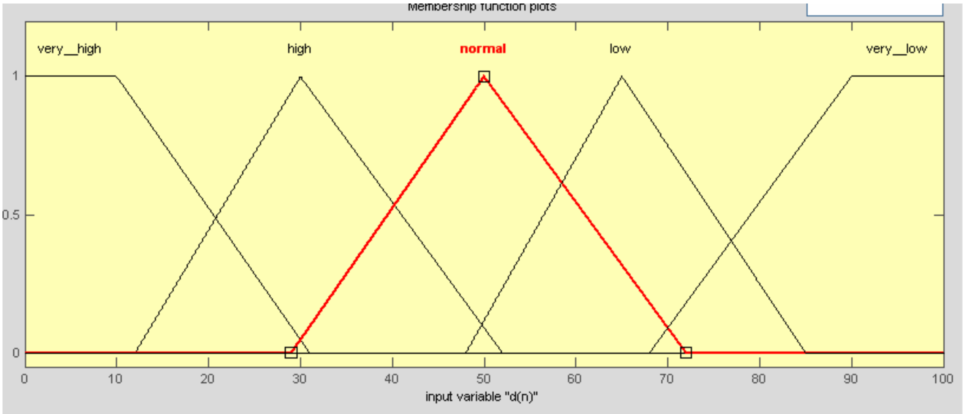
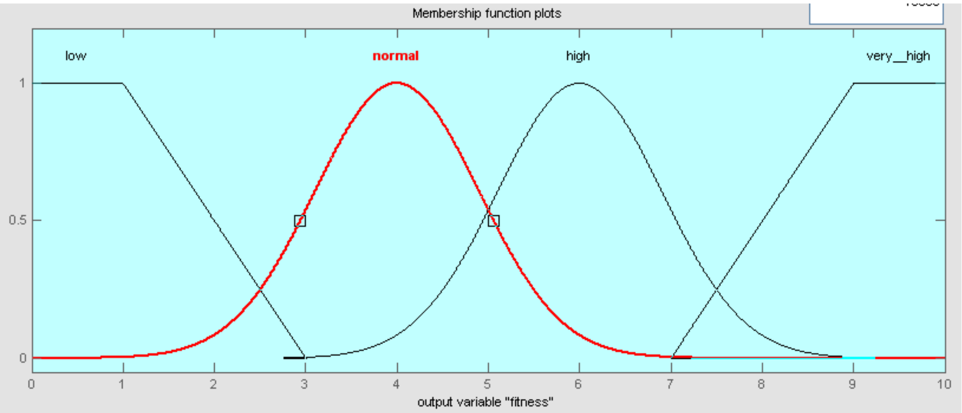


Figure 5 Fuzzy membership function $\tilde{f}(n)$ (see online version for colours)



Fuzzy membership equation (4) are calculated as follows:

$$\mu_{d(n)}high = \begin{cases} x-12/18 & 12 \le x < 30 \\ 30-x/22 & 30 \le x \le 52 \end{cases} \qquad \mu_{d(n)}normal = \begin{cases} x-30/20 & 30 \le x < 50 \\ 72-x/22 & 50 \le x \le 72 \end{cases}$$

$$\mu_{d(n)}low = \begin{cases} x-48/19 & 48 \le x < 67 \\ 85-x/18 & 67 \le x \le 85 \end{cases} \tag{4}$$

Figure 5 shows the membership function of $\tilde{f}(n)$.

This function consists of four fuzzy membership functions in the order of *low, normal, high, very high*. *Low* and *very low* functions are of trapezoidal fuzzy membership functions, while normal and high functions are of Gaussian functions.

3.2 The second proposed method (FSMA*PSO)

Cost functions $g(n)$ and heuristic function $h(n)$ were fuzzified through fuzzy SMA* method. Moreover, a new function called unfavourability function $d(n)$ was introduced to control the movements. It would absorb high energy from the agent. This function was also fuzzified and simulated with (very low, low, normal, high, very high) membership functions. Then 15 fuzzy rules were set for different conditions of the agent along the search route. The rules help the agent make a logical and clever decision while encountering different conditions. Finally, Centre of Gravity (CoG) method was used for defuzzification.

Fuzzy-SMA* method flow chart was illustrates as Figure 6. However, it should be noted that some details were not mentioned in the flow chart to avoid excessive complexity and to have a better understanding .It was meant to indicate the performance of the method to express more complex FSMA*PSO method more conveniently.

There is a general principle for the children nodes after fuzzy calculation of $f(n)$ value in the first proposed method; that is the selection of the node with the minimum value rooted in the classic method of A*. The selection of the minimum child might seem logical at first sight; however, this selection is not always true because it is made greedy! This is a fully local selection in the next step. That is because we always face the problem of backtracking and selecting another node in A* family searches (A*, IDA*, SMA*). Although the robot has no reaction until the optimal path is found, the very back-trackings cause an increase in system overhead.

The possibility of evaluating several cells later must be provided in searches to solve this problem. That is, the agent must have a spectrum where it can identify the optimal points. The PSO algorithm is capable of discovering the global optimal point in an environment. This method proposes an optimal way depending on the converging property of the particles and social intelligence. Hence, a determined number of particles start to search for space in the agent spectrum and are converged around an optimum point after a determined number of repetitions. This point may be a good candidate for the agent to continue its operation. Figure 7 presents an outline of SMA*PSO operation.

The PSO Algorithm structure is simple while it contains intricate points. For example, the parameters of this algorithm have a considerable effect on the searching performance. There are two types of critical parameters in this algorithm: acceleration values C_1, C_2 and the inertia weight (ω). Applying the inertia weight can cause the reconciliation between the global and local group discoverabilities. The high inertia weight is in fact a motivation for discovery all over the areas (some already unsearched environments), whereas the lower weight is for discoveries in local areas. A lower weight actually leads to a more accurate search in previously searched regions. Selecting the appropriate weigh for it insures a desirable balance between the global and local discoverabilities which then leads to a more effective algorithm functioning. The experimental results indicate that the selection of high amounts for it at the beginning of the search causes the priority of global discoveries to be higher than the local ones, and the local search will be continued more seriously on the gradual reduction of the search (Zhan et al., 2012).

