

# Pattern-Based Genetic Algorithm Approach to Coverage Path Planning for Mobile Robots

Muzaffer Kapanoglu, Metin Ozkan, Ahmet Yazıcı, and Osman Parlaktuna

Eskisehir Osmangazi University, College of Engineering,  
Eskisehir, Turkey  
{muzaffer, meozkan, ayazici, oparlak}@ogu.edu.tr

**Abstract.** Sensor-based mobile robot coverage path planning (MRCP) problem is a challenging problem in robotic management. We here develop a genetic algorithm (GA) for MRCP problems. The area subject to coverage is modeled with disks representing the range of sensing devices. Then the problem is defined as finding a path which runs through the center of each disk at least once with minimal cost of full coverage. The proposed GA utilizes prioritized neighborhood-disk information to generate practical and high-quality paths for the mobile robot. Prioritized movement patterns are designed to generate efficient rectilinear coverage paths with no narrow-angle turn; they enable GA to find optimal or near-optimal solutions. The results of GA are compared with a well-known approach called backtracking spiral algorithm (BSA). Experiments are also carried out using P3-DX mobile robots in the laboratory environment.

**Keywords:** Genetic algorithms, Coverage path planning, Mobile robot.

## 1 Introduction

Start-goal oriented path planning methods do not address the problem of complete coverage path planning, in which every point in a given workspace should be covered at least once [1]. Complete coverage is needed for a variety of applications such as floor cleaning, lawn mowing and street sweeping, and painting [2]. These applications require robot apparatus to move over all the points in the free workspace of the environment. On the other hand, landmine detection, foraging, and patrolling kind of applications require robot sensing range to cover all the points in the environment [3], so called sensor-based coverage.

Different coverage path planning algorithms are proposed for the former applications above ([4], [5], [6], [7]). In the approaches ([4], [5], [6]), the environment is modeled using squares proportional to robot's tool size. However, most of mobile robots use sonar-rings or laser range sensors as sensing devices. Since these devices have a circular sensing range, a disk-based modeling approach may be more effective. Therefore in this study we adopted disk-based modeling approach.

After modeling the environment, the next step is to determine an effective path for coverage. In [4], a method called backtracking spiral algorithm (BSA) is proposed using spiral filling paths built based on squares of robot's tool size. In [5], the workspace is divided into squares with dimension of four times of the robot's tool. Then a

spanning tree based approach is used to generate the path. In [6], the path is planned to guide the autonomous agricultural equipment to completely cover a field while avoiding all known obstacles.

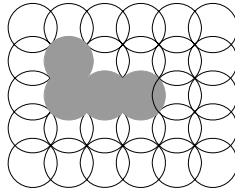
In this study, a pattern-based genetic algorithm is proposed for sensor-based coverage path planning using disk shaped modeling. To the best of our knowledge there are a few genetic algorithm-based coverage planning methods ([6], [7]) for the former applications, and there is no genetic algorithm-based method for the sensor-based coverage. In [8], a disk-based modeling has been studied with a single neighbor-disk prioritization pattern. However, performance degradation due to double-coverage has been observed for some obstacle layouts. In this study, the number of patterns is increased to eight to overcome double coverage problem.

Genetic algorithms utilize the Darwinian and Mendelian principles of genetic evolution. Following its proposal by Holland in 1975, GA's have achieved a high level of popularity owing to its success [9]. Especially, combinatorial and/or non-convex problems have significantly benefited this unorthodox way of problem solving. GAs work on the list of decision variables called chromosomes in which a solution can be obtained directly or by decoding. There are several works related to GA for autonomous mobile robots ([10], [11]) in the literature planning a path from a start position to a goal position rather than dealing with a complete coverage.

In the following section, the proposed algorithm is presented. The experiments and comparisons with BSA have been displayed for real environments with obstacles in Section 3. Conclusions are given in Section 4.

