

A Hybrid Evolutionary Approach for Multi Robot Path Exploration Problem

K. S. Senthilkumar and K. K. Bharadwaj

Abstract - Robot Path Exploration problem or Robot Motion planning problem is one of the famous problems in robot's offline decision making algorithms. In this paper, a hybrid approach is presented that combines clustering and Genetic Algorithm (GA) to solve the Multi Robot Path Exploration Problem. The aim is to find collision free path, which Robot can follow to reach the target from its starting position. Environment is considered as a complete weighted graph representing the locations or points in the world environment and Traveling Salesman Problem (TSP) solving approach, based on GA is tried to solve this problem. Clustering is used to group the points (land marks) in the environment and rendezvous point is selected where all the robots finally meet. Experimental results are presented to illustrate the performance of the proposed scheme.

Index Term - Clustering techniques, Genetic Algorithms, Multi Robots, Rendezvous, Traveling Salesman.

I. INTRODUCTION

In many contexts, Multiple-Robot Exploration systems are faster or more powerful than a single Robot system. For Multiple Mobile Robots to be effective in real-world applications, more than one Robot must be able to safely share a potentially unknown workspace. The standard motion planning problem can be extended and generalized in many possible ways such as the cases when the geometry of environment is not fully known to the Robot. As a result, studies of the motion planning problems tend to make the use of many algorithmic techniques including intelligent and classical methods in computational geometry and problem modeling. However, there are difficulties associated with the use of such multi-robot systems. Task allocation, synchronization and coordination are significant problems, as is maximizing the efficiency of the distributed team. In applications where one Autonomous Robot is good, more than one Robot is likely to be even better is shown in [3], [4]. Multiple mobile Robots can carry out tasks which are inherently distributed in location, time, and/or functionality. Multiple Robots provide a distributed system with all the advantages inherent such as, fault tolerance, redundancy, and reliability.

Manuscript received November 21, 2007 A Hybrid Evolutionary Approach for Multi Robot Path Exploration Problem.

K. S. Senthilkumar is with the School of Computer & System Sciences, Jawaharlal Nehru University, New Delhi 110067 (e-mail: ksskumar16@yahoo.com).

K. K. Bharadwaj is with School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi 110067 (e-mail: kbharadwaj@gmail.com).

For small numbers of Robots and goals, it is feasible to search for the optimal tour within the mission context in a short period of time. As the size of the problem gets larger, it is necessary to rely on alternative search methods whose computational complexity does not grow nonlinearly.

To solve the Multi Robot Exploration problem, the environment is considered as a complete weighted graph representing the landmarks or points in the world environment. Robot has to go through these points only once. This idea imitates the well known Traveling Salesmen Problem (TSP). We consider the Multi-Robot Path Exploration problem as MTSP type of problem. In this paper the Robot should explore all the points (land marks) in the network by using a shortest path. The points in the environment are divided into clusters using clustering technique and each cluster is assigned to one Robot.

The number of cluster is same as the number of Robots participated in the exploration. Each Robot should visit all the points in the cluster and meet at the rendezvous point in the environment. The intended application is collaborative map exploration. In the proposed scheme a Multi Robot path exploration problem is mapped to Multiple Traveling Salesmen Problem (MTSP). We are very much concern to eliminate the redundancy in the initial and evolved populations. Here we used the chromosomes to represent the possible shortest paths. In this approach, each robot's paths are represented in separate multiple populations and evolves in parallel.

The remainder of the paper is organized as follows: section 2 gives background about the methods that are used in our Hybrid scheme. The next section discusses the problem and the proposed approach. Section 4 describes the experimental results. Finally the paper is concluded in section 5.

II. BACKGROUND

A. Genetic Algorithms

Genetic Algorithm (GA) is a search procedure motivated by the mechanisms of biological evolution. It maintains a population of strings, called chromosomes that encode candidate solutions to a problem. The population of strings is randomly initialized giving a diverse range of possible solutions. Each of this solution is evaluated and given a fitness score. At this point the population is examined to see if a suitable solution has been found or improvement has slowed to such an extent that is not worth searching further. This could be stopped when, a given goal has been reached or a certain level of improvement has not been achieved over a fixed number of generations.

