Computing and Informatics, Vol. 32, 2013, 697–721

MULTIPLE ROUTE GENERATION USING SIMULATED NICHE BASED PARTICLE SWARM OPTIMIZATION

Kashif Zafar

National University of Computer & Emerging Sciences, Islamabad
e-mail: kashif.zafar@nu.edu.pk

Abdul Rauf Baig

Al Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia
e-mail: raufbaig@ccis.imamu.edu.sa

Communicated by Vladimír Kvasnička

Abstract. This research presents an optimization technique for multiple routes generation using simulated niche based particle swarm optimization for dynamic online route planning, optimization of the routes and proved to be an effective technique. It effectively deals with route planning in dynamic and unknown environments cluttered with obstacles and objects. A simulated niche based particle swarm optimization (SN-PSO) is proposed using modified particle swarm optimization algorithm for dealing with online route planning and is tested for randomly generated environments, obstacle ratio, grid sizes, and complex environments. The conventional techniques perform well in simple and less cluttered environments while their performance degrades with large and complex environments. The SN-PSO generates and optimizes multiple routes in complex and large environments with constraints. The traditional route optimization techniques focus on good solutions only and do not exploit the solution space completely. The SN-PSO is proved to be an efficient technique for providing safe, short, and feasible routes under dynamic constraints. The efficiency of the SN-PSO is tested in a mine field simulation with different environment configurations and successfully generates multiple feasible routes.

Keywords: Swarm, particle swarm optimization, swarm intelligence, route planning

1 INTRODUCTION

Particle swarm optimization [1] and ant colony optimization [1] are two major techniques in the family of swarm intelligence. In particle swarm optimization (PSO), a swarm of particles is placed in a hypothetical solution space with multiple constraints to satisfy. Each particle represents a solution in the solution space and has respective position and velocity in the solution space. The particle position and velocity update by using particle position and velocity updating equations. The particle size remains constant in a specific run of the system. On the other hand, each particle keeps track of its own best previous position so far that can be considered as simple nostalgia. The global best position and velocity of the particle guides the swarm for attaining optimum position and velocity. PSO is a population based technique and can be used effectively for route optimization problem. There are many approaches for solving route planning problem using A* [8], LRTA* [8], genetic algorithm [8], etc. The route planning problem requires robust heuristic algorithm to deal with the constraints of the route planning environment. The PSO is easy to implement and can be easily applied for optimization problems.

The route planning is an important problem in artificial intelligence research and has been addressed over the years; it is still considered to be a challenging area and requires efficient and robust technique to solve [3, 4, 5, 6, 41, 42, 43]. This paper presents simulated particle swarm for route planning. The physical existence of simulated particle swarm can be conceived in the form of a simulation that runs on computing devices. To say that they are autonomous computational entities implies that to some extent they have control over their behavior and can act without the intervention of other systems. The simulated particle swarm uses the modified PSO algorithm for finding optimum route between start and the goal. The SN-PSO has been tested for single route as well as multiple routes optimization and proved to be a robust technique for combinatorial optimization problems. We have tested the system for mine field simulation and obtained promising results. The major contributions are given below:

- Multiple route generation
- Route generation in complex environments
- Multiple peak optimization
- Optimization of the generated routes.

Section 2 describes the problem formulation, Section 3 gives the literature review, Section 4 presents the simulated niche based particle swarm optimization system, Section 5 presents the experiments and Section 6 presents the statistical results. Section 7 describes the methodology for route optimization and Section 8 presents the conclusion.

2 PROBLEM FORMULATION

Route planning and optimization of routes have been considered as NP complete problem [7]. Non-deterministic polynomial time complete is a complexity class of problems with two properties, i.e. any given solution to the problem can be verified quickly and if the problem can be solved in polynomial time then every problem in NP class can be solved. Such problems are usually tackled by approximation algorithms [7] and meta-heuristic approach is one of them.

An offline planner generates the complete plan before the task is performed [3]. The traditional offline planners often assume that environment is completely known and they try to find the plan based on shortest distance criteria. They cannot handle dynamic environments and are limited to their initial plan. When environment remains constant throughout the planning process, the route planning is called static offline route planning. A* [8], RTA* [9, 10, 11, 12, 13] are the algorithms that belong to this family. The planner is provided with the complete picture of the environment along with the starting and destination points. The destination point remains constant throughout the planning process. Static environments are those which, once discovered completely, do not change with respect to the environment components. The static environment can be known, partially known [9] or unknown and requires different approach for each category. On the other hand, an online planner generates a partial plan during the execution of the task. Online planners are mostly used for dynamic route planning. When the state space changes during the planning phase, the route planning is called dynamic online route planning [13]. Online planning approach is based on the assumption that the agents would be dealing with dynamic environment, and it would neither be feasible nor practical to re-plan the complete route. The online planners are mostly heuristic in nature as the problem instance reveals incrementally and is often prone to slow response time. They require sophisticated algorithmic approach to find the shortest route.

The unknown and dynamic environments are most difficult to deal with. There are a number of constraints to be considered during the planning phase. Obstacle avoidance and finding the shortest route are the complex tasks to deal with.

3 LITERATURE SURVEY

In addition to the modifications made to the basic PSO algorithm, a variety of PSO variations have also been developed. These include the sub-swarm based PSO algorithms and PSO with niching capabilities. The PSO variations in which grouping of particles into sub-swarms has been incorporated are called sub-swarm based PSO. These sub-swarms can either exist in cooperative or competitive mode [14]. Some examples of sub-swarm based PSO algorithms include the hybrid particle swarm optimizer with breeding and subpopulations by Lovberg et al. [15], Multi-phase PSO (MPPSO) by Al-Kazemi and Mohan [16], Life-cycle PSO (LCPSO) by Krink and Lovberg [17], Clustering based PSO with Stereotyping by Kennedy [18], Clustering based PSO by Thompson et al. [19], Cooperative Split PSO (CPSO-Sk) by Van den Bergh and Engelbrecht [20] and Predator-Prey PSO by Silva et al. [21].

Niching algorithms are those which are capable of locating multiple solutions to a problem. Niches can be defined as partitions of an environment which represent one solution to a problem. Speciation is the process of finding multiple niches or solutions. Species are the partitions of a population competing within an environment. They are the group of particles which converge on a single niche [14]. Some examples of niching algorithms include the Sequential Niching PSO employed by Kassabalidis et al. [22], PSO with Objective Function Stretching by Parsopoulous et al. [23] and nbest PSO by Brits et al. [24]. There are some PSO variants which are capable of finding multiple solutions to multi-modal problems by employing a sub-swarm based niching approach. These include the NichePSO presented by Brits et al. [25] and Species based PSO proposed by Parrott and Li [26].

N. C. Sahoo et al. [27] used the particle swarm optimization for finding the shortest path problem for network. This paper proposed a modified priority-based encoding incorporating a heuristic operator for reducing the possibility of loop formation in the path construction process for particle representation in PSO. They conducted experiments using 50–70 nodes network and found optimal paths, and proved to be better in performance then genetic algorithm.