## 2 A Pattern-Based Genetic Algorithm for Mobile Robots

In this study, a pattern-based genetic algorithm utilizing the rectilinear moves represented by eight neighbor-disk prioritization patterns is proposed. The coverage area is divided into overlapping disks, such that, no redundant overlaps are generated and no area is left out of robot's coverage as in Fig.1. The radius of each disk equals to sensing range of the robot sensor which is much greater than robot's physical dimension.



**Fig. 1.** Disk placement pattern

In Fig. 1, obstacles are shown with gray disks. Here we assume that a disk is either fully occupied or completely free. With this modeling, if the robot passes through the center of each disk, the environment would be fully covered. Therefore, the sensor based coverage problem turns into planning a path to visit all of the disk centroids. The objective of this approach is to minimize the multiple visits of disks while supporting rectilinear moves.

To determine the visiting order of the disks, a GA-based method is developed. Eight premeditated neighbor-disk-prioritization patterns (PP) (first four are given in Fig. 2) are designed to provide disciplined, reasonable rectilinear moves. The robot is relatively located in the middle of the each PP indicated by P-prefix in Fig.2. The numbers in the patterns indicate the preferred neighbor-disks. In other words, each PP prioritizes the neighbors in ascending order so that the disk numbered 1 has the highest priority if unvisited yet. 1-4 numbered disks are one-disk away from the robot’s current disk, while 5-12 disks are two-disks away, and so on. Although 1-4 numbered disks are in the same distance, they are also prioritized according to the direction they represent. The PP will guide the robot until a dead-end is arrived. In this case, an unvisited neighbor disk at the shortest rectilinear distance is sought. If one found, the path is directed to that disk, by double-visiting at least one disk. The operation taking the robot from a dead-end to an unvisited disk is called *recovery*. Since one PP may guide the robot to a dead-end while another may never cause such a dead-end, different premeditated PPs are used to obtain sophisticated possible patterns for the path.

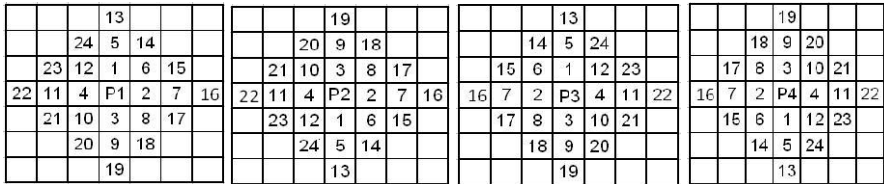
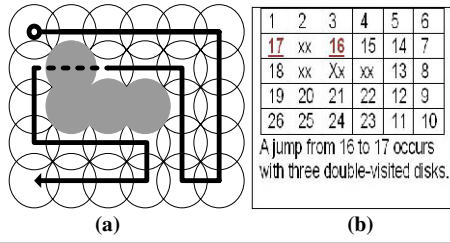


Fig. 2. Neighbor-disk prioritization patterns

**Representation**

In the proposed algorithm, a *k<sup>th</sup>-neighbor* representation is preferred. Each gene represents the next neighbor’s number as indicated by their driving PP. The order of genes in chromosomes (i.e., locus) corresponds to the move number. For an N-disk coverage problem, the length of the chromosome equals (N-1) genes corresponding to free disks. A sample environment is given in Fig. 3a. The environment is modeled by 30 disks: 4 of them are occupied by obstacles hence requiring (30-4-1=) 25 genes. Starting from the upper left corner and by using pattern P1, a possible coverage path is determined as in Fig. 3b. In this figure the number in a cell corresponds to the place of the gene in the chromosome. Since P1 assigns the first and second priorities to up and right moves respectively, and the up moves face the boundary, second priority move which is to the right is applied for next five disks. Following the same pattern, P1, the next four moves occurs at the third priority move which is downward leading to the bottom boundary. This operation continues till the robot faces a dead-end after 15 moves. First 15 moves leading to disk number 16 are considered regular pattern-based moves. At that point, a shortest path to an unvisited disk is determined. The disk number 17 has the 11<sup>th</sup> priority with respect to P1. For accessing nearest unvisited disk, shortest path procedure generates a path starting at disk 16 and ending at disk 17 and crossing over disks 1, 2, and 3 once again. Dashed lines in Fig. 3.a represent the shortest path between disks 16 and 17 via disks 1, 2, and 3. The remaining path is a result of P1. Encoded chromosome of this example is given in Fig. 3.c. In this figure, for example, the value of gene#1 is 2 representing right move.