Generally, the functioning of the second proposed algorithm can be presented in a comprehensive outline as follows (Figure 8).

Figure 6 Fuzzy-SMA* method flow chart

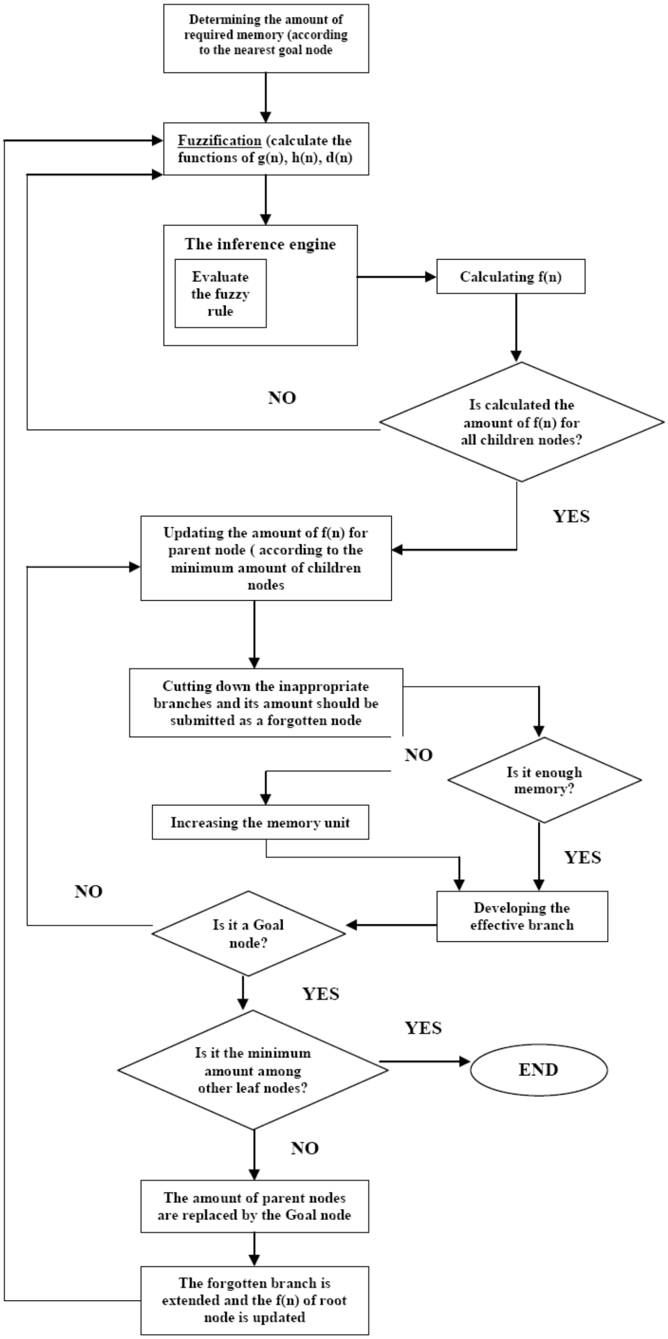

Determining the amount of required memory (according to the nearest goal node
Fuzzification (calculate the functions of g(n), h(n), d(n)
The inference engine
Evaluate the fuzzy rule
Calculating f(n)
NO
Is calculated the amount of f(n) for all children nodes?
YES
Updating the amount of f(n) for parent node (according to the minimum amount of children nodes
Cutting down the inappropriate branches and its amount should be submitted as a forgotten node
NO
Is it enough memory?
Increasing the memory unit
YES
NO
Developing the effective branch
Is it a Goal node?
YES
YES
Is it the minimum amount among other leaf nodes?
END
NO
The amount of parent nodes are replaced by the Goal node
The forgotten branch is extended and the f(n) of root node is updated

Figure 7 The PSO algorithm operation in the agent spectrum (see online version for colours)

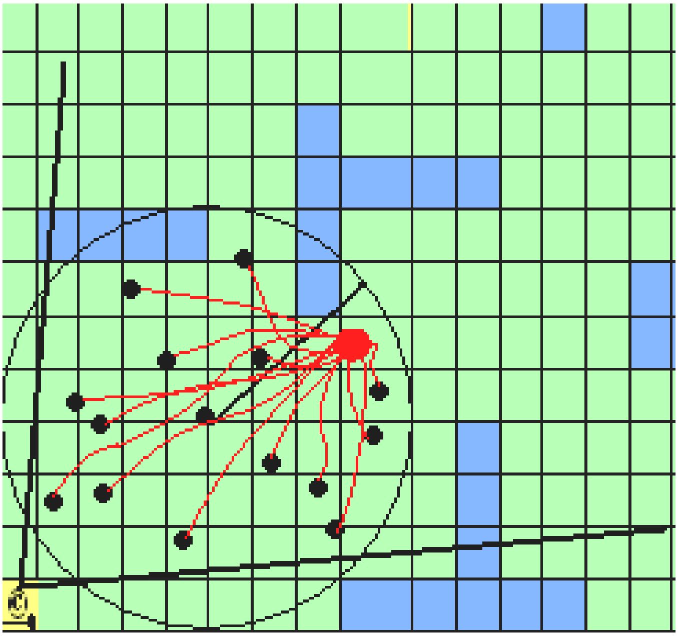
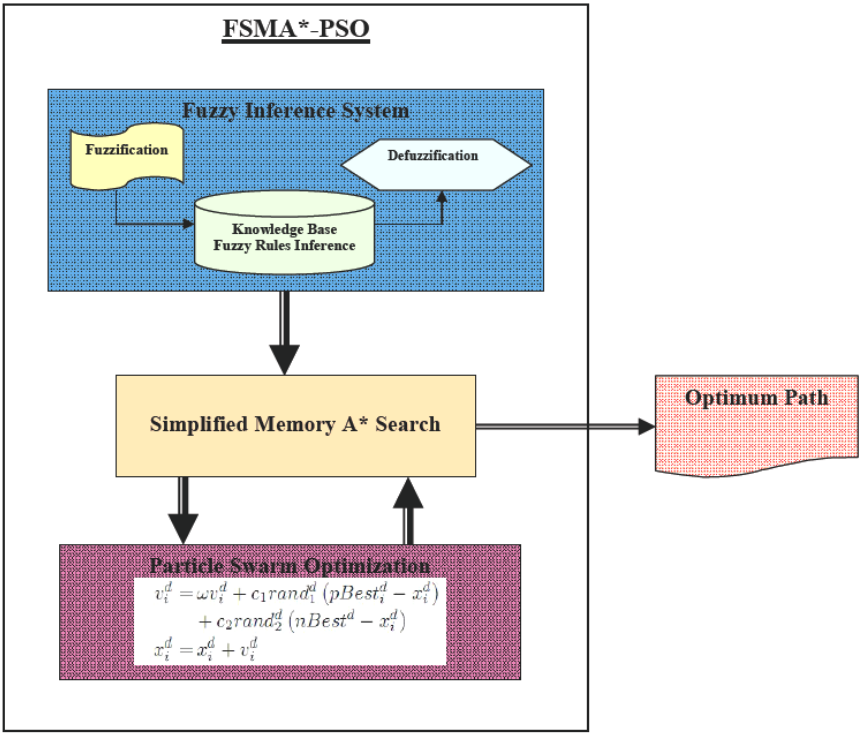