Y. Hui et al. [29] presented a technique using niche PSO for planning multiple routes for air vehicles. The planning of multiple routes for air vehicles can be considered as a multiple-peak function optimization problem and can be solved by using niche based PSO. Each niche can be considered as a sub-population of PSO and optimizes single function. This strategy gives multiple routes for air vehicles. The complexity of the environment, particle size and size of sub-populations have direct impact on the convergence of niche based PSO. The parameters of PSO need to be optimized in order to reduce convergence time. They used fixed parameters for the size particles, number of particle dimensions and iterations. The experiments were conducted by using 1, 2 and 3 sub-populations and generate maximum of 3 routes, one from each sub-population.

L. F. Hsieh et al. [30] used PSO to schedule order picking routes in a distribution center. Order picking consumes 40 % of the operation time in a distribution center. The order picking requires optimization of storage strategy, order processing and route planning and it evolves into a multi objective optimization problem. They compared the results with ant system and generated initial solution by genetic algorithm. They first applied the genetic algorithm to find the initial solution and later on to help PSO in finding the optimal solution.

Mingquan et al. [44] used improved PSO algorithm for power distribution network expanding path optimization. They optimize power loss and investment cost of feeders simultaneously. They improved the parameter inertia weight and discrete idea is added so that it can easily run out the local optimum and provide high speed of convergence. Similarly, Chen et al. [45] and Zafar et al. [46] used Particle Swarm Optimization for dynamic route guidance. They used a simple simulation of network to find a shortest route. Instead of finding the best route, they are focusing on real time route.

4 SIMULATED NICHE BASED PARTICLE SWARM OPTIMIZATION

In single route optimization, the system tries to find a feasible and optimal solution for an entity to move from a start node to the goal node. In multiple routes optimization, more than one feasible and optimal routes are searched from all possible areas of the environment. It is a difficult job to find a safe, short, and feasible route in an environment with obstacles and optimization constraints. This research has applied particle swarm optimization algorithm for finding single feasible route and niche based particle swarm optimization algorithm for finding multiple feasible routes. This paper uses grid based environment representation with different environment configurations for experimentation and testing.

4.1 Single-Route Optimization

A simple PSO has been used for single route optimization problem. The solution space consists of particles that can be considered as initial population. Each particle represents a complete route from the start state to the goal state. The routes are categorized into feasible and infeasible routes based on the parameters like clearness, smoothness, distance, and cost. A route list is maintained for generated routes. The first solution in the route list represents the most feasible route and similarly the second solution represents the second to most feasible solution in the list. There is a possibility that we get a lower number of feasible solutions than we select in the solution count. The route list shows only the feasible solutions.

4.1.1 Simple PSO Algorithm

Let p be the total number of particles in the swarm. The best ever fitness value of a particle at design coordinates p_k^i is denoted by f_{best}^i and the best ever fitness value of the overall swarm at coordinates p_k^g by f_{best}^g. At the initialization time step $k = 0$, the particle velocities v_0^i are initialized to random values within the limits $0 \leq v_0 \leq v_0^{\max}$. The vector $v_0^{\max}$ is calculated as a fraction of the distance between the upper and lower bounds $v_0^{\max} = \zeta(x_{UB} - x_{LB})$ with $\zeta = 0.5$.

1. Initialize
   Set constants $k_{\max}$, c_1, c_2, w_0
   Randomly initialize particle positions $x_i 0 \in D$ in R^n for $i = 1, \ldots, p$
   Randomly initialize particle velocities $0 \leq v_0^i \leq v_0^{\max}$ for $i = 1, \ldots, p$
   Set $k = 1$
2. Optimize
   Evaluate f_k^i using design space coordinates x_k^i
   If $f_k^i \leq f_{\text{best}}^i$ then $f_{\text{best}}^i = f_k^i$, $p^i = x_k^i$

If $f_k^i \leq f_{\text{best}}^g$ then $f_{\text{best}}^g = f_k^i$, $p^g = x_k^i$
If stopping condition is satisfied then go to 3
Update particle velocity vector $v_k^i + 1$
Update particle position vector $x_k^i + 1$
Increment i. If $i > p$ then increment k, and set $i = 1$
Repeat the steps from 2
3. Results and Terminate

4.1.2 Particle Encoding

The particle represents a set of nodes as a path between two points on the map. Each particle comprises n nodes as shown in Figure 1, where n is user defined. The first node represents the start point and the last node represents the goal point, while the intermediate nodes are randomly selected from the map.

node(1)	node(2)	node(3)	node(4)	...	...	...	...	node(n)

Figure 1. Particle encoding

In Figure 2, an example of a particle is shown with $(2, 3)$ as the start node and $(36, 38)$ as the goal node for the value of n equal to 5. The intermediate nodes $(12, 22)$, $(9, 4)$ and $(22, 23)$ are randomly generated.

$(2, 3)$	$(12, 22)$	$(9, 4)$	$(22, 23)$	$(36, 38)$

Figure 2. Random initialization of a particle with $(2, 3)$ as start node and $(36, 38)$ as goal node

4.1.3 Fitness Function

For the evaluation of a route, the construction of complete path between the nodes is required. The A* algorithm as the shortest distance algorithm has been used for the construction of intermediate paths between the nodes. The nodes are generated randomly between the start point and the goal point and arranged in the same sequence as generated. After finding intermediate paths between all the nodes, a complete route is generated that can be evaluated by the fitness function. The fitness function of a particle p accommodates three optimization goals to minimize cost and is a linear combination of distance measure, number of obstacles, and number of mines as shown in Equation (1). Each obstacle and mine is considered as a blocked cell in the grid.

$$\text{fitness}_f(p) = w_d \cdot \text{distance}(p) + w_o \cdot \text{obstacles}(p) + w_m \cdot \text{mines}(p), \tag{1}$$

where the constants w_d, w_o and w_m represent the weights on the total cost of the path's length i.e. distance, presence of obstacles and mines, respectively.

The feasible route is considered as mine-free and obstacle-free, while infeasible route has mines and obstacles. The feasible route with minimum distance measure is considered as the global best. A path list is maintained for feasible routes and can be used for visualization of routes on the map based on user selection. The best of the best path will always remain on top of the path list. The fitness function evaluates each particle and assigns a particular value as fitness and divides the result into two components, i.e. feasible and infeasible routes. The feasible and infeasible routes are further ranked based on the values of the fitness. The infeasible routes are those that have at least one mine or at least one obstacle in its path. If there is an obstacle between two nodes, it will be considered as blocked cell or blocked cells. The obstacles are counted as the number of blocked cells in the particle and similarly each mine is evaluated as an individual cell and considered as a blocked cell.

4.2 Multiple-Route Optimization

For multiple route optimization, a niche based PSO has been used. It produces sub-swarms and behaves as self-organization of particles. Each sub-swarm acts independently and generates solutions to find the best solution. There is no exchange of information between different sub-swarms. Each sub-swarm maintains a niche, independent of other sub-swarms and acts as a stable swarm. It starts as one main swarm and gradually generates sub-swarms for multiple solutions. As the particle converges, a sub-swarm is generated by grouping particles that are close to the potential solution. This group of particles is then removed from the main swarm, and the process continues within their sub-swarm for the refinement of solution. The main swarm becomes disintegrated into sub-swarms gradually. When sub-swarms no longer improve the solutions they represent, it is considered as the convergence point. The global best position from each sub-swarm is taken as the optimum solution.