Chromosome.1																									
Genes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Neighbor	2	2	2	2	2	3	3	3	3	4	1	1	1	4	4	11	3	3	2	2	2	3	4	4	4

(c)

**Fig. 3.** a) Sample environment, b) Decoded chromosome, c) Genetic representation

The chromosomes generated by using patterns increase the variety in the initial population with rectilinear moves. If the area contains no obstacles, GA will less likely improve the paths built based on pattern-driven moves. Patterns imply a relative location; therefore they must be evaluated for each disk when needed. As the number of PPs increases, the proposed pattern-based GA should be expected to improve paths with less number of redundant coverage at earlier generations. However, patterns included in this study are designed empirically leaving design, selection, and optimal number of PP to the future studies.

The steps of the proposed pattern-based GA are as follows:

- Generate Initial Population;
- While (stopping criteria are not satisfied)
  - Evaluation & Selection
  - Reproduction
  - Crossover
  - Mutation
  - Enlarged New Population
- Loop

These steps are explained in the following subsections.

**Generation of Initial Population**

In building the initial population, all eight PPs are used to generate a variety of rectilinear movement patterns overcoming obstacles. The population size is divided as equally as possible among PPs. However, for increased variety, we perform a perturbation over PP, and then let the pattern-driven moves handle the rest. Population is first filled with chromosomes generated with respect to PPs, 8 chromosomes for this case. For generating the remaining chromosomes, these 8 chromosomes are manipulated by perturbing a gene’s value corresponding to a feasible move randomly. These perturbations act as a mutation in the initial population. Main expectation from these pattern-perturbed moves is to let the path out from the local best solutions. When a pattern is interrupted by an obstacle, then a *Shortest Unvisited Neighbor procedure* is run the same way as its name implies. It means that there will be a “jump” from one disk to another indicating an inevitable repeated coverage of some disk(s).

**Evaluation and Selection**

The fitness value of a chromosome corresponds to the number of disks visited by the robot for complete coverage of the environment. Since an ideal solution contains no redundant coverage, the ideal fitness is equal to the distance of travelling each disk exactly once. Therefore the fitness of a solution increases by minimizing the number of disks visited more than once. Stochastic tournament selection with elitism is applied based on fitness values.

**Crossover Operator**

We customized parametric crossover for aligning with pattern-driven moves. A crossover mask is used with a masking probability. Masking probability determines the percent of the genetic material that will be exchanged between the chromosomes. If both chromosomes were built based on the same PP, this operator picks a random number for each gene. If this random number is greater than or equal to the masking probability then that gene is replaced with the gene value of the chromosome#1 if it leads to a feasible move. If the corresponding move is not feasible, then priority number is increased until a feasible move is reached. If the two chromosomes are generated with respect to different PPs, then one of the patterns must be expressed in terms of the other PP. The PP that is kept as is will be called the *reference pattern*. An example of crossover operator is illustrated in Fig. 4. In Fig.4, the P1 is used as the reference pattern. Chromosome 1 and 2 are selected for crossover operator. Chromosome 1 is generated with respect to P1, and chromosome 2 is generated with respect to P4. Therefore, the first priority disk of P4 corresponds to the 3<sup>rd</sup> disk of P1. For ensuring that each PP refers to the same disk during crossover operation, chromosome 2 is expressed in terms of reference pattern which is P1. While generating Child.1, we process chromosome 1 (Chrom.1 of Fig.4) and converted chromosome 2 (Conv. Chrom.2 of Fig.4). The row labeled Mask in Fig.4 represents if value is taken from the chromosome 1 or 2 depending on the random values as selected. This way, Child.1 takes value 2 in the first gene from the first chromosome depending on the mask value. If the selected random number were less than masking probability, the value of the first gene would turn out to be 3 which is the value of the converted chromosome 2. Another parameter related to this operator is crossover rate which determines what percent of the chromosomes will be subject to the crossover operation.