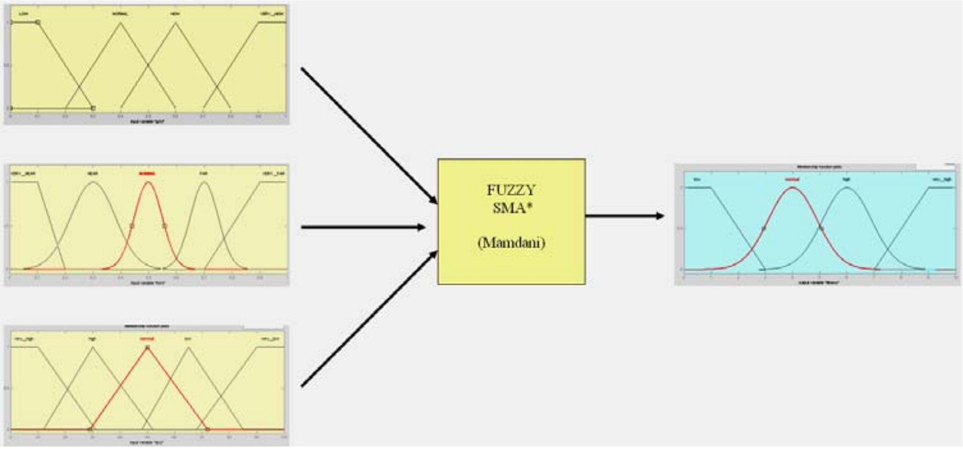
Figure 8 Presents the method of interaction between FSMA*-PSO (see online version for colours)


FSMA*-PSO
Fuzzy Inference System
Fuzzification
Defuzzification
Knowledge Base
Fuzzy Rules Inference
Simplified Memory A* Search
Optimum Path
Particle Swarm Optimization
$v_i^d = \omega v_i^d + c_1 rand_1^d (pBest_i^d - x_i^d) + c_2 rand_2^d (nBest^d - x_i^d)$
$x_i^d = x_i^d + v_i^d$

4 Results and discussion

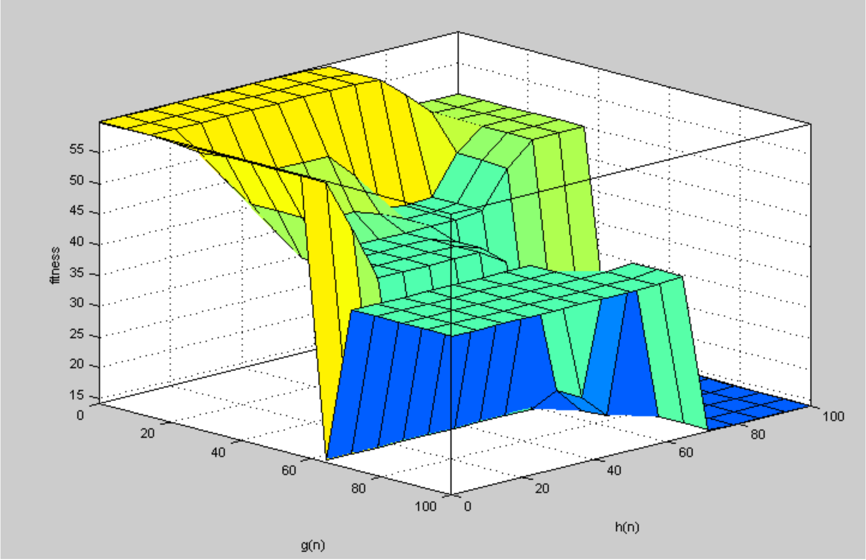
The proposed method is implemented using MATLAB 7.0.4. The overall structure of SMA* fuzzy system is presented in Figure 9.

Figure 9 The outline of SMA* method (see online version for colours)



The method of selecting the next node based on *f*(*n*) of each node is performed based on the conventional classic method of SMA*. There are some relationships among the input functions. The relationship between the cost $\tilde{g}(n)$ and the approximate (heuristic) $\tilde{h}(n)$ functions is presented in Figure 10.

Figure 10 The fuzzy relationship between both $\tilde{h}(n)$ and $\tilde{g}(n)$ functions (see online version for colours)



As seen in the Figure 11, the value of $\tilde{h}(n)$ is at the maximum level at the beginning, is reduced gradually as we get closer to the target and, finally, reaches to 0 in the target point. On the other hand, the $\tilde{g}(n)$ value equals 0 at the beginning; however, its value increases to reach to the maximum value at the target point. It is, of course, an optimal maximum value. These two functions have a linear reverse relationship in the SMA* classic method; however, this relationship has intricate complexities in the FSMA* method occurring in the searching process. These complexities are shown in Figure 11.