4.2.1 Niche Based PSO Algorithm

Create and initialize n_x-dimensional main swarm, S_j
repeat
    Train the main swarm, S, for one iteration using the cognition-only model;
    Update the fitness of each main swarm particle, $S \cdot x_i$;
for each sub-swarm S_k do
    Train sub-swarm particles, $S_k \cdot X_i$, using a full PSO model;
    Update each particle's fitness;
    Update the swarm radius $S_k \cdot R$;
endFor
    If possible, merge sub-swarms;
    Allow sub-swarms to absorb any particles from the main
        swarm that moved into the sub-swarm;

If possible, create new sub-swarms;
until stopping condition is true;
Return $S_k \cdot' Y$ for each sub-swarm S_k as a solution;

4.2.2 Explanation for Niche Based PSO Algorithm

The Niche based PSO [14] was developed to find multiple solutions to general multimodal problems. The basic operating principle of Niche based PSO is the self-organization of particles into independent sub-swarms. Each sub-swarm locates and maintains a niche. Information exchange is only within the boundaries of a sub-swarm. No information is exchanged between sub-swarms. This independency among sub-swarms allows sub-swarms to maintain niches. Each sub-swarm functions as a stable, individual swarm, evolving on its own, independent of individuals in other swarms. The Niche based PSO starts with one swarm, referred to as the main swarm, containing all particles. As soon as a particle converges on a potential solution, a sub-swarm is created by grouping together particles that are in close proximity to the potential solution. These particles are then removed from the main swarm, and continue within their sub-swarm to refine and to maintain the solution.

5 EXPERIMENTATION

The experimentation phase has been divided into two phases, i.e. single route optimization and multiple route optimizations. Two different environments have been used for experimentations, i.e. static environment and dynamic environment. We used four different size environment configurations, i.e. 20×20, 40×40, 60×60, and 80×80 grid maps as shown in Figure 3. Each grid map has different percentage of obstacles and mines. The starting and the ending points have been fixed for static experiments. After loading of the map, we need to set the parameters for experiment. There is an option for enabling dynamic environment by randomly changing the goal during online planning phase. We provide the number of such changing states for each run of the simulation and goal changes randomly during planning. After loading the map, we need to enter population size, cells count and other parameters for the experiment.

The first experiment deals with static environment using simple PSO as shown in Table 1 with a population size of 20. A high obstacle ratio and randomly generated maps have been used to simulate complex environments. The main parameters for the simulation are C1 and C2 values that are two constants that effect the weighted deviation from self-best position and global-best position. The SN-PSO is able to track the moving target similar to the online planning and takes care of the dynamic environment as shown in Table 3. During each run, the goal changes randomly and it behaves like a moving target search. Whenever the goal changes the SN-PSO is capable of tracking it and again finds the shortest route to the new goal. The number of generated paths is automatically generated in a drop down menu list. The niche based PSO has been applied for both static and dynamic environment

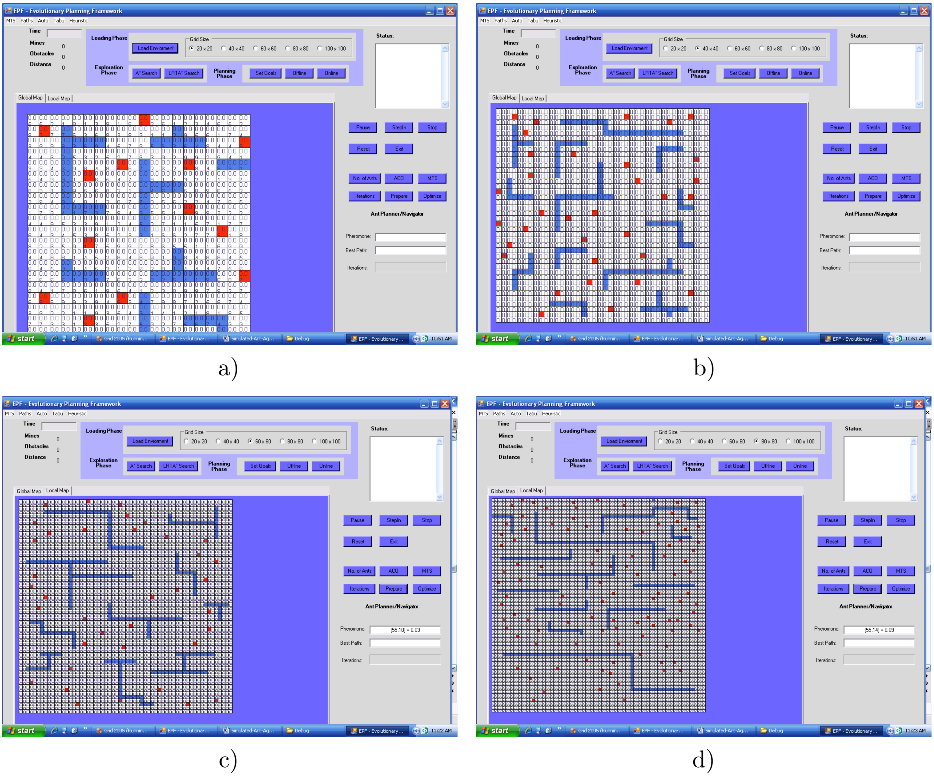


Figure 3. Grid maps of different sizes, obstacle ratio, and complexity: a) 20×20, b) 40×40, c) 60×60, d) 80×80

and successfully generates feasible routes. A different niche count has been used for experiments and results have been reported for comparison.

5.1 Parameter Settings for Simple PSO

The typical range is 20–40 particles. We have used a population size of 20 particles. It is determined by the problem to be optimized. For this problem, we have used 2 dimensions of particles. The range of particles is determined by the problem to be optimized; different ranges for different dimension of particles can be specified. We used one range for each dimension of the particle. The maximum change one particle can take during one iteration can be determined by $V_{\max}$. We have tested different values for $V_{\max}$. C_1 and C_2 are usually equal to 2. The stopping condition can be the maximum number of iterations the PS (Particle Swarm) executes and/or the maximum fitness achieved. This research used maximum number of iterations as the stopping condition.

5.2 Parameter Settings for Niche Based PSO Algorithm

If the population size is 20 and niche count is 4, then we have four sub-swarms each with a population of 5 particles. Each sub-swarm independently produces an optimized route; so for a population size of 20 with niche count 4, we can have 4 optimized routes. The population is initialized randomly with the selection of simple PSO as well as niche based PSO. The heuristic function used for static as well as dynamic environment is Manhattan distance [39] and Euclidean distance [39]. Finding the solution in dynamic environment is difficult when multiple constraints are enforced. The SN-PSO has been applied for dynamic environment in the same way as moving target search (MTS) [40, 41]. The simulated niche based particle swarm optimization is capable of locating the moving target by using modified PSO. Again swarms and sub-swarm have been used for handling of moving targets. After loading and preparation of the map, an option is selected for incorporating moving target. During the execution of the task, the goal changes and SNPSO reconfigures the planning phase for the new goal state. SN-PSO has been tested for different map sizes and configurations.

