J. Appl. Environ. Biol. Sci., 4(9S)210-216, 2014

ISSN: 2090-4274
Journal of Applied Environmental and Biological Sciences
www.textroad.com

Dynamic Route Planning using Hybrid (ACO-NPSO) Algorithm for Handling Multiple Constraints

Shahan Arshad[1], Kashif Zafar[2] and Asif Gilani[3]

Department of Computer Science, National University of Computer and Emerging Sciences, Lahore,

Received: September 12, 2014
Accepted: November 23, 2014

ABSTRACT

Route planning is an important challenge in time critical systems. It has been observed that the conventional route planning approaches are not suitable for real-time route planning which involves dynamic environment. Such methods have shown to perform well in static or known environments that do not involve satisfying multiple constraints. Efficient solutions based on swarm intelligence algorithms and genetic algorithms have been proposed for dynamic environment. In this research, we take up the route planning problem in dynamic environment, and present a hybrid (ACO-NPSO) algorithm which takes into consideration multiple constraints. The constraints to be satisfied are shortest distance and obstacle avoidance. This research uses NPSO with ACO algorithm, where NPSO is an improved version of basic PSO algorithm which controls the inertia weight of the PSO algorithm. The proposed hybrid algorithm unifies the features of above mentioned algorithms such that NPSO is employed to obtain optimized parametric values of ACO algorithm which are in turn used in ACO algorithm. A comparison of the proposed technique with other swarm intelligence Algorithms has also been performed. The result showed the effectiveness of the proposed algorithm as it provides better and shorter routes as compared to original ACO algorithm. Plus the result obtained from the proposed technique comes out much quickly as compared to ACO algorithm.

KEYWORDS: Dynamic route planning, ACO, NPSO, PSO, Inertia weight.

1 INTRODUCTION

Developing Time critical systems play a very important role in our lives. These systems include ambulance system, Navigation system, Unmanned Aerial Vehicle (UAV) system, Fire Evacuation system etc. Time needs to be saved in such systems and decision has to be made in a very short time. The optimization of time and other factors are needed to be done in order to find the correct solutions. The route planning is heavily involved in such problems. The Selection of path should satisfy or optimize the constraints. The path selection can be done manually if you know your environment very well or if the problem size is small. For large areas or city, path planning without using some navigation system is literally impossible. Because in such problems multiple constraints like Shortest travel time, Shortest Path, Easiness of driving, lesser traffic etc. gets involved .So efficient path planning solutions should satisfy such constraints and give efficient and effective routes in a timely manner. Path planning can be done with the help of static methods or Dynamic methods. Static methods like Dijktra, A* etc. provide much optimized solutions in Static environments. But these methods fail to deliver in dynamic environments. Path planning in Dynamic environments is a difficult task as compared to static or non–changing environments where problem size doesn't change or the requirements and constraints remain constant in number. In Dynamic environments the requirements are constantly changing and the problem size gets bigger and bigger. Due to these challenges the dynamic route planning is considered as a NP-Hard [1] problem.

Route planning in dynamic environment requires a robust and efficient technique which should satisfy the multiple constraints and provide solutions in time. Plus it should cater the dynamic nature of the environment size and parameters involved. A comprehensive analysis of the presently developed algorithms and methods with the proposed technique will be done in order to prove the feasibility of the proposed technique. In order to cater the constantly changing problem size and parameters in dynamic environment a robust and effective technique is proposed in this research. Both Static and Dynamic route planning is implemented in this research. In static environment the parameters are not changing and the location of source and destination remain fixed. In order to introduce dynamicity in the proposed method

*** Corresponding Author:** Kashif Zafar, Department of Computer Science, National University of Computer and Emerging Sciences, Lahore, kashif.zafar@nu.edu.pk

the location of the destination gets changed after running certain number of iterations. The Proposed method will also cater multiple constraints that are involved in the dynamic route planning. Finding the shortest distance between source and destination is based on Euclidean distance formula. Obstacles are generated randomly in the environment and the generated routes should avoid them. Algorithms like A* can be used in the route generation as an obstacle avoidance measure. The fitness of the routes gets reduced due to presence of the obstacles in them. Feasible routes are those which are not only shortest in distance but also avoid obstacles effectively and efficiently. The implementation will be done in a grid because in a grid environment route generation is simple and efficient. Plus in a grid environment it is easier to locate the obstacles. The whole problem area can be represented as a 2-dimensional grid. A hybrid swarm based approach is proposed in this research which basically utilizes the virtues of swarm intelligence algorithms. This hybrid algorithm or method will not only perform obstacle avoidance but will also provide feasible shortest routes between source and destination. There are approaches for routing problems [12] and optimization using machine learning techniques [13] and are performed well in such domains.