Figure 11 Fuzzy relationship between two functions $\tilde{d}(n)$ and $\tilde{g}(n)$ (see online version for colours)

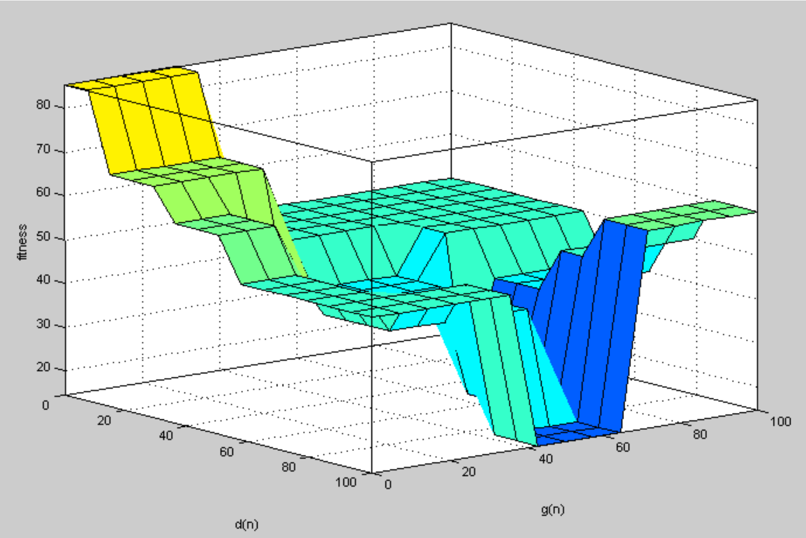
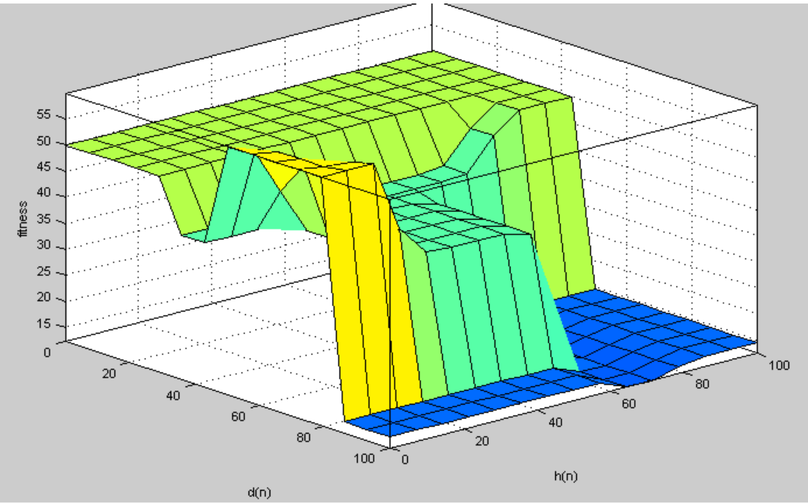


Figure 12 Fuzzy relationship between two functions $\tilde{h}(n)$ and $\tilde{d}(n)$ (see online version for colours)



In robot routing, the lower costs of $\tilde{g}(n)$ and $\tilde{h}(n)$ are, the more optimal the value of fitness is because $\tilde{f}(n)$ decides to select a more optimal node based on the sum of these two values. However, the value of unfavourability function is also effective. Figure 11 indicates the relationship between two functions $\tilde{g}(n)$ and $\tilde{d}(n)$, and Figure 12 shows the relationship between $\tilde{h}(n)$ and $\tilde{d}(n)$.

In both images, the impact of unfavourability function on the fitness value is clear. The smaller the value of function $\tilde{d}(n)$ is, the better the fitness will be. Given the input values of the fuzzy system SMA*, the set of movement rules are indicated in Figure 13.

Figure 13 Fuzzy rules of inference

1 if (g(n) is VERY_HIGH) and (h(n) is VERY_FAR) and (d(n) is very_low) then (fitness is low) (1)

2 if (g(n) is VERY_HIGH) and (h(n) is VERY_FAR) and (d(n) is low) then (fitness is low) (1)

3 if (g(n) is NORMAL) and (h(n) is NORMAL) and (d(n) is normal) then (fitness is normal) (1)

4 if (g(n) is LOW) and (h(n) is VERY_NEAR) and (d(n) is very_high) then (fitness is very_high) (1)

5 if (g(n) is LOW) and (h(n) is NERA) and (d(n) is high) then (fitness is high) (1)

.
.
.
.
.
.
.
.
.
.

12 if (g(n) is NORMAL) and (h(n) is VERY_NEAR) and (d(n) is low) then (fitness is high) (1)

13 if (g(n) is VERY_HIGH) and (h(n) is NEAR) and (d(n) is low) then (fitness is normal) (1)

14 if (g(n) is VERY_HIGH) and (h(n) is NEAR) and (d(n) is very_low) then (fitness is high) (1)

15 if (g(n) is LOW) and (h(n) is VERY_FAR) and (d(n) is very_high) then (fitness is high) (1)

One of the disadvantages of classic SMA* is the assessment of detour routes which would slow down this method. Since it has always been very difficult to estimate function $h(n)$, if a unsuitable function $h(n)$ is selected in routing, the speed search will decrease, and the optimality of path will be questioned. An environment of 64*64 cells were considered in order to compare SMA* with FSMA*. There were some barriers in this environment. Robot was placed in cell 1*1. It was supposed to find the goal which was put in cell 64*64. Figure 14 indicated the test environment of the application:

Figure 14 The environment simulated to test two methods (see online version for colours)