The key to finding a good solution using a GA lies in developing a good chromosome representation of solutions to the problem [10]. A well-designed chromosome should reduce or eliminate redundant chromosomes from the population and allows the GA operators to work effectively to generate better solutions as the iterative evolutionary process continues. Redundancy in the chromosome representation refers to a solution being able to be represented in more than one way and appearing in the population multiple times. These multiple representations increase the search space and slow down the search.

Solving the TSP using GAs has generated a great deal of research focused on how best to perform the action of evolving an optimal solution to the problem [5], [7], [9] and [10]. TSP is NP-hard but has many real world applications and a vast amount of research [1], [9] exists into methods for solving the TSP via randomized search techniques including Genetic Algorithms, Neural Nets, and Simulated Annealing. The TSP is a classic combinatorial optimization problem, which is simple to state but very difficult to solve.

B. Rendezvous

The Rendezvous search problem [8] for mobile Robots is a search optimization problem based on the following question: How should two mobile Robots move along the nodes of a network in order to minimize the time required to meet? In the simplest, idealized, noise-free case, the robots have a pre-arranged notion of what constitutes a good rendezvous point. At a pre-arranged time, the Robots go to the best rendezvous point, and wait for the other Robot(s) to arrive. In the context of cultural environments, typical notions of good rendezvous locations [8], we refer to these points as landmarks – generally rely upon some *a priori* knowledge of the environment. Multi-robot systems also have an inherent need for inter-robot rendezvous. The ability to meet facilitates localization, allows collaborative map learning [3], [4]. We try the problem of rendezvous between two robots collaborating in learning the path of an unknown environment. That is, how can two autonomous exploring Robots that cannot communicate with one another over long distances meet if they start at different locations in the environment?

C. Clustering

Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. But in our problem we group all the points in the environment, which are distance wise close to some predefined points. The number of predefined points is same as the number of Robot participated in the exploration. Clustering heuristic methods [1], [2], [6], and [10] has been used by many researchers in Multi Robot exploration. We introduce a clustering heuristic based procedure for Multi-Robot Path Exploration. This technique contributes to the improvement of the path planning solutions. The method of clustering subsets of landmarks and connecting the subsets may not necessarily yield optimal results, but the savings in computational time and space could be significant enough to offset this minor inconvenience in a time critical scenario.

III. PROBLEM SPECIFICATION & APPROACH

In the simplest form, the Robot motion planning problem can be defined as follows: Let R be a Robot system and it is free to move in two dimensional spaces, denoted by W . The space W contains land marks and some obstacles, which their position and geometry characteristics are all exactly or approximately known. Now, the path planning strategy for R is to take the initial position S , a desired final position T and a set of obstacle characteristics that is presented in the robot's environment and generates an appropriate path between S and T . The environment is represented by networking of points such as land marks and obstacles. The Robots are supposed to visit those land marks avoiding the obstacle points. We design our scheme to work in a Multi-Robot path Exploration environment. So we have to assign the points to each and every Robots participating in the exploration task. We used clustering method to group the points using the parameter distance and assign them to each Robot. Robots have to find the shortest path inside their assigned clusters to reach the rendezvous point in the environment. We defined the rendezvous point as the 'Superpoint' in our algorithm.

First we find the merged point as a *Superpoint* by calculating the centroid of the all points. For example, if $p(x_1, y_1) \dots p(x_m, y_m)$ are m points defined in the domain, the *Superpoint* S is defined by $((x_1 + \dots + x_m)/m, (y_1 + \dots + y_m)/m)$. The distance between any two points $p(x_1, y_1)$ and $q(x_2, y_2)$ can be defined as Euclidian distance d . The Robot starts from a specified point and visits each point, and reach the rendezvous point. We experiment the algorithm for two Robots. Since there are two robots, we divide the points into two clusters. Clustering example is shown in the Figure 1.

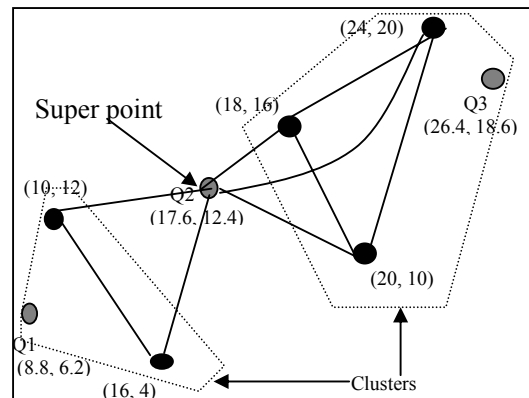


Figure 1: Cluster making

Calculations:

Superpoint (Q2)

$$x = \frac{(10 + 18 + 16 + 20 + 24)}{5} = 17.6$$

$$y = \frac{(12 + 16 + 4 + 10 + 20)}{5} = 12.4$$

$$(Q1) \quad x = \frac{17.6}{2} = 8.8$$

$$y = \frac{12.4}{2} = 6.2$$

(Q3)

$$x = 17.6 + 8.8 = 26.4$$

$$y = 12.4 + 6.2 = 18.6$$

Clusters are made by using points Q1 and Q2. The points close to Q1 is assigned to one cluster and others assigned to the other cluster.

The chromosome represents the points in the path. The size of the chromosome is equal to the points in the cluster. Since there are two Robots, we create two populations for each Robot. The chromosome format is as shown in figure2. Genes represents the decimal value assign to each point or nodes in the environment. First gene represents the starting location of the Robot and last gene represents the rendezvous point. For example a tour sequence is: 1→4→3→6→2→7→8→9. This is represented in the chromosome as follows:

1	4	3	6	2	7	8	9
---	---	---	---	---	---	---	---

Figure 2: Chromosome format

The initial populations are created using random values. A Chromosome containing repetitive values or obstacle values is treated as infeasible path and such chromosomes are rejected during initial population generation, application of crossover and mutation operator. Objective function or fitness function is the distance between the points (N) in the path represented by the chromosome. Chromosome is checked that the first gene in the chromosome is the start node and last gene is target node and the other genes represent all the nodes in the cluster in some order. It is decided for the testing that the two robots were started at different locations and reached the rendezvous point using a shortest path visiting the entire nodes in their cluster. Fitness function (f) is calculated using the following formula:

$$f = \sum_{i=1}^{N-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \quad (1)$$

Here N is the number of points in the environment or cluster.

A. Proposed Genetic Algorithm

1. Create an initial population of P chromosomes for both robots (Generation 0).
2. Evaluate the fitness of each chromosome in both populations.
3. Select the best two parent chromosomes from the current population.
4. Exchange bit strings with the uniform order based crossover to create two offspring.
5. Process each offspring by the exchange mutation operator, and evaluate them.
6. Replace the worst parent with the best offspring out of the two offsprings in the population.

7. Go back to Step 3 if the number of generations is less than some upper bound. Otherwise, the final result is the best chromosome created during the search.

The above algorithm introduces concepts, like the uniform order based operator to exchange bit strings, and the exchange mutation operator. Since the order of the node should be precise and nodes can be exist only once in the chromosome, we suggested these concepts. These concepts are now defined more precisely.

B. Order based Crossover

The Order based Crossover operator is aimed at exchanging bit strings between two parent chromosomes. A random string is created using 1 and 0; the size of the string is the length of the chromosome. Using the random string the parents' genes were exchanged as shown below.

Parent 1	1	2	3	4	5	6	7	8
Parent 2	8	2	6	4	7	3	5	1
String	1	0	0	1	1	0	1	1

Offspring 1	1	2	6	4	5	3	7	8
Offspring 2	1	2	6	4	5	3	7	8

Figure 3: Order Based Crossover

Child 1: if string bit contains 1, then reproduce the parent 1's gene, else place the values in the order according to parent 2.

Child 2: if string bit contains 0, then reproduce the parent 2's gene, else place the values in the order according to parent 1.

A probability is associated to the application of the crossover operator. In this way, good chromosomes can be preserved from one generation to the next. High crossover rates create more new offspring, at the risk of losing many good chromosomes in the current population. Conversely, low crossover rates tend to preserve the good chromosomes from one generation to the next, via a more conservative exploration of the search space.