Genes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Chrom.1	2	2	2	2	2	3	3	3	3	4	1	1	1	4	4	11	3	3	2	2	2	3	4	4	4
Chrom.2	1	1	1	1	1	3	1	1	1	3	1	1	1	3	3	2	3	2	2	3	2	11	1	1	1
Conv. Chrom.2	3	3	3	3	2	1	2	3	2	1	2	3	2	1	1	4	1	4	4	1	4	7	2	2	3
Mask	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1	1	1
Child.1	2	2	2	2	2	3	3	3	3	4	2;4	3;4	2;4	1	1;2	4;2	1;2	4;1	4;1	1;4	4	11	2;3	2;3	3

**Fig. 4.** Illustration of the modified parametric crossover operator for two chromosomes with masking probability

The impact of the crossover for the example environment is shown in Fig. 5. The path presented in Fig.5 is built based on the moves obtained from Child.1 chromosome of Fig.4. First 10 moves obtained from parents yield a feasible pattern. As a result of crossover of the parents, gene#11 of child chromosome 1 would take the value of 2. However this move is not feasible, the priority number is increased by 1 until the first feasible move is detected which is 4. The priority 4 corresponds to the 12<sup>th</sup> disk in Fig.5.

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

Fig. 5. Path generated based on the crossover of the sample chromosomes

**Mutation Operator**

The mutation operator is designed to incorporate the exploration impact. A randomly selected gene’s value is changed from its current PP value to a feasible one arbitrarily. Following the mutated gene, all the moves again are generated with respect to its current PP. Although one gene change seems to be a modest change, it might result in a drastic change in the entire chromosome depending on the order of the gene selected for mutation. Due to this fact, we allowed only one mutation per chromosome. In Fig.6, the value of the gene 19 is changed from 2 to 3 arbitrarily. As a result, two-third of the remaining genes was changed yielding a different pattern.

Genes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Chrom.1	2	2	2	2	2	3	3	3	3	4	1	1	1	4	4	11	3	3	2	2	2	3	4	4	4
After Mut.	2	2	2	2	2	3	3	3	3	4	1	1	1	4	4	11	3	3	3	2	1	2	2	3	4

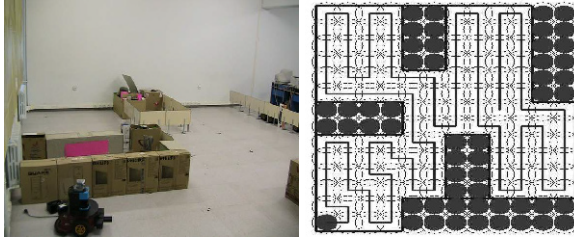
Fig. 6. A Illustration of the mutation operator (the number of the gene mutated=19)

The proposed pattern-based GA’s typical parameters are set as follows: Population size: 500, maximum number of generations: 100, mutation probability:1/#ofGenes, crossover probability: 10%; tournament size: 2, crossover masking probability:70%, elite rate: 1%. In the proposed GA, the chromosomes of the current population, populations after crossover and after mutation are put together to build the enlarged population. Then selection operator reduces the size of the enlarged population to the original size again.

**3 Experimental Results**

The algorithm has been tested both in simulation environment, and at AI& R lab [12] test platform using Pioneer-3 DX mobile robots. A picture of the 840cmx720cm platform is given in Fig. 7.a. The map of the environment modeled with 116 disks is

given in Fig. 7.b. The obstacles are represented with gray disks. It has three rooms and a hallway to connect the rooms. The environment is designed to pose a challenge for MRCPP methods. In Fig. 7.b, thick lines show constructed path by the proposed method. The total distance of movement of the proposed method is 115 units, and there is no repeated coverage. During the tests several additional obstacles are also placed into different parts of this environment to increase the complexity of the environment.