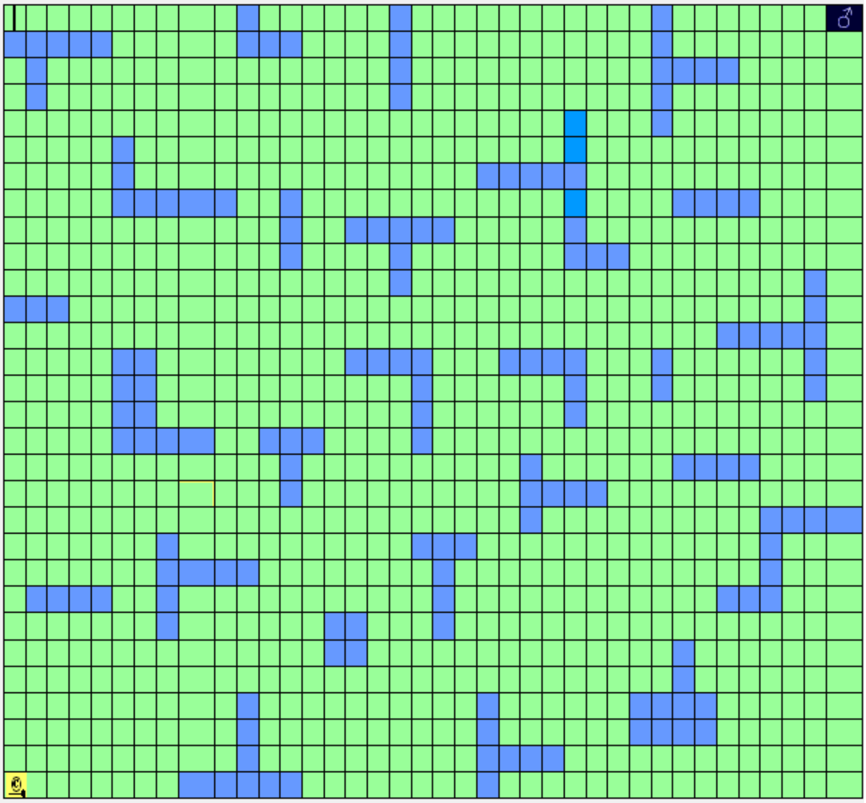


Table 1 The comparison of FSMA*, SMA*, A*, and IDS with each other

d	*Search cost (f(n))*				*Average branching factor*			
	IDS	*A**	*SMA**	*FSMA**	*IDS*	*A**	*SMA**	*FSMA**
2	15	9	9	7	7.71	6.54	6.54	6.54
6	7,850	128	128	115	7.56	6.16	5.42	5.29
10	18,569	1,580	1,580	784	7.12	5.67	5.23	5.02
16	5,894,156	9,784	7,590	4,362	7.32	5.41	5.09	4.57
20	12,586,904	13,852	9,015	6,519	7.02	5.02	4.86	4.31
25	-	22,549	14,521	9,751	-	4.59	4.64	4.16
30	-	46,750	19,843	12,470	-	4.38	4.19	3.94
35	-	91,586	25,410	16,843	-	4.08	3.82	3.62

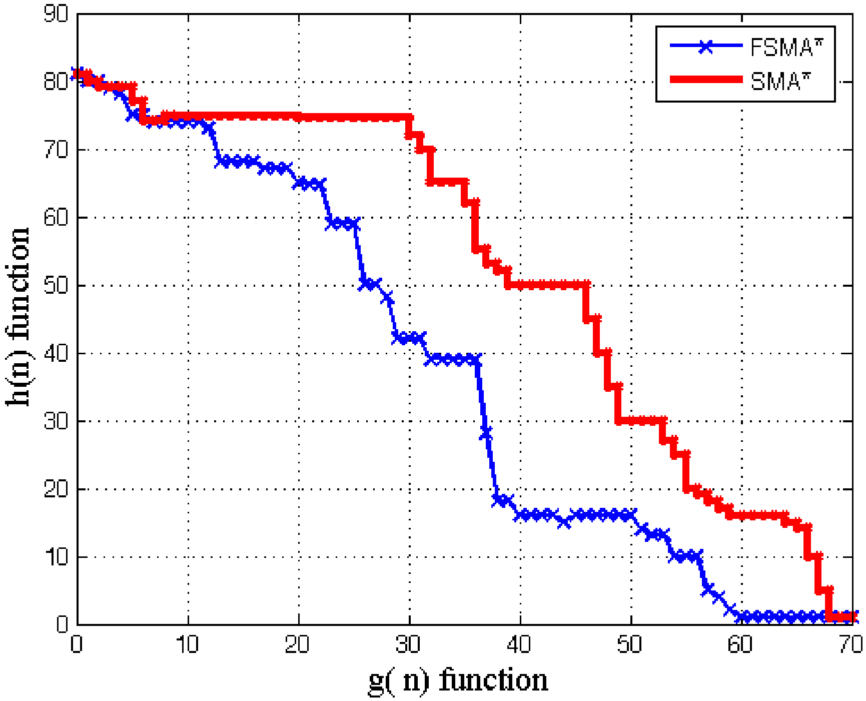
Note: Based on search cost and the mean of branching factor.

The value of cost function $g(n)$was 1 for each horizontal or vertical movement, whereas it was $\sqrt{2}$ for each diagonal one. Function $h(n)$ was based on formula Manhattan distance to goal. Function *d*(*n*) of unfavourability coefficient was calculated like Figure 3.The following table presents the comparison of different search methods such as iterative deepening search (IDS) which is an uninformed search method. It also indicates search methods such as A*, SMA* and FSMA*. The parameters which were compared included the mean of branching factor and search cost, which is equal to the expanded nodes.

As observed in Table 1, the cost of search is very high in IDS method because this method would expand many of repetitive nodes. This matter would increase the cost of search and the mean of branching factor. Two methods of SMA* and A* would have similar performances at low depths. It means that their search costs are almost the same. However, increasing the depth of search reduced the cost of SMA* search because this method would prune unnecessary nodes and send them to disk. The proposed search method (FSMA*) did not have a very different performance from SMA* and A* at low depths such as 2. At greater depths, FSMA*, however, indicated better performances on both the cost of search and the mean of branching factor due to using fuzzy functions *h*(*n*) and $\tilde{d}(n)$.

As observed, the goal was reached at a lower cost in FSMA* search method, and the movements of robot were much more flexible at turns. In this method, the robot could get close to the barriers as much as possible without hitting them. It could also avoid additional and misleading movements. Figure 15 indicates the performances of SMA* and FSMA* in reaching the goal.

Figure 15 The comparison of performances of SMA* and FSMA* with respect to functions *g*(*n*) and *h*(*n*) (see online version for colours)



The same environment of 64*64 cells was used in order to test the second proposed method. The fuzzification method and fuzzy membership functions $\tilde{d}(n)$, $\tilde{g}(n)$ and $\tilde{h}(n)$ were the same as the first proposed method. Different values were selected for coefficients of acceleration and weight of inertia in order to evaluate and analyse the impact of PSO parameters on the result of search cost so that the most optimal value would be obtained. Table 2 indicates different values of C_1, C_2 and ω. Moreover, the depth of search tree was equal to 35.