C. Exchange Mutation

The exchange mutation operator randomly selects two nodes in the path and exchanges them. It is also called as swap mutation operator. A probability is associated to the application of the mutation operator. For example consider the path represented by

Offspring 1	1	2	6	4	5	3	7	8
Offspring 1	1	4	6	2	5	3	7	8

Figure 4: Exchange Mutation

IV. EXPERIMENTAL RESULTS

We have considered two different cases. In the first case one robot was used to explore the collision free shortest path by visiting all the points or landmarks in the environment by avoiding the obstacle points. In the second case two robots were used. Landmarks in the environment were divided into two clusters according to the Euclidean distance between landmarks. The robots starting points are defined and rendezvous point was the super point in the environment, since it is the centroid to all the point in the exploration environment. The following GA parameters were used in both experiments:

Case I: Single Robot

Chromosome size:	30
Population size:	20
Crossover probability:	0.8
Mutation Probability:	0.01
Number of Generations:	40
Start point:	1
Target point:	17

Case II: Two Robots

Chromosome sizes:	14 & 17
Population size:	20 each
Crossover probability:	0.8
Mutation Probability:	0.01
Number of Generations:	20
Start points:	1 & 16
Rendezvous point:	11 & 6

Experiments were done for different initial populations, and we tabled the best paths in Table 3. A Single Robot covers all 30 identified landmarks in the environment. To get an optimized result, we performed several simulation experiments. The fitness achieved for two Robots in the same environment in a clustered manner, are tabled in Table 1 & 2. The best out of the shortest path's fitness is 150.74, for a single robot and the path is as follows:

1→3→6→10→19→20→21→27→28→22→25→23→
 26→29→24→15→8→12→16→18→14→7→13→11
 →9→4→2→5→0→17.

The best out of the shortest path's fitness for Robot no. 1 is 61.79 and the path is as follows:

1→2→4→5→7→10→13→12→8→6→3→0→9→11

The best out of shortest path's fitness for robot no.2 is 78.83 and the path is as follows:

16→13→12→10→7→4→5→9→11→15→14→8→2→3→
 0→1→6

The rendezvous point for the two robots is at point 17. We assumed that the robots are homogeneous and the speed is constant. We used a 20 × 20 grid environment as depicted in Figure 5. The Screen shoot of the clustered environment is shown in figure 7. The shortest paths for a single Robot and Multiple Robots are shown in figure 6 and 8 respectively. By these results, to cover the area by single Robot took 150.74 km run and by two Robots maximum distance covered by one robot is 79.32 km. So this is clearly indicates that multi robots cover the whole area in lesser time period than a single Robot.

Paths for cluster 1	Fitness
1 4 2 5 7 9 10 13 12 8 6 3 0 11	62.13
1 5 4 2 0 3 6 9 7 13 12 8 10 11	66.84
1 5 2 4 7 9 10 13 12 8 6 3 0 11	62.14
1 5 2 4 7 12 13 10 6 3 0 8 9 11	67.72
1 2 4 5 7 10 13 12 8 6 3 0 9 11	61.79
1 2 4 5 9 10 13 12 8 6 0 3 7 11	64.09
1 2 4 5 7 9 10 13 12 8 6 3 0 11	61.53
1 5 2 4 9 13 12 10 8 6 3 0 7 11	63.72
1 2 4 5 7 9 8 12 13 10 6 3 0 11	65.41
1 2 4 5 7 13 12 10 8 3 0 6 9 11	63.19

Table 1: Best Path for Robot 1

Paths for cluster 2	Fitness
16 12 10 13 7 4 3 5 9 15 14 8 11 1 0 2 6	83.00
16 13 12 9 11 15 14 8 2 1 3 4 10 7 0 5 6	79.32
16 3 4 7 13 10 12 9 11 15 14 8 2 5 1 0 6	85.29
16 11 8 15 14 12 9 10 13 7 2 1 0 3 4 5 6	87.99
16 12 15 14 11 8 0 4 7 13 10 9 5 3 1 2 6	86.32
16 13 10 7 12 9 8 14 15 11 3 1 0 4 5 2 6	80.52
16 5 0 1 2 8 14 15 11 9 12 13 10 7 4 3 6	82.57
16 13 12 10 7 4 5 9 11 15 14 8 2 3 0 1 6	78.83
16 13 7 5 3 1 2 11 15 14 8 9 12 10 4 0 6	84.44
16 13 12 11 15 14 8 9 2 1 5 10 7 4 0 3 6	80.45