6 RESULTS

The experiments have been conducted separately for static and dynamic goal environment. Similarly, the experiments for single route optimization and multiple route optimizations are reported separately. The simple PSO has been used for static route planning and compared with classical optimization techniques along with SN-PSO as shown in Table 1. The performance of SN-PSO remains consistent with the increase in map size. The optimum values require an intermediate value of number of niches. Each experiment has been conducted 30 times and the average value for each 30 runs is reported. The maps are generated randomly with different number of mines, obstacle ratio and complexity. The red color cell symbolizes the mine, blue color cell represents the obstacle and white cell is the empty cell that can be traversed.

The graphs in Figures 4–7 show the cells traversed using simulated niche based particle swarm system for different particle size and map size. It can be observed in each of the tables that the route becomes shorter with moderate value of number of niches. The performance remains consistent throughout the simulation run. The experimental results for simple to complex environments have shown the scalability and robustness of the SN-PSO.

The SN-PSO has been applied to dynamic environment for presenting moving target search as shown in Table 3. After the selection for the number of times to change the goal state, the SN-PSO runs for tracing the first goal. After tracing the first goal with the shortest route the goal changes randomly to a new random configuration. The next goal appears within 10 % of the area of the previous goal. SN-PSO is capable of acquiring the moving goal with shortest route.

Number of Particles = 20	Comparison of Static Route Planning Optimization Strategies								
Map Size	20×20	20×20	40×40	40×40	60×60	60×60	80×80	80×80	STATIC
Case	1	2	3	4	5	6	7	8	
Initial State	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	
Goal State	$(3,18)$	$(6,15)$	$(1,15)$	$(1,14)$	$(0,18)$	$(3,11)$	$(2,15)$	$(6,18)$	
Manhattan Distance	21	21	16	15	18	14	17	24	
Cells traversed-A*	17	21	15	14	18	11	15	18	
Cells traversed-SPSO	20	25	17	16	23	12	16	19	
Cells traversed-ACO	20	24	17	16	23	12	16	21	
Cells traversed-SN-PSO	20	25	16	15	20	11	15	18	

Table 1. Comparison of SN-PSO with classical optimization techniques for static route planning

20×20 map																		
Start Point	$(0, 0)$																	
Goal Point	$(12, 10)$																	
Number of particles	10	10	10	20	20	20	20	30	30	30	30	40	40	40	40	100	100	100	100
Number of niche	2	5	10	4	5	10	20	2	3	5	10	2	4	8	10	2	4	10	20
Cells count by SNPSO	28	19	26	19	20	18	24	20	19	20	22	24	20	22	21	20	19	18	19
Cells count by A*	18																		
40x40 map																			
Start Point	$(0, 0)$																		
Goal Point	$(20, 25)$																		
Number of particles	10	10	10	20	20	20	20	30	30	30	30	40	40	40	40	100	100	100	100
Number of niche	2	5	10	4	5	10	20	2	3	5	10	2	4	8	10	2	4	10	20
Cells count by SNPSO	40	29	36	46	27	28	30	29	28	29	31	29	27	30	31	28	29	27	29
Cells count by A*	27																		
60x60 map																			
Start Point	$(0, 0)$																		
Goal Point	$(20, 35)$																		
Number of particles	10	10	10	20	20	20	20	30	30	30	30	40	40	40	40	100	100	100	100
Number of niche	2	5	10	4	5	10	20	2	3	5	10	2	4	8	10	2	4	10	20
Cells count by SNPSO	46	53	65	44	45	39	41	51	75	43	45	47	48	46	63	46	41	39	45
Cells count by A*	38																		
80x80 map																			
Start Point	$(0, 0)$																		
Goal Point	$(30, 70)$																		
Number of particles	10	10	10	20	20	20	20	30	30	30	30	40	40	40	40	100	100	100	100
Number of niche	2	5	10	4	5	10	20	2	3	5	10	2	4	8	10	2	4	10	20
Cells count by SNPSO	136	90	79	87	82	70	70	87	70	70	82	70	70	70	70	70	70	70	70
Cells count by A*	70																		

Table 2. Static & multiple route planning using simulated niche based particle swarm system

Number of Particles = 20 Number of Niches = 4	Dynamic Route Planning using Simulated Niche Based Particle Swarm System							
Map Size	20×20	20×20	40×40	40×40	60×60	60×60	80×80	80×80
Case	1	2	3	4	5	6	7	8
Initial State	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
Goal 1	$(3,18)$	$(6,15)$	$(1,15)$	$(1,14)$	$(0,18)$	$(3,11)$	$(2,15)$	$(6,18)$
Manhattan Distance	21	21	16	15	18	14	17	24
Cells traversed-SNPSO	18	23	15	15	20	11	15	19
Cells traversed-A*	17	21	15	14	18	11	15	18
Goal 2 random	$(4,15)$	$(4,16)$	$(2,17)$	$(4,13)$	$(1,17)$	$(2,10)$	$(4,13)$	$(5,16)$
Distance from Start	19	20	19	17	18	12	17	21
Distance from Goal 1	4	3	3	4	2	2	4	3
Cells traversed-SNPSO	21	19	18	15	18	10	13	18
Cells traversed-A*	21	17	17	13	18	10	13	17
Goal 3 random	$(1,14)$	$(3,15)$	$(4,15)$	$(2,10)$	$(2,18)$	$(1,7)$	$(3,11)$	$(8,13)$
Distance from Start	15	18	19	12	20	8	14	21
Distance from Goal 2	4	2	4	5	2	4	3	6
Cells traversed-SNPSO	15	17	20	12	18	7	12	17
Cells traversed-A*	14	17	16	10	16	7	11	13
Goal 4 random	$(6,17)$	$(7,13)$	$(1,12)$	$(1,9)$	$(4,17)$	$(3,4)$	$(2,9)$	$(5,11)$
Distance from Start	23	20	13	10	21	7	11	16
Distance from Goal 3	8	2	6	2	3	5	3	5
Cells traversed-SNPSO	21	17	14	9	18	6	12	12
Cells traversed-A*	20	15	12	9	17	4	9	11
Goal 5 random	$(3,16)$	$(4,11)$	$(2,9)$	$(2,7)$	$(5,15)$	$(6,1)$	$(1,8)$	$(3,9)$
Distance from Start	19	15	11	9	20	7	9	12
Distance from Goal 4	4	5	4	3	3	6	2	4
Cells traversed-SNPSO	19	27	9	7	22	6	9	14
Cells traversed-A*	18	24	9	7	17	6	8	9

Table 3. Dynamic route planning using simulated niche based particle swarm system