2. MATERIALS AND METHODS

2.1.Novel Particle Swarm Optimization Algorithm (NPSO)

Novel Particle Swarm Optimization (NPSO) algorithm [2] is an improved variant of PSO algorithm [7]. NPSO tries to solve to problem of local optima, convergence velocity and premature convergence that occurred in original Particle Swarm Optimization (PSO) algorithm and Linear Decreasing Weight Particle Swarm Optimization (LDWPSO) algorithm. NPSO dynamically updates the inertia weight which is dependent on fitness function and number of iterations set for the algorithm. PSO performance gets improved due to changing of inertia weight. The algorithm also changes the convergence speed of the particles.

$$w_i^{k+1} = w_{max} - (w_{max} - w_{min}) \times \left(\frac{iter}{iter_{max}}\right)^{0.5} \times \left(\frac{Fb^k}{Fit_i^k}\right)^m$$

Fig. 1. Inertia weight equation of NPSO algorithm

Fb^k is the global optimal solution and Fit_i^K is the optimal solution of the i^{th}-particle. Current iteration is denoted by iter and $iter_{max}$ is the maximum iteration set. m is a constant. It was experimentally determined that the value of m should be 2 in order to achieve optimal performance of algorithm. NPSO was evaluated with three test benchmark functions. The three test functions were Sphere function, Rastrigin function and Ackley function. All three Test functions were non-linear in nature. After the testing the results showed that the local optima problem of the algorithm gets resolved and Algorithm also enhances the particles to converge quickly and thus it increased the performance of the algorithm.

2.2Ant Colony Optimization Algorithm(ACO)

Ant Colony Algorithm gets its inspiration from swarm intelligence. Real life Ants are skilled in finding the shortest route between two points and show a tremendous cooperative behavior in search of food items. ACO tries to emulate their behavior and tries to find the feasible routes between two points. It also shows a fantastic quality of parallelism and gives good optimal routes. ACO [3] is an intelligent algorithm and it was proposed by Italian scholar Marco Dorigo. The first step involved in the ACO algorithm is the identification of links or edges. Then random generation of Ants is done and the selection of links or edges is based on the probabilistic objective function.

$$P_{ij} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{k \in S}[\tau_{ik}]^{\alpha}[\eta_{ik}]^{\beta}}$$

Fig. 2. Probabilistic objective function equation of ACO algorithm

Tij is the amount of pheromone on the edge.ηij =1/dij is a heuristic value. α and β are constants that determines the relative influences of the pheromone value and that of the heuristic value on the decision of the ant. When each ant completes a route or solution, local pheromone is updated based on the following equation:

$$\tau ij = \tau ij + \frac{Q}{L}$$

Fig. 3. Pheromone updating equation of ACO algorithm

The amount of pheromone added to such an edge is Q/L, where L is the length of the tour that was found and Q is a constant. But before this is done old pheromone is evaporated based on Eq.4

$$\tau ij = \rho \times \tau ij$$

Fig. 4. Pheromone evaporation equation of ACO algorithm

P is a constant whose value is usually less than 1.Pheromone evaporation is done in order to avoid old pheromone influencing the future selection of routes. After pheromone updation, the selection of global best solution is done. If the convergence criterion is met then the global best solution is considered as a final solution, else the steps mentioned above are repeated again.

2.3Proposed Solution

In the aim of this research is to solve the problem of route finding in a grid environment. The algorithm proposed in this research is a hybrid (ACO-NPSO) algorithm. The algorithm is a combination of Novel Particle Swarm Optimization (NPSO) algorithm and Ant Colony Optimization (ACO) [3].Finding shortest route between source and destination and obstacle avoidance are the constraints involved in this problem. NPSO is a variant of original PSO algorithm which as explained in [2] updates the inertia weight values. The inertia weight controls the difference between local best and global best solution. Reducing the inertia weight will bring the two closer to each other. In this way fast convergence of the particles will be obtained. Similarly it avoids the problem of local optima which occurs in the original PSO algorithm. The NPSO will be used to optimize the parameters of ACO algorithm which will be used in path planning in grid environment. The bitmap method will help in distinguishing between the area which is occupied by obstacles and those which are not. As in [4] the problem domain is divided into environmental modeling and Path planning. Environmental Modeling is done with the help of bitmap method. The Bitmap Method will help in identification of obstacles that are currently present in the environment by assigning 0 bit value to them. The 1-bit value will be assigned to that portion which is not occupied by the obstacles. The Path planning will be done by ACO.NPSO is used to optimize the parameters of ACO. In [4] PSO was used to optimize the parameters of ACO. This research will use NPSO which is a better variant of PSO as proposed by [2] in optimization of parameters of ACO. Firstly few iterations of ACO algorithm will be run on randomly set parameters. The values of ACO parameters are very important and play a vital role in the performance of ACO. The NPSO will be used to determine the values of parameters of ACO for which the NPSO will give the best optimal routes. These values of parameters will be used directly in ACO to achieve fast convergence. Fig.1 shows the flowchart of the proposed technique:

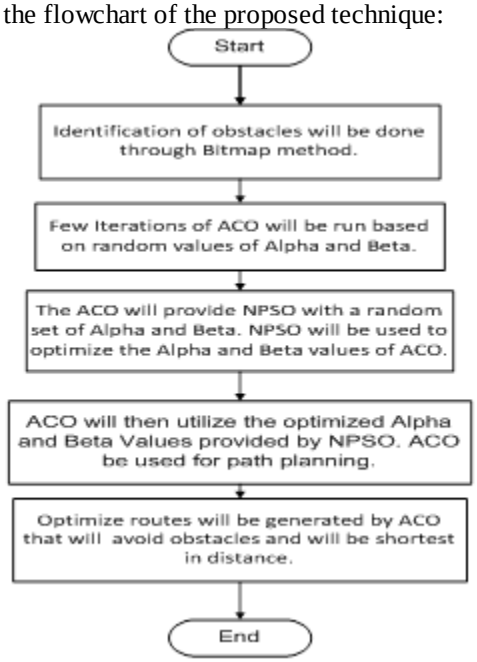


Fig.1 Flowchart of the NPSO-ACO Algorithm

The ACO is generally more applicable for those problems that demand crisp results and PSO is efficient for those problems that are fuzzy in nature. PSO computational complexity is very low and it can be easily applied in scientific research and engineering problems. But PSO suffers from the problem of local optima and slow convergence so that's why NPSO is used in this research which avoids such problems. ACO has the virtue of inherit parallelism and it is efficient to solve optimization problems. ACO convergence time is unpredictable but convergence is guaranteed [6].ACO as compared to GA takes less number of iterations in dynamic complex environment. Plus ACO takes less time to converge to a solution. Similarly the ACO easily adapts itself to complex, dynamic environments as compared to GA [5]. The hybrid Algorithm (ACO-NPSO) is proposed to take into account the qualities of ACO and NPSO algorithms and apply it to solve the path planning problem in a grid environment.

3. RESULTS

Novel Particle Swarm Optimization (NPSO) algorithm was implemented in java and number of experiments was conducted with different parameters settings. 25, 50, 75, 100, 125, 150, 175, 200, 300, 400 and 500 iterations were run. Particle size was set to be 10 and 20. Swarm Size varied from 10 to 100.The fitness function was based on the Euclidean distance between source and destination and obstacle avoidance. In order to bring dynamicity in the environment, the position of destination was randomly changed after 50 iterations of the algorithm. The obstacles are randomly generated in the environment. NetBeans IDN 7.2 is used to perform the implementation of the NPSO algorithm. The experimentation was done on Dell Inspiron-5520 Laptop with 1 TB Hard Disk and 8GB RAM. The optimal result usually comes out to be in small number of iteration with smaller swarm size. Table 1 shows the Routes generated by Novel PSO algorithm using different parameter settings.

Table 1:Routes generated by Novel PSO algorithm using different parameter settings