Table 2 Search cost with respect to different values for coefficients of acceleration and inertia weight (see online version for colours)

ω	$C_1 = 1.4$ $C_2 = 2.6$	$C_1 = 1.8$ $C_2 = 2.2$	$C_1 = 2.2$ $C_2 = 1.8$	$C_1 = 2.6$ $C_2 = 1.4$	$C_1 = 3$ $C_2 = 1$
0.1	15,356	15,401	15,522	15,560	15,762
0.2	15,192	15,389	15,410	15,680	15,607
0.3	15,290	15,269	15,570	15,523	15,792
0.4	15,049	15,310	15,287	15,405	15,569
0.5	15,005	15,224	15,301	15,344	15,541
0.6	14,859	15,014	15,294	15,362	15,408
0.7	14,792	14,762	14,953	15,209	15,325
0.8	14,620	14,368	14,708	14,811	15,107
0.9	14,255	14,290	14,379	14,652	14,886
1.0	14,211	14,251	14,221	14,243	14,391

As indicated in Table 2 and Figure 16, the best case was represented with $\omega = 1$, $C_1 = 1.4$, $C_2 = 2.6$, and the highest search cost was related to $\omega = 0.3$, $C_1 = 3$, $C_2 = 1$.

Figure 16 Effect of the acceleration value changes rate in the presence of inertia ratio on the search cost (see online version for colours)

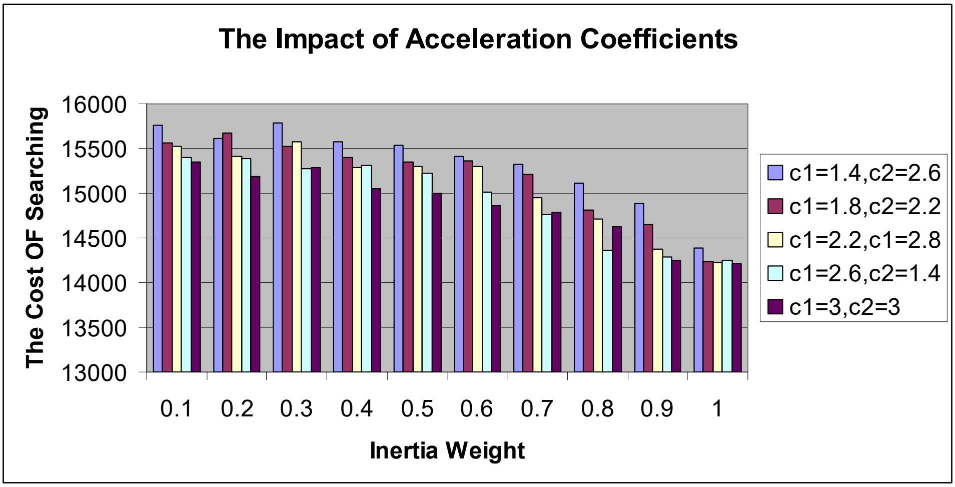


Table 3 Comparing the performance of the proposed method with the other three methods in terms of search cost and branching factor

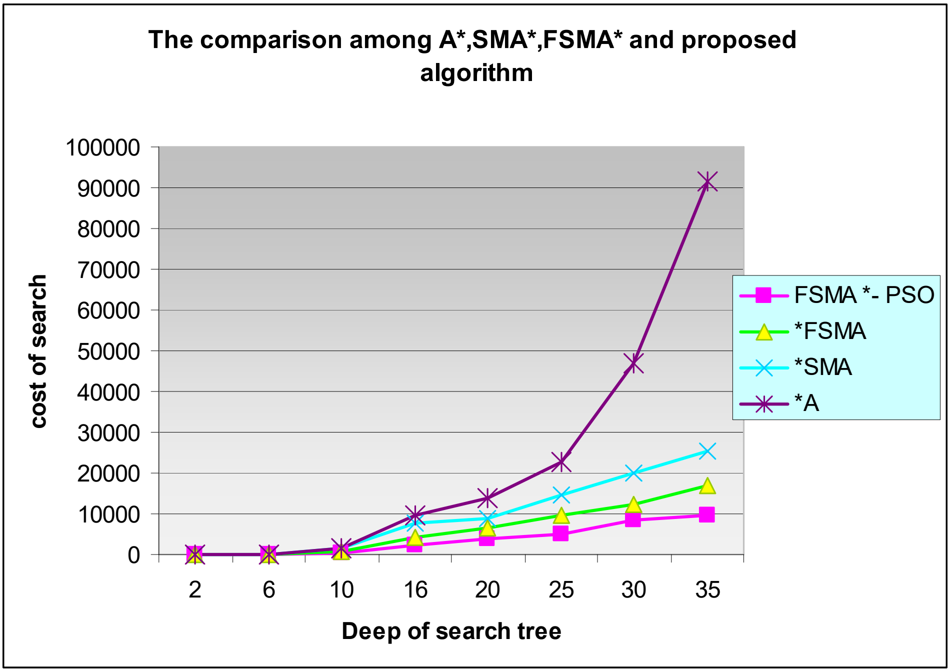
d	*Search cost (f(n))*					*Average branching factor*				
	IDS	*A**	*SMA**	*FSMA**	*FSMA*-PSO*	*IDS*	*A**	*SMA**	*FSMA**	*FSMA*-PSO*
2	15	9	9	7	7	7.71	6.54	6.54	6.54	5.69
6	7,850	128	128	115	105	7.56	6.16	5.42	5.29	5.47
10	18,569	1,580	1,580	784	548	7.12	5.67	5.23	5.02	5.11
16	5,894,156	9784	7,590	4,362	2,413	7.32	5.41	5.09	4.57	4.22
20	12,586,904	1,3852	9,015	6,519	3,809	7.02	5.02	4.86	4.31	4.06
25	-	22,549	14,521	9,751	5,079	-	4.59	4.64	4.16	3.22
30	-	46,750	19,843	12,470	8,542	-	4.38	4.19	3.94	2.54
35	-	91,586	25,410	16,843	9,661	-	4.08	3.82	3.62	2.38

Figure 16 includes useful knowledge about how to select coefficients of acceleration and inertia weight. Definitely, an inertia value of 1 can be the best candidate, and the worst average value of search cost is certainly related to the inertia weight 0.1.