Normalized Error with A* Algorithm for Dynamic Route Planning using Simulated Niche Based Particle Swarm Optimization				
Map Size	Goal	SNPSO	A*	Error
20×20	$(3, 18)$	18	17	0.058
	$(4, 15)$	21	21	0.000
	$(1, 14)$	15	14	0.071
	$(6, 17)$	21	20	0.050
	$(3, 16)$	19	18	0.055
	$(6, 15)$	23	21	0.095
	$(4, 16)$	19	17	0.117
	$(3, 15)$	17	17	0.000
	$(7, 13)$	17	15	0.133
	$(4, 11)$	27	24	0.125
40×40	$(1, 15)$	15	15	0.000
	$(2, 17)$	18	17	0.058
	$(4, 15)$	20	16	0.250
	$(1, 12)$	14	12	0.166
	$(2, 9)$	9	9	0.000
	$(1, 14)$	15	14	0.071
	$(4, 13)$	15	13	0.153
	$(2, 10)$	12	10	0.200
	$(1, 9)$	9	9	0.000
	$(2, 7)$	7	7	0.000
60×60	$(0, 18)$	20	18	0.111
	$(1, 17)$	18	18	0.000
	$(2, 18)$	18	16	0.125
	$(4, 17)$	18	17	0.058
	$(5, 15)$	22	17	0.294
	$(3, 11)$	11	11	0.000
	$(2, 10)$	10	10	0.000
	$(1, 7)$	7	7	0.000
	$(3, 4)$	6	4	0.500
	$(6, 1)$	6	6	0.000
80×80	$(2, 15)$	15	15	0.000
	$(4, 13)$	13	13	0.000
	$(3, 11)$	12	11	0.090
	$(2, 9)$	12	9	0.333
	$(1, 8)$	9	8	0.125
	$(6, 18)$	19	18	0.055
	$(5, 16)$	18	17	0.058
	$(8, 13)$	17	13	0.307
	$(5, 11)$	12	11	0.090
	$(3, 9)$	14	9	0.555

Table 4. Normalized error with A* algorithm for dynamic route planning using simulated niche based particle swarm system

Number of Particles = 20 Number of Niches = 4	Multiple Route Planning using Niche Based Particle Swarm Optimization								
Map Size	20×20	20×20	40×40	40×40	60×60	60×60	80×80	80×80	STATIC
Case	1	2	3	4	5	6	7	8	
Initial State	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	
Goal State	$(3,18)$	$(6,15)$	$(1,15)$	$(1,14)$	$(0,18)$	$(3,11)$	$(2,15)$	$(6,18)$	
Manhattan Distance	21	21	16	15	18	14	17	24	
Route 1 using SN-PSO	20	25	17	16	23	12	15	19	
Route 2 using SN-PSO	22	26	19	19	27	15	17	21	

Table 5. Multiple route generation using niche based particle swarm optimization

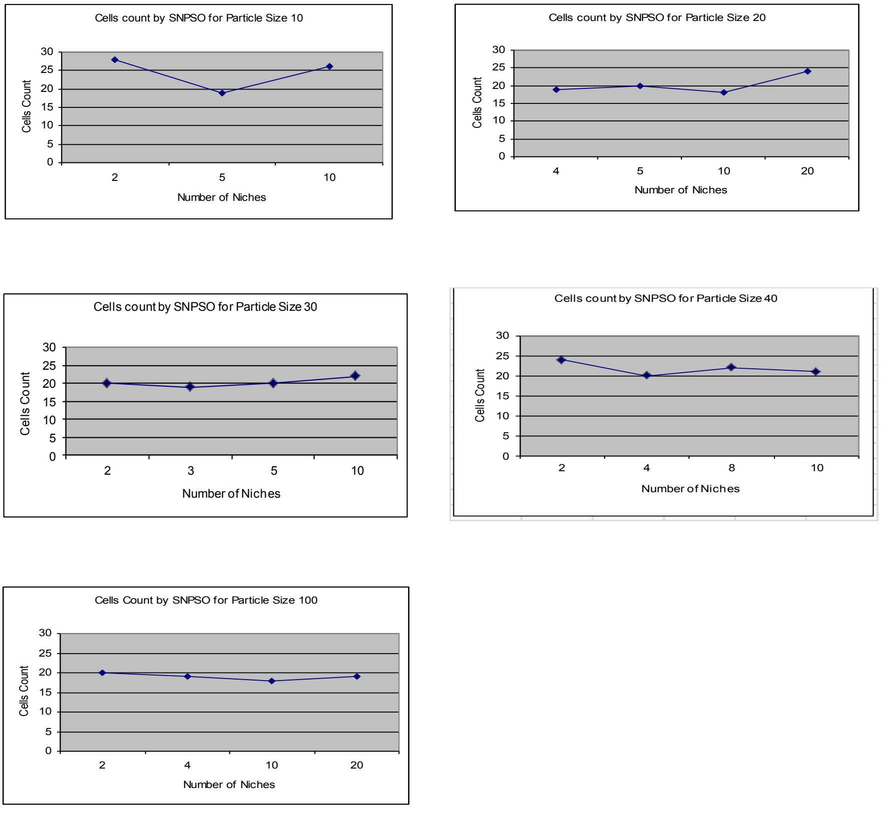


Figure 4. Route planning using SNPSO for map size 20×20

The SN-PSO is tested for static environment with multiple route generation and results are comparable with Manhattan distance as shown in Table 5. The performance of SN-PSO is better for complex and large size maps.

The SN-PSO has been compared with the methodology given in [29] and results from NS-PSO are better compared to their method as shown in Table 6 given below. Niche particle swarm optimization technology performs well for less complex environments and gives optimal peaks in each sub-population while SN-PSO has been tested for complex environments cluttered with obstacles and with different map sizes. The Niche particle swarm optimization technology was tested in the same size environment while SN-PSO has been tested for different environments with different obstacle ratios. The SN-PSO not only provides optimal peaks in each sub-swarm, it also provides the second best peak as well and gives more options for multiple routes. The SN-PSO is applied to the same application used in [29] with the same parameter settings and results have shown better performance as shown in Table 6.

Experiment	Sub-population	Generation of feasible route			Generation of approximate optimal route				
		Iterative times	N-PSO	SN-PSO	Iterative times	N-PSO	SN-PSO	Total consumed time	
			Time (s)	Time (s)		Time (s)	Time (s)	N-PSO	SN-PSO
1	1	16	0.804	**0.654**	51	2.851	2.952	**7.944**	**6.341**
	2	19	0.851	**0.738**	55	3.014	**2.800**		
2	2	22	1.422	**1.233**	57	3.525	**3.325**	**8.485**	**5.528**
	3	28	1.533	**1.342**	72	3.695	**3.230**		