iterations	Particle Size	Swarm Size	Grid Size	fittness value	FSP1	FSP2	FSP3	FSP4	FSP5	FSP6	FSP7	FSP8	FSP9	FSP10
25	10	10	200*200	**304**	**1,180**	**8,140**	**24,149**	**60,124**	**61,103**	**86,86**	**135,70**	**164,56**	**172,20**	**200,1**
50	10	10	200*200	823	1,180	106,75	84,55	49,86	157,10	135,72	36,157	130,119	188,115	200,1
75	10	10	200*200	655	1,180	41,101	20,121	76,60	164,124	68,163	57,136	42,33	99,28	50,1
100	10	10	200*200	535	1,180	142,180	200,180	200,118	200,118	190,180	200,180	197,180	200,103	139,1
125	10	10	200*200	528	1,180	175,171	200,180	191,180	200,180	200,180	200,150	178,180	143,180	44,1
150	10	10	200*200	392	1,180	163,172	200,171	200,180	200,180	200,180	200,180	200,180	200,180	160,1
175	10	10	200*200	411	1,180	200,180	200,180	200,180	200,180	200,180	200,180	200,180	200,180	86,1
200	10	10	200*200	413	1,180	200,180	200,180	200,180	200,180	200,180	200,180	200,180	200,180	52,1
300	10	10	200*200	457	1,180	200,180	200,180	200,180	200,180	200,180	200,180	200,180	200,180	14,1
400	10	10	200*200	383	1,180	200,180	200,180	200,180	200,180	200,180	200,180	200,180	200,180	154,1
500	10	10	200*200	418	1,180	200,180	200,180	200,180	200,180	200,180	200,180	200,180	200,180	73,1

The ACO algorithm has been implemented in java. The IDE used was NetBeans 7.3.1. A Series of experimentation were performed in static and dynamic environment. In static environment the final destination did not changed and the overall environment remains the same. For dynamic environment, final destination changes randomly after certain number algorithm iterations. The Experimentation was conducted using different parameter settings. The grid sizes used were 20*20, 40*40 and 60*60. The algorithm iterations used were 5, 10, 20, 40, 60 and 100. The numbers of ants used in the algorithm were 5, 10, 15, 20 and 30. The Alpha and Beta values varied from 0.1 to 1. The min, max and Avg shows the minimum, maximum and average length found by the algorithm. Best solutions are those which are shortest in length. Some of the experimentation results are as follows

Table 2: Optimized Routes generated by ACO algorithm using alpha=1.0, Grid Size=20*20.

						Beta								
Iterations	Ant	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	1	Min	Max	Avg
5	5	64	73	78	85	96	93	77	89	108	69	64	108	83.2
10	10	55	61	63	70	53	83	83	76	54	57	53	83	65.5
20	20	62	59	65	58	69	60	80	55	65	49	49	80	62.2
40	30	51	59	42	49	50	64	51	43	47	61	42	64	51.7
60	30	48	59	46	50	63	49	63	50	56	49	46	63	53.3

Table 3: Optimized Routes generated by ACO algorithm using alpha=1.0, Grid Size=40*40.

						Beta								
Iterations	Ant	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	1	Min	Max	Avg
10	5	173	193	248	231	208	174	200	182	285	176	173	285	210.4444
10	10	205	163	155	188	164	170	198	166	204	130	130	205	179.2222
20	20	159	160	166	161	167	166	183	206	161	158	158	206	169.8889
40	30	174	177	178	176	177	194	190	190	182	186	174	194	182
60	30	188	187	191	189	197	199	197	200	198	194	187	200	194

Table 4: Optimized Routes generated by ACO algorithm using alpha=1.0, Grid Size=60*60.

						Beta								
Iterations	Ant	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	1	Min	Max	Avg
10	5	426	304	423	385	345	346	439	344	383	281	304	439	377.2222
10	10	299	254	293	332	380	395	323	340	348	266	254	395	329.3333
20	20	297	269	314	289	280	267	350	338	339	242	242	350	304.7778
40	30	268	223	322	312	317	290	342	344	346	268	223	346	307.1111
60	30	241	215	257	256	300	263	251	318	313	269	215	318	268.2222

The hybrid Algorithm is a combination of ACO and NPSO algorithm. Through NPSO algorithm the optimized values of alpha and beta are obtained. These optimized values are used in ACO algorithm to achieve optimized results. The grid sizes used in this experimentation are 20*20, 40*40 and 60*60. For creating dynamism in the algorithm the final destinations changes randomly after running a certain number of iterations. For Example for every 50 iterations set for the algorithm, the final destination changes after every 10 iterations of the algorithm. The Alpha and Beta values are obtained from NPSO algorithm to perform the experimentation. The iterations used are 5, 10, 20, 40, 60 and 100. The ants deployed for this experimentation are 5, 10, 20 and 30. The following tables show a 5 run average comparison of Hybrid (ACO-NPSO) performance with the ACO algorithm. The result showed the feasibility and effectiveness of the hybrid algorithm as better results are obtained using this hybrid approach as compared to using ACO algorithm for route planning.