Table 2: Best Path for Robot 2

Best Paths	Fitness
1 5 13 11 25 29 26 14 23 18 9 16 22 28 27 20 24 19 21 15 10 12 2 4 7 8 6 3 0 17	166.21
1 8 6 5 2 4 12 10 15 20 16 11 9 7 13 14 18 22 23 26 29 25 24 28 27 19 21 3 0 17	155.51
1 9 16 13 8 0 2 25 18 14 7 4 5 26 29 23 22 21 19 10 3 6 12 15 27 28 24 20 11 17	178.27
1 23 25 22 24 28 27 21 19 15 20 29 26 14 7 4 2 5 6 10 12 11 16 18 13 9 8 0 3 17	152.02
1 2 3 15 20 23 25 29 26 18 14 16 11 8 5 9 19 21 28 27 24 22 13 7 4 0 10 6 12 17	170.22
1 7 4 9 14 18 23 26 29 25 16 12 19 10 15 8 5 2 11 13 22 24 27 28 21 20 3 6 0 17	165.30
1 3 0 20 24 28 27 19 15 10 6 4 7 9 2 5 8 11 16 29 26 23 25 22 13 14 18 12 21 17	157.23
1 5 10 15 28 20 24 22 13 9 7 23 25 29 26 18 14 4 2 8 11 16 12 6 19 27 21 3 0 17	167.99
1 3 6 10 19 20 21 27 28 22 25 23 26 29 24 15 8 12 16 18 14 7 13 11 9 4 2 5 0 17	150.74
1 2 3 10 15 19 21 22 16 13 14 23 25 29 26 18 8 12 6 5 4 7 9 11 20 24 27 28 0 17	174.70
1 5 8 19 21 27 28 24 20 15 10 6 2 4 7 13 9 11 22 25 23 18 14 29 26 16 12 3 0 17	150.91