Table 6. Comparison of SN-PSO with niche particle swarm optimization technology

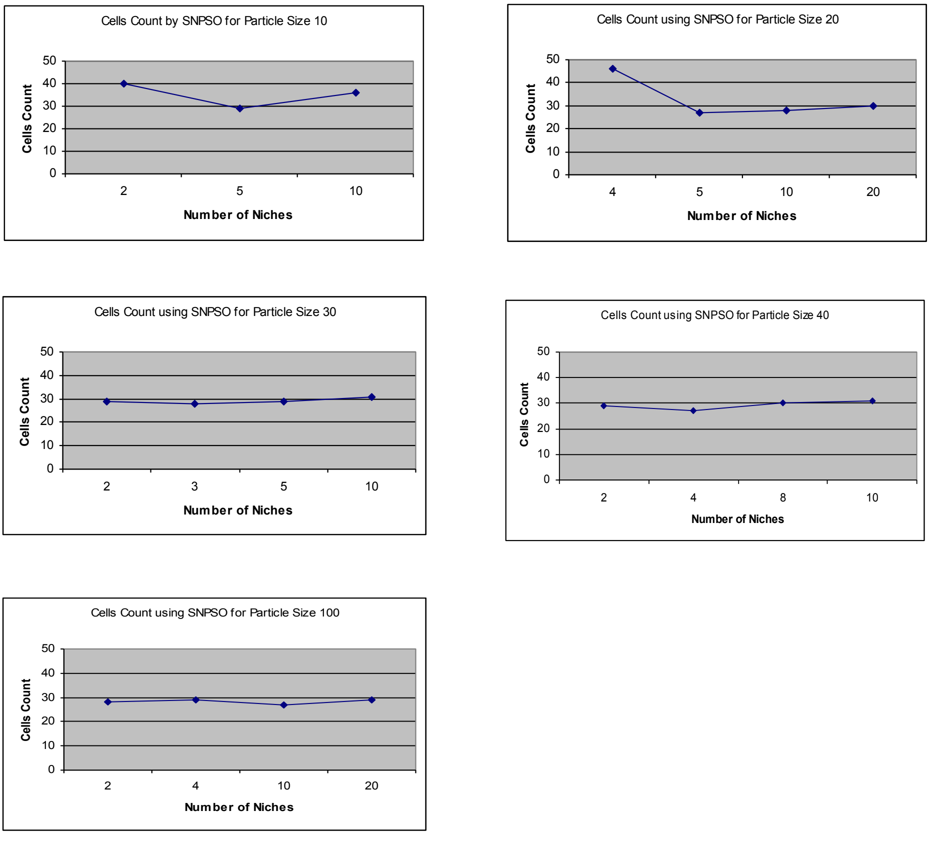


Figure 5. Route planning using SNPSO for map size 40×40

7 ROUTE PLANNING OPTIMIZATION

A separate module for route optimization has been implemented to repair the generated routes. The concept of v-edge has been used for the optimization of the route. The v-edge is the combination of diagonal v-shaped cells that are adjacent to each other but not in a straight line. Their positions can be changed without extending the distance or number of cells by only swapping their positions and the route can be repaired. The number of v-edges was counted in each generated route and fixed by taking the route with same number of cells. It does not reduce the number of cells traversed; it only increases the smoothness of the route. By replacing the v-edge with the adjacent cell, the route becomes smooth and straight. The route optimizer compares all the combinations of the v-edges along the route and removes the v-edges by replacing with diagonal cells along with keeping same cells and distance as shown in Figures 8 and 9.

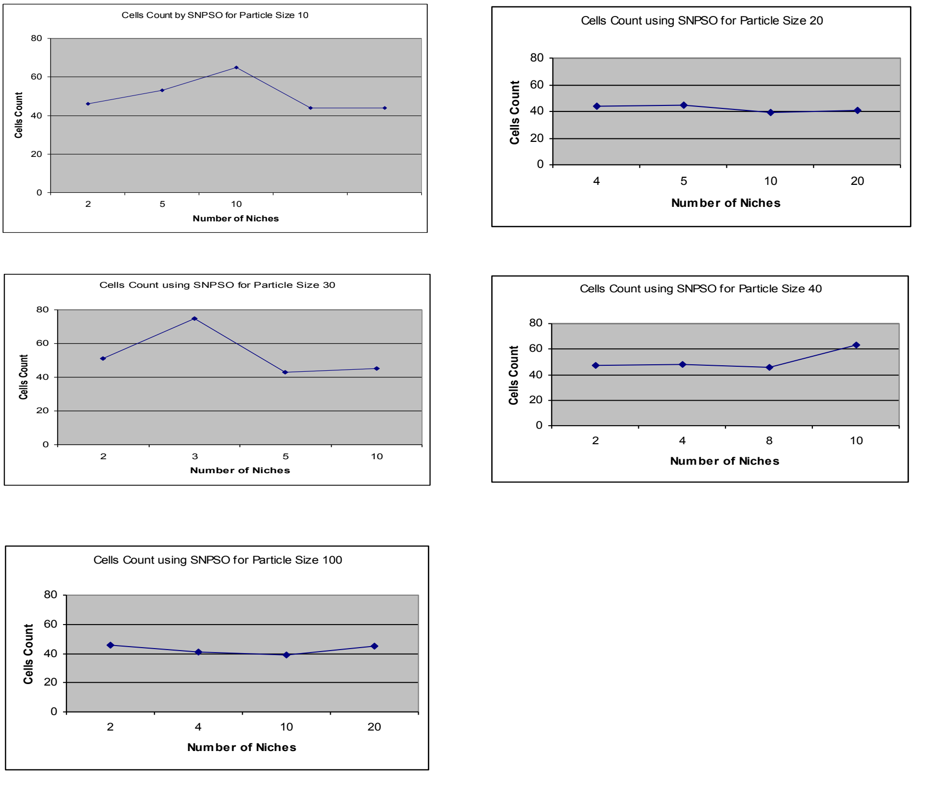


Figure 6. Route planning using SNPSO for map size 60×60

The diagonal movement generates v-edges that need to be repaired by using route optimizer. The smoothness of the routes depends on the number of v-edges and clearness of the route depends on obstacle avoidance and avoiding mines. The clearness of the route has been incorporated in the fitness function but the smoothness of the route requires a separate module. The main objective of this module is to repair the path without altering the original route plans. For this purpose, each generated route has been compared with the number of cells traversed for each of the repaired path and the resultant selected route will have the same number of cells and distance measure.

8 CONCLUSIONS

The simulated niche based particle swarm optimization has been used successfully for route planning and optimization of routes in static and dynamic environments. The SN-PSO uses online planning strategy for route generation and effectively deals with unknown environments. SN-PSO has been tested with simple to complex