Table 5: Grid Size=20*20, Alpha=0.3, Beta=0.3

Iterations	Ant	ACO	Hybrid(ACO-NPSO)
5	5	67.0	62.6
10	10	55.4	54.0
20	20	49.2	43.6
40	30	43.0	41.0
60	30	42.0	38.4
100	30	41.4	39.8

Table 6: Grid Size=40*40, Alpha=0.3, Beta=0.3

Iterations	Ant	ACO	Hybrid(ACO-NPSO)
5	5	240.2	224.0
10	10	172.2	162.2
20	20	188.0	147.6
40	30	180.2	141.8
60	30	161.4	132.8
100	30	148.2	120.6

Table 7: Grid Size=60*60, Alpha=0.3, Beta=0.3

Iterations	Ant	ACO	Hybrid(ACO-NPSO)
5	5	404.4	394.8
10	10	405.8	329.6
20	20	339.8	308.4
40	30	318.8	294.2
60	30	309.8	250.2
100	30	295.4	267.2

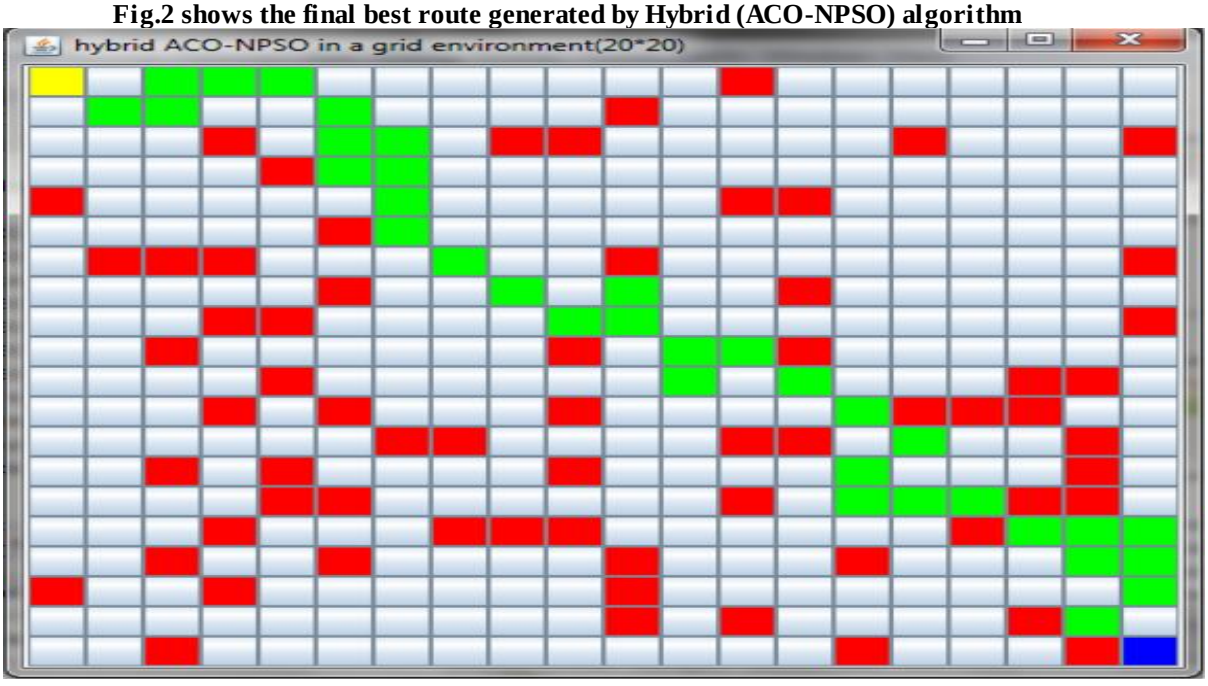
A comparison of other swarm Intelligence algorithms and Proposed hybrid (ACO-NPSO) has been performed. The Algorithms used in this experimentation are ACO, PSO, NPSO and Hybrid (ACO-NPSO). The grid size used for the Experimentation is 20*20.The obstacles were 5%, 10% and 20% of the problem area. For ACO and Hybrid (ACO-PSO) the alpha and beta values selected are 1.0 and 0.1 respectively. 100 iterations were run for each algorithm. The Best Route length is compared in all of these algorithms. The result of the experimentation is as follows:

Table 8: Performance Comparison of Swarm Intelligence Algorithms with Hybrid (ACO-NPSO) Algorithm

Obstacles	PSO	NPSO	ACO	Hybrid(ACO-NPSO)
5%	50	49	41	34
10%	52	51	44	36
20%	53	52	45	39

Fig.2 shows the final best route generated by Hybrid (ACO-NPSO) algorithm. 50 iterations of the algorithm were run. The alpha and beta used were 0.3 and 0.3 respectively. The red boxes are the obstacles that are present in the environment. The green boxes indicate the final path obtained from the algorithm. The yellow and blue boxes are the source and destination.