The proposed method was tested on the problem of previous chapter to evaluate this method more accurately (Figure 14). The results were shown in Table 3. This method was compared with A*, SMA* and FSMA * methods. The search cost and the average branching factor parameters were studied to measure and compare the performance of these searches.

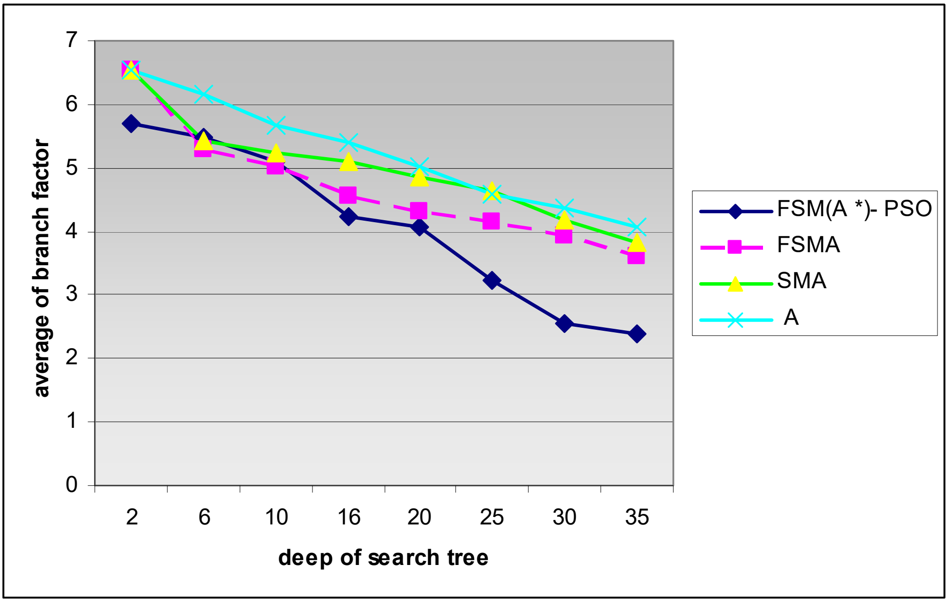
According to Table 3, the search cost at the beginning of the search was almost identical to that of FSMA*. For example, up to Depth 6, costs were similar. Although improvement was seen in comparison with the classical methods SMA* and A*, it was not significant. However, at Depth 10 onwards, search cost was reduced, and a significant difference was observed. Perhaps a better performance of the proposed algorithm would be observed in the branching factor of the state space tree because this method searches the paths that have higher probability to achieve a solution at the beginning of the search instead of searching additional spaces. As a result, the branching factor would have a lower value than other methods. Two line charts (1, 2) indicate the comparison between the mentioned methods with respect to search costs.

Figure 17 A comparison of methods A*, SMA*, and FSAM* with the proposed method with respect to search cost (see online version for colours)



As indicated in the above diagram, FSMA* had a good optimality trend from Depth 10 onwards; however, it is faced with the problem of falling in a local optimum point at Depth 25. But FSMA*-PSO solved this problem and indicated a very good and targeted behaviour from that stage.

Figure 18 A comparison of methods A*, SMA*, and FSAM* with the proposed method with respect to the average branching factor (see online version for colours)



5 Conclusions

The use of fuzzy logic in SMA* and defining a new function called unfavourability function improved the performance of SMA* search. However, this search had some weaknesses such as falling in local optimums, as shown in the branching factor chart. To solve this problem, a collective intelligence algorithm called PSO was used. Defining a number of particles in a certain radius, this algorithm would search the problem space, and then particles would converge around an optimum. This point could indicate the operating movement path. The results indicated improved search for the proposed algorithm in terms of search cost and the branching factor of search tree.

References

Anwary, A.R. (2008) 'Comparison of fuzzy BK-product and A* search algorithm for optimal path finding in unsupervised underwater environment', *International Journal of Systems Applications, Engineering & Development*, Vol. 2, No. 4, pp.57–63.

Ataul, B. et al. (2008) 'Clustering strategies for improving the lifetime of two-tiered sensor networks', *Comput. Commun.*, Vol. 31, No. 14, pp.3451–3459.

Bakdi, A., Hentout, A., Boutami, H., Maoudj, A., Hachour, O. and Bouzouia, B. (2017) 'Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control', *Robotics and Autonomous Systems*, Vol. 89, No. 1, pp.95–109.

Bi, J., Li, Z. and Wang, R. (2010) 'An ant colony optimization-based load balancing routing algorithm for wireless multimedia sensor networks', in *Proceedings of the 12th IEEE International Conference on Communication Technology* (*ICCT*), pp.584–587.

Boulmakoul, A. (2004) ‘Generalized path-finding algorithms on semi rings and the fuzzy shortest path problem’, *Journal of Computational and Applied Mathematics*, Vol. 162, No. 1, pp.263–272.

Cai, W., Jin, X., Zhang, Y., Chen, K. and Wang, R. (2006) ‘ACO based QoS routing algorithm for wireless sensor networks’, in *Proceedings of the 3rd International Conference on Ubiquitous Intelligence and Computing* (*UIC*), Vol. 4159, No. 1, pp.419–428.

Camilo, T., Carreto, C., Silva, J.S. and Boavida, F. (2006) ‘An energy efficient ant-based routing algorithm for wireless sensor networks’, *International Workshop on Ant Colony Optimization and Swarm Intelligence*, pp.49–59, Springer, Berlin, Heidelberg.

Chen, W., Li, C., Chiang, F. and Chao, H. (2007) ‘Jumping ant routing algorithm for sensor networks’, *Computer Communications*, Vol. 30, Nos. 14–15, pp.2892–2903.

Chengming, Q. (2011) ‘Application of improved discrete particle swarm optimization in logistics distribution routing problem’, *Procedia Engineering*, Vol. 15, pp.3673–3677.