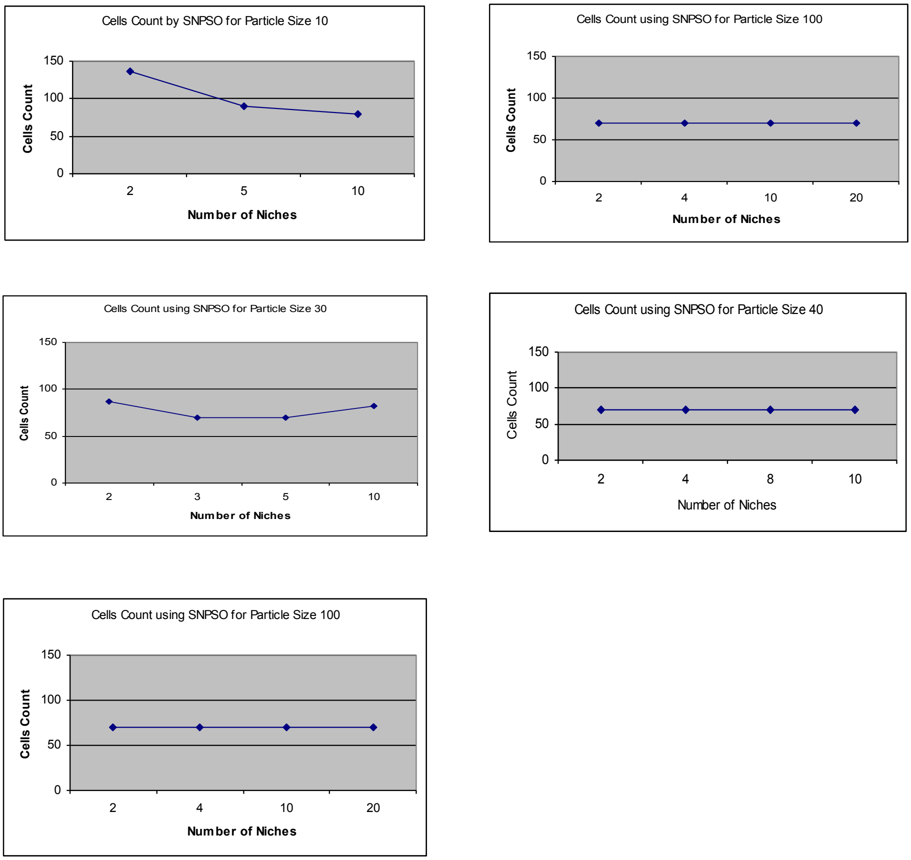


Figure 7. Route planning using SNPSO for map size 80×80

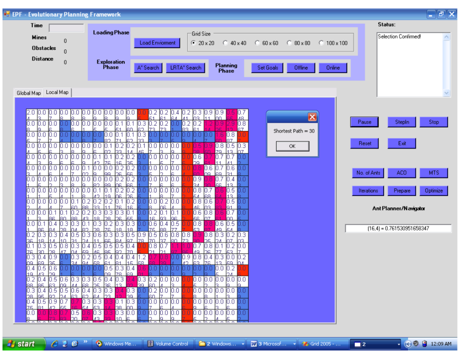


Figure 8. Shortest path without optimization

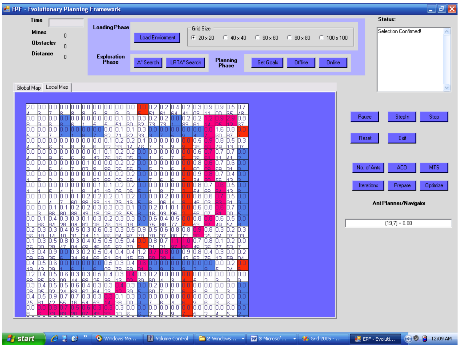


Figure 9. Shortest path with optimization

environments and found to be robust, efficient and scalable. The resultant routes have been classified by number of v-edges and unwanted curves. The route optimizer repairs the generated routes while keeping the number of cells and distance the same. The distances of the resultant routes have been compared with the A* algorithm and are found to be closer to algorithm A*. The SN-PSO successfully generates multiple feasible routes and has been implemented for mine detection and route optimization problem. The consistent experimental results with different size maps have shown the scalability and robustness of the system for handling dynamic environments. The SN-PSO can be further tested for other constraint satisfaction and multi-objective optimization problems in different application areas.

REFERENCES

[1] Engelbrecht, A. P.: Computational Intelligence; An introduction. John Wiley & Sons, ISBN 0-470-84870-7, 2002.

[2] Dorigo, M.—Socha, K.: An Introduction to Ant Colony Optimization. IRIDIA, Bruxelles, technical report series, 2007.

[3] Xiao, J.—Michalewicz, Z.—Zhang, L.—Trojanowski, K.: Adaptive Evolutionary Planner/Navigator for Mobile Robots. IEEE Transactions on Evolutionary Computation, Vol. 1, 1997, No. 1.

[4] Qiang, Z.—Xing, Z.: Global Path Planning Approach Based on Ant Colony Optimization Algorithm. LNCS, 2006.

[5] Vien, N. A.—Viet, N. H.: Obstacle Avoidance Path Planning for Mobile Robot Based on Ant-Q-Reinforcement Learning Algorithm. ISNN, LNCS, Springer-Verlag, 2007.

[6] Mei, H.—Tian, Y.—Zu, L.: An Hybrid Ant Colony Optimization Algorithm for Path Planning of Robot in Dynamic Environment. International Journal of Information Technology, Vol. 12, 2006, No. 3.

[7] Cormen, T. H.—Leiserson, C. E.—Rivest, R. L.—Stein, C.: Introduction to Algorithms. MIT Press, Cambridge, MA, ISBN-81-203-2141-3, 2001.

[8] Russell, S. J.—Norvig, P.: Artificial Intelligence – A Modern Approach. Prentice Hall, Englewood Cliffs, New Jersey, ISBN 0-13-103805-2, 1995.

[9] Stentz, A.: Optimal and Efficient Path Planning for Partially-Known Environments. In Proceedings of the IEEE International Conference on Robotics and Automation, May 1994.

[10] Korf, R. E.: Real-Time Heuristic Search: New results. In Proc. of the AAAI, 1988, p. 139–144.

[11] Korf, R. E.: Real-Time Heuristic Search. Artificial Intelligence, Vol. 42, 1990, No. 2-3, pp. 189–211.

[12] Korf, R. E.: Real-Time Heuristic Search: First results. In Proc. of the AAAI, 1987, p. 133–138.

[13] Korf, R. E.: Real-Time Heuristic Search. Artificial Intelligence, Vol. 42, 1990, No. 2-3, pp. 189–211.

[14] Engelbrecht, A. P.: Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, 2006.

[15] Lovberg, M.—Rasmussen, T. K—Krink, T.: Hybrid Particle Swarm Optimizer with Breeding and Subpopulations. In Proceedings of the Genetic and Evolutionary Computation Conference, 2001, pp. 469–476.

[16] Al-Kazemi, B.—Mohan, C. K.: Multi-Phase Discrete Particle Swarm Optimization. In Proceedings of the International Workshop on Frontiers in Evolutionary Algorithms, 2002, pp. 622—625.

[17] Krink, T.—Lovberg, M.: The Life Cycle Model: Combining Particle Swarm Optimization, Genetic Algorithms and Hill Climbers. In Proceedings of the Parallel Problem Solving from Nature Conference, Lecture Notes in Computer Science, Volume 2439, 2002, pp. 621–630.

[18] Kennedy, J.: Stereotyping: Improving Particle Swarm Performance with Cluster Analysis. In Proceedings of the IEEE Congress on Evolutionary Computation, Volume 2, July 2000, pp. 1507–1512.

[19] Thompson, B. B.—Marks, R. J—El-Sharkawai, M. A.—Fox, W. J.—Miyamoto, R. T.: Inversion of Neural Network Underwater Acoustic Model for Estimation of Bottom Parameters Using Modified Particle Swarm Optimizer. In Proceedings of the International Joint Conference on Neural Networks, 2003, pp. 1306.

[20] van den Bergh, F.—Engelbrecht, A. P.: Cooperative Learning in Neural Networks Using Particle Swarm Optimizers. South African Computer Journal, Vol. 26, 2000, pp. 84–90.

[21] Silvia, A.—Neves, A.—Costa, E.: An Empirical Comparison of Particle Swarm and Predator Prey Optimization. In Proceedings of the Thirteenth Irish Conference on Artificial Intelligence and Cognitive Science, Lecture Notes in Artificial Intelligence, Volume 2464, 2002, pp. 103–110.

[22] Kassabalidis, L. N.—Ei-Shurkawi, M. A.—Marks, R. J.—Moulin, L. S—Silvia, A. P.: Dynamic Security Border Identification Using Enhanced Particle

Swarm Optimization. IEEE Transactions on Power Systems, Vol. 17, August 2002, No. 3, pp. 723–729.