Fig.2 shows the final best route generated by Hybrid (ACO-NPSO) algorithm



4. CONCLUSION

A hybrid (ACO-NPSO) algorithm is proposed in this research. This hybrid Algorithm tries to resolve the dynamic route planning problem. The NPSO [2] is an improved version of PSO algorithm. PSO local optima problem is not present in NPSO. The ACO algorithm is well equipped to resolve the optimization problems. ACO and NPSO algorithm combine to form this hybrid Algorithm. The NPSO, ACO and Hybrid ACO-NPSO algorithms are implemented in java. The experimentation was carried out using different parameters settings. A comparison with other swarm intelligence algorithms with the proposed method is also performed. The result of the experiments proved the effectiveness of the proposed technique as proposed technique enabled ACO algorithm to perform better. This major benefit of using a hybrid algorithm (ACO-NPSO) is that it helps in setting the optimized values of alpha and beta parameters in ACO algorithm. As Parameter settings in ACO algorithm is very time consuming so by using this technique the optimized values of alpha and beta were quickly found out. So this technique can be used in all those areas where ACO algorithm is applied. This technique will help in rapid and quick route planning. Some of the potential areas or fields where this technique can be applied are Mine Detection, Fire Evacuation System, Courier Services, Ambulance Systems, Navigation Systems & Public Transport Systems.

5. REFERENCES

1. R. J. Szczerba *et al.*, Robust Algorithm for Real-Time Route Planning, in IEEE Transactions on Aerospace and Electronic Systems, vol. 36, July, 2000.
2. W. Dongyun *et al*., A Novel Particle Swarm Optimization Algorithm, in IEEE International Conference on Software Engineering and Service Sciences (ICSESS), pp. 408-411, 2010.
3. M. Dorigo and L.M. Gambardella, A cooperative learning approach to the traveling salesman problem, in IEEE Transactions on Evolutionary Computation,pp.53-66, 1997.
4. C.Shi *et al.*, Path Planning for Deep Sea Mining Robot Based on ACO-PSO Hybrid Algorithm, in International Conference on Intelligent Computation Technology and Automation, 2008.
5. F. K. Purian *et al.*, Comparing the Performance of Genetic Algorithm and Ant Colony Optimization Algorithm for Mobile Robot Path Planning in the Dynamic Environments with Different Complexities, in Journal of Academic and Applied Studies, vol. 3, pp. 29-44, February, 2013.
6. V. Selvi and R. Umarani, Comparative Analysis of Ant Colony and Particle Swarm Optimization Techniques,in International Journal of Computer Applications, vol. 5, August, 2010.
7. J. Kennedy and R. Eberhart, Particle swarm optimization, in Proceedings of the IEEE International Conference on Neural Networks, pp.1942-1948, 1995.
8. M. Dorigo et al., Ant Colony Optimization, in IEEE Computational Intelligence Magazine, pp.28-39, November, 2006.
9. K. Zafar, S. B. Qazi, A. R. Baig, Mine detection and route planning using frontier based exploration. In Proceedings of the IEEE International Conference on Engineering Semantic Agent Systems, IEEE-ESAS 2006
10.] K. Zafar, A. R. Baig, Optimization of Route Planning and Exploration using Multi Agent System, Multimedia Tools and Applications, Vol. 56, No. 2, pp. 245-265, Jan 2012
11. K. Zafar, A. R. Baig, H. Naveed, Mine Detection and Route Planning using Learning Real Time A* Algorithm, in Proceedings of the IEEE International Conference on Computer Science and its Applications, IEEE-CSA 2009
12. S. Ahmad, A. Shahzad, and N. A. Mian, A Comprehensive Study of Energy Efficient Routing in WSN towards QOS, VAWKUM Transaction on Computer Sciences, Vol. 3, No. 1, 2014
13. S. Goyal, G. N. K. Goyal, Machine Learning Cascade Algorithm for Analyzing Shelf Life of Processed Cheese, VAWKUM Transactions on Computer Sciences, Vol. 2, No. 1, 2013