Table 3: Best Paths for a Single Robot

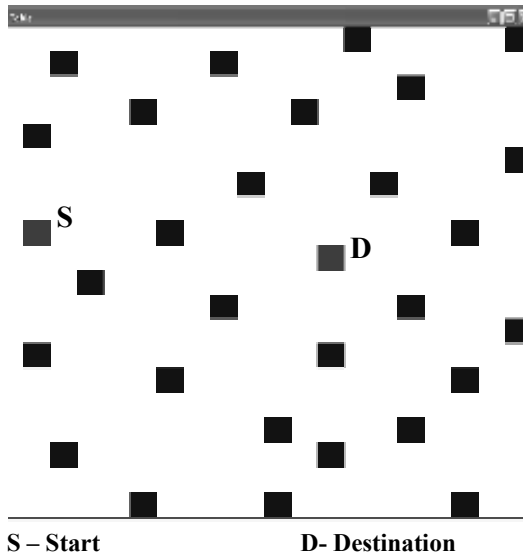


Figure 5: Landmarks in the Environment

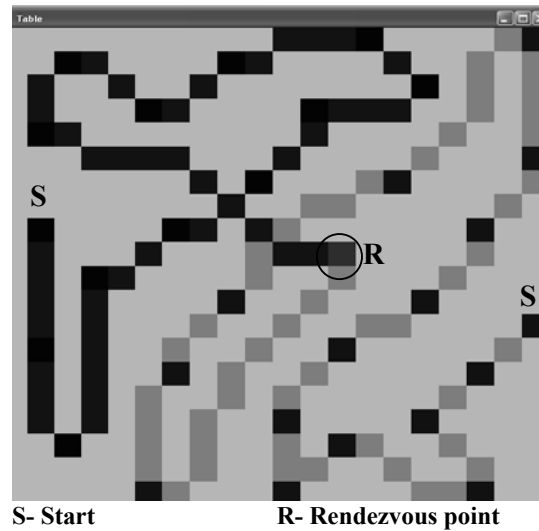


Figure 8: Paths for two Robots & Rendezvous Point

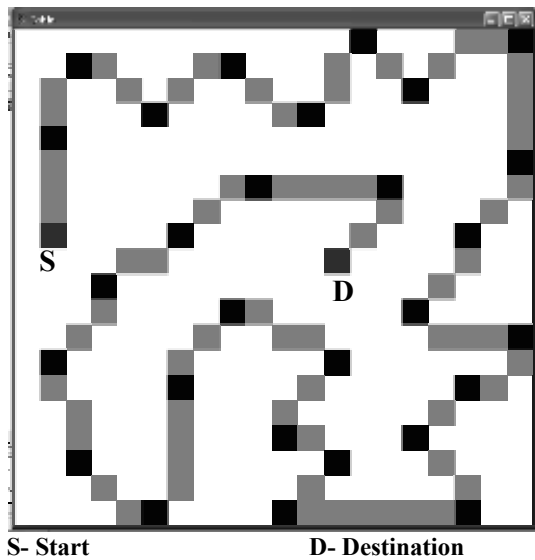


Figure 6: Single Robot's path

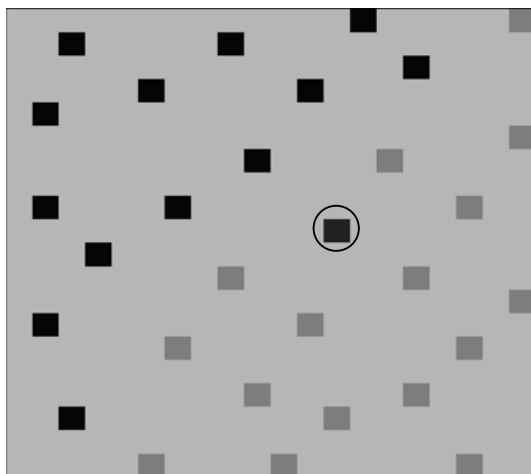


Figure 7: Landmarks in Two clusters

V. CONCLUSIONS

In this study, we propose a hybrid scheme using Genetic Algorithm and Clustering technique for Multi Robot Path Exploration problem. We used a fixed length chromosome that results in reduced computational time when evolution takes place. The experimental results for a single Robot and two robots are quite encouraging. The major advantage of this method is that the optimization is performed off-line and optimal paths are obtained in advance for all the Robots. Also through clustering we achieved a faster path exploration by using multiple Robots. Further extension of the proposed scheme would include incorporation of tour improvement heuristic [11], neighborhood attractor scheme [9] and experimentation with other chromosome representation schemes with appropriate genetic operators.

REFERENCES

- [1]. Achim Bachem, Barthel Steckemetz and Michael Wottawa, An efficient parallel cluster-heuristic for large Traveling Salesman Problems, Technical Report, 1994.
- [2]. Bittencourt L. F., Madeira E. R. M, Cicerre F. R. L., Buzato L. E., A Path Clustering Heuristic for Scheduling Task Graphs onto a Grid, *Proc. the 3rd ACM/IFIP/USENIX International workshop on Middleware for Grid Computing, Vol. 1, pp 1-3, 2005.*
- [3]. Burgard W., Moors M., Fox D., Simmons R. and Thrun S., Collaborative Multi-Robot Exploration, *Proc. the IEEE International Conference on Robotics and Automation, 2000, pp. 476-481.*
- [4]. Burgard, W., Moors, M., Stachniss, C. and Schneider, F., Coordinated Multi-Robot Exploration, *Proc. the IEEE Transactions on Robotics (21), 2005, pp. 376-378.*
- [5]. Giardini G., Kalmar-Nagy T., Genetic Algorithm for Combinatorial Search Problems, *Proc. the IEEE International Workshop on Safety, Security, and Rescue Robotics, 2006.*

- [6]. Mahmuda N, Paul J., Kendall N., and Karl A., A Clustering Heuristic by Effective Nearest Neighbor Selection, *Proc. the 40th Annual Midwest Instruction and Computing Symposium*, 2007.
- [7]. Ramakrishnan, R., Zein-Sabatto, S., Multiple Path Planning for A Group of Mobile Robots in A 2D Environment Using Genetic Algorithms, *Proc. the IEEE Southeast Conference*, 2002, pp.65-71.
- [8]. Roy N. and Dudek G., Collaborative Robot Exploration and Rendezvous: Algorithms, Performance Bounds and Observations, *Journal of Autonomous Robots 11(2)*, 2001, pp. 117-136.
- [9]. Sofge A., Schultz A., and Kenneth De Jong, Evolutionary Computational Approaches to solving the Multiple Travelling Salesman Problem Using a Neighborhood Attractor Schema, *Evo workshops 2002*, LNCS 2279, 2002, pp. 153-162.
- [10]. Solanas A. and Garcia M. A., Coordinated Multi-Robot Exploration through Unsupervised Clustering of Unknown Space, *Proc. the IEEE International Conference on Intelligent robots and Systems*, 2004.
- [11]. Zixing Cai and Zhihong Peng, Corporative Coevolutionary Adaptive Genetic Algorithm in Path Planning of Cooperative Multi-Mobile Robot Systems, *Journal of Intelligent and Robotics Systems (33)*, 2002, pp 61-71.