[23] Parsopoulos, K. E.—Plangianakos, V. P.—Magoulas, G. D.—Vrahatis, M. N.: Stretching Technique for Obtaining Global Minimizers Through Particle Swarm Optimization. In Proceedings of the IEEE Workshop on Particle Swarm Optimization, 2001, pp. 22–29.

[24] Brits, R.—Engelbrecht, A. P.—van den Berg, F.: Solving Systems of Unconstrained Equations Using Particle Swarm Optimization. In Proceedings of the IEEE Conference on Systems, Man, and Cybernetics, Volume 3, 2002, pp. 102–107.

[25] Brits, R.—Engelbrecht, A. P.—van den Berg, F.: A Niching Particle Swarm Optimizer. In Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning, 2002, pp. 692–696.

[26] Parrott, D.—Li, X.: Locating and Tracking Multiple Dynamic Optima by a Particle Swarm Model Using Speciation. IEEE Transactions on Evolutionary Computation, Vol. 10, 2006, No. 4, pp. 440–458.

[27] Mohemmed, A. W.—Sahoo, N. C.—Geok, T. K.: Solving Shortest Path Problem Using Particle Swarm Optimization. Applied Soft Computing, Elsevier, 2008, pp. 1643–1653.

[28] Li, S.—Sun, X.—XU, Y.: Particle Swarm Optimization for Route Planning for Unmanned Aerial Vehicles. In Proceedings of the IEEE International Conference on Information Acquisition, August 20–23, Weihai, Shandong, China, 2006.

[29] Hui, Y.—Zhong, Y.—Weihua, L.: Research on Planning Multiple Routes for Air Vehicles Based on Niche Particle Swarm Optimization Technology. In Proceedings of the IEEE International Conference on Computer Science and Software Engineering, 2008.

[30] Hsieh, L. F—Huang, C. J.—Huang, C. L.: Applying Particle Swarm Optimization to Schedule Order Picking Routes in a Distribution Center. Asian Journal of Management and Humanity Sciences, Vol. 1, 2007, No. 4, pp. 558–576.

[31] Bin, N.—Xiong, C.—Liming, Z.: Recurrent Neural Network for Robot Path Planning. PDCAT, LNCS, Springer-Verlag, 2004, pp. 188–191.

[32] Liu, S.—Tian, Y.: Multi Mobile Robot Path Planning Based on Genetic Algorithm. In Proceedings of the 5th World Congress on Intelligent Control and Automation, 2004, pp. 4706–4709.

[33] Beom, H. R.—Cho, H. S.: A Sensor-Based Navigation for Mobile Robot Using Fuzzy Logic and Reinforcement Learning. IEEE Transactions on SMC, Vol. 25, 1995, No. 3, pp. 464–477.

[34] Li, Z.—Chen, X.: A New Motion Planning Approach Based on Artificial Potential Field in Unknown Environment. PDCAT 2004, LNCS, Springer-Verlag, 2004, pp. 376–382.

[35] Howard, A.—Mataric, M. J.—Skhatme, G. S.: Mobile Sensor Network Deployment Using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem. 6th International Symposium on Distributed Autonomous Robotics Systems, 2002, pp. 299–308.

[36] Russell, S. J.—Norvig, P.: Artificial Intelligence – A Modern Approach. Prentice Hall, EnglewoodCliffs, NewJersy, ISBN0-13-103805-2, 1995.

[37] Li, Z.—Chen, X.: A New Motion Planning Approach Based on Artificial Potential Field in Unknown Environment. PDCAT 2004, LNCS, Springer-Verlag, 2004, pp. 376–382.

[38] Baig, A. R.—Rashid, M.—Zafar, K.: Niching with Sub-Swarm Based Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Information Theory and Engineering, Malaysia, ICITE, 2009.

[39] Russell, S. J.—Norvig, P.: Artificial Intelligence – A Modern Approach. Prentice Hall, Englewood Cliffs, New Jersy, ISBN 0-13-103805-2, 1995.

[40] Ishida, T.—Korf, R.: Moving Target Search: A Real-Time Search for Changing Goals. In Proceedings of the IEEE PAMI, Vol. 17, 1995, No. 6, pp. 609–619.

[41] Ishida, T.—Korf, R. E.: Moving Target Search. In Proceedings of the IJCAI, 1991, pp. 204–210.

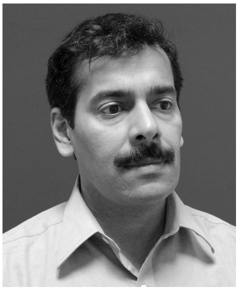
[42] Zafar, K.—Qazi, S. B.—Baig, A. R.: Mine Detection and Route Planning Using Frontier Based Exploration. In Proceedings of the IEEE International Conference on Engineering Semantic Agent Systems, IEEE-ESAS 2006.

[43] Zafar, K.—Baig, A. R.—Naveed, H.: 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.

[44] Mingquan, Z.—Xiaorong, C.—Huawei, M.—Chao, D.: Improved PSO Algorithm for Power Distribution Network Expanding Path Optimization. WISM, 2009, pp. 775–778, International Conference on Web Information Systems and Mining.

[45] Chen, Q.: Dynamic Route Guidance Method Based on Particle Swarm Optimization Algorithm. Proceedings of the Second International Conference on Intelligent Computation Technology and Automation, Vol. 1, 2009, pp. 267–270, ISBN: 978-0-7695-3804-4-1.

[46] Zafar, K.—Baig, A. R.: Optimization of Route Planning and Exploration Using Multi Agent System. Multimedia Tools and Applications, Multimedia Tools and Applications, Vol. 56, 2012, No. 2, pp. 245–265.



Abdul Rauf Baig received the B. Eng. degree in electrical engineering from the NED University of Engineering and Technology, Karachi, Pakistan, in 1987, the Diplóme de Spécialisation in computer science from Supélec, Rennes, France, in 1996, and the Ph. D. degree in computer science from the University of Rennes-I, France, in 2000. Since 2010, he has been a Professor with College of Computer and Information Sciences, Al Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia. Previously, he has been associated with National University of Computer and Emerging Sciences, Islamabad, Pakistan as a full time faculty member from 2001 to 2010. He has more than 100 publications in journals and international conferences. His current research interests are data mining and evolutionary algorithms.

Kashif Zafar is associated with National University of Computer and Emerging Sciences since 2004 in the Department of Computer Science. Prior to joining NUCES, he was a faculty member at GIKI in the Faculty of Computer Science & Engineering from 2002 till 2004; he specializes in artificial intelligence. He received his M. Sc. degree from The City University of New York, USA and Ph. D. degree from National University of Computer & Emerging Sciences, Islamabad. He has worked in the area of machine learning, swarm intelligence, computational intelligence, evolutionary algorithms, data mining, expert systems etc. He has been involved in teaching and research at undergraduate as well as graduate levels and has supervised numerous final year projects and theses. He has been awarded full scholarship from Foundation for the Advancement of Science and Technology, Islamabad for his postgraduate studies. He has published a number of research papers in international journals and conferences of high repute.