Gerdelan, A., Iskandar, D., Djohar, A.F. and Reyes, N. (2006) ‘Utilizing the hybrid fuzzy A* algorithm in a cooperative multi-agent system’, in *6th International Conference on Hybrid Intelligent Systems* (*HIS ‘06*).

Gerdelan, A.P. and Reyes, N.H. (2006a) ‘Synthesizing adaptive navigational robot behaviors using a hybrid fuzzy A* approach’, *Advances in Soft Computing: Computational Intelligence: Theory and Applications*, Vol. 1, No. 1, pp.699–710, Springer, Berlin, Heidelberg.

Gerdelan, A.P. and Reyes, N.H. (2006b) ‘A novel hybrid fuzzy A* robot navigation system for target pursuit and obstacle avoidance’, in *Proceeding of the First Korean-New Zealand Joint Workshop on Advance of Computational Intelligence Methods and Application*, pp.75–79.

Ghasemaghaei, R, Rahman, M.A., Gueaieb, W. and El Saddik, A. (2007) ‘Ant colony-based reinforcement learning algorithm for routing in Wireless Sensor Networks’, In *Proceedings of IEEE Instrumentation and Measurement Technology Conference*, IMTC, pp.1–6.

Jaafar, J. and McKenzie, E. (2007) ‘Smaill, A, a fuzzy action selection method for virtual agent navigation in unknown virtual environments’, *FUZZ-IEEE 2007. IEEE International*, *Fuzzy Systems Conference*.

Karaman, O. and Temelta, H. (2005) ‘Navigation of mobile robots in unstructured environment using grid based fuzzy maps’, *International Conference on Fuzzy Systems and Knowledge Discovery*, pp.925–930.

Kemal, A. and Mohamed, Y. (2005) ‘A survey on routing protocols for wireless sensor networks’, *Ad Hoc Netw.*, Vol. 3, No. 3, pp.325–349.

Kuila, P. and Jana, P.K. (2014) ‘Energy efficient clustering and routing algorithms for wireless sensor networks: particle swarm optimization approach’, *Engineering Applications of Artificial Intelligence*, Vol. 33, No. 1, pp.127–140.

Kuila, P., Gupta, S.K. and Jana, P.K. (2013) ‘A novel evolutionary approach for load balanced clustering problem for wireless sensor networks’, *Swarm Evol. Comput.*, Vol. 12, No. 1, pp.48–56.

Li, K., Torres, C.E., Thomas, K., Rossi, L.F. and Shen, C-C. (2011) ‘Slime mold inspired routing protocols for wireless sensor networks’, *Swarm Intelligence*, Vol. 5,Nos. 3–4, pp.183–223.

Low, C.P. et al. (2008) ‘Efficient load-balanced clustering algorithms for wireless sensor networks’, *Comput. Commun.*, Vol. 31, No. 3, pp.750–759.

Mac, T.T., Copot, C., Tran, D.T. and De Keyser, R. (2017) ‘A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization’, *Applied Soft Computing*, Vol. 59, No. 1.

Neshat, M. (2013) ‘FAIPSO: fuzzy adaptive informed particle swarm optimization’, *Neural Computing and Applications*, Vol. 23, No. 1, pp.95–116.

Neshat, M. and Sepidname, G. (2015) ‘A new hybrid optimization method inspired from swarm intelligence: fuzzy adaptive swallow swarm optimization algorithm (FASSO)’, *Egyptian Informatics Journal*, Vol. 16, No. 3, pp.339–350.

Neshat, M., Sargolzaei, M., Masoumi, A. and Najaran, A. (2012) 'A new kind of PSO: predator particle swarm optimization', *International Journal on Smart Sensing & Intelligent Systems*, Vol. 5, No. 2, pp.521–539.

Neshat, M., Sepidnam, G. and Sargolzaei, M. (2013) 'Swallow swarm optimization algorithm: a new method to optimization', *Neural Computing and Applications*, Vol. 23, No. 2, pp.429–454.

Saleem, M. and Farooq, M. (2007) 'Beesensor: a bee-inspired power aware routing protocol for wireless sensor networks', in *Proceedings of EvoWorkshops* (*EvoCOMNET*), Vol. 4448, No. 1, pp.81–90.

Saleem, M. et al. (2011) 'Swarm intelligence based routing protocol for wireless sensor networks: survey and future directions', *Inf. Sci.*, Vol. 181, No. 20, pp.4597–4624.

Sarangi1, S. and Thankchan, B. (2012) 'A novel routing algorithm for wireless sensor network using particle swarm optimization', *IOSR Journal of Computer Engineering* (*IOSRJCE*), Vol. 4, No. 1, pp.26–30.

Soltani, A.R. and Fernando, T. (2004) 'A fuzzy based multi-objective path planning of construction sites', *Automation in Construction*, Vol. 13, No. 6, pp.717–734.

Stuart, J.R. and Norvig, P. (2003) *Artificial Intelligence A Modern Approach*, Prentice-Hall, Malaysia.

Tanasie, R., Tudor, C. and Dorian, C. (2007) 'A fuzzy path finding algorithm based on artificial potential fields', *Proceedings of the Second International Conference on Computer Graphics Theory and Applications*, Vol. AS-IE, Barcelona, Spain, 8–11 March 2007, pp.229–234.

Toofani, A. (2012) 'Solving routing problem using particle swarm optimization', *International Journal of Computer Applications*, Vol. 52, No. 18, pp.975–8887.

Yun, Y., Jo, J. and Gen, M. (2017) 'Adaptive hybrid genetic algorithm with modified cuckoo search for reliability optimization problem', in *Proceedings of the Tenth International Conference on Management Science and Engineering Management*, Springer, Singapore, pp.35–365.

Zhan, Z-H., Zhang, J., Li, Y. and Chung, H.S-H. (2012) 'Adaptive particle swarm optimization', *IEEE Transactions on Systems, Man, And Cybernetics – Part B: Cybernetics*, No. 6, pp.1362–1381

Zungeru, A.M., Ang, L-M. and Seng, K.P. (2012) 'Classical and swarm intelligence based routing protocols for wireless sensor networks: a survey and comparison', *J. Netw. Comput. Appl.*, Vol. 35, No. 5, pp.1508–1536.