**Fig. 7.** AI&Robotic laboratory, **a)** Picture of the platform **b)** Map of the platform with a tour

In the following, the results of the proposed method are compared with the results of an existing grid based method, BSA [4]. BSA was chosen for comparison, because it divides the environment into individual cells and uses each cell as a decision unit to plan the coverage path similar to the approach presented in this paper. Another candidate method for comparison would be STC, but classical STC uses squares with dimension of four times of the robot's tool size to model the environment and assumes that if there is one obstacle in one of the squares, all four squares are treated as occupied. Therefore, some of free disks left as uncovered. Since our method aims complete coverage, it is not compared with STC. Later, experiments are carried out using Pioneer-3 DX robot in the laboratory environment.

### 3.1 Comparison of the Proposed Method with BSA

BSA is based on the execution of spiral filling paths. Before starting a spiral path, the robot is located nearby an obstacle which is situated in the reference lateral side RLS. RLS indicates the relative direction where obstacles are to be referenced during the spiral filling procedure. RLS is fixed in advance and can't be modified. OLS is the opposite lateral side, it identifies the antipode of RLS. The following set of four simple reactive rules allows the correct execution of the spiral coverage procedure [4]:

- RS1 IF (obstacles\_all\_around): THEN ending\_spiral\_point\_detected
- RS2 IF (NOT obstacle\_in\_RLS): THEN turn\_to(RLS) and move\_forward
- RS3 IF (obstacle\_in\_front): THEN turn\_to(OLS)
- RS4 OTHERWISE move\_forward

Once the robot has reached the central ending point of a spiral filling path, the backtracking mechanism is invoked. It is employed to get back to unvisited areas, where a new spiral filling procedure can be performed. Backtracking points are detected and stored during the execution of a normal spiral path.

In order to compare the proposed method and the BSA, several obstacle configurations are used. In Figure 8.a-b, a five-disk obstacle is placed in the hallway blocking entrance of one room. Thick lines show the coverage path of the robot while the thick dashed-dot lines show redundant moves of the robot to access uncovered disks. The total distance of movement of the proposed method in Fig. 8.a is 114 units. For the same environment, the result of the BSA is 127 units as in Fig. 8.b. Therefore, 4 and 17 disks are covered repeatedly with the proposed method and BSA, respectively.

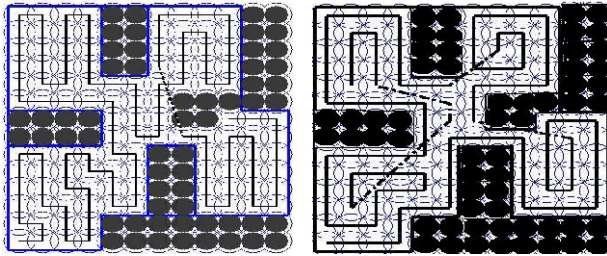


Fig. 8. a) Result of proposed method, b) Result of BSA

Due to page restrictions only one obstacle layout is given as in Fig. 8.a-b, and the results of the methods for five different obstacle configurations are summarized in Table 1. For each obstacle layout, ND.P.M. denotes the number of disks for the proposed method, N.RD P.M. denotes the number of repeated disks for the proposed method, RC denotes percentage of repeated coverage, ND.P.M. denotes the number of disks for the BSA, N.RD BSA. denotes the number of repeated disks for the BSA, and IMP denotes percentage improvement in the tour by the proposed method compared to BSA.

Table 1. Comparison the results of the proposed method and BSA

	N.D P.M.	N.RD P.M.	RC % P.M.	N.D. BSA	N.RD BSA	RC % BSA	IMP.%
Env#1	115	0	0	137	22	16	16
Env#2	113	0	0	137	24	17.5	17.5
Env#3	117	4	3.4	132	19	14.5	11.4
Env#4	112	1	1	132	20	15	15
Env#5	114	4	3.5	127	17	13.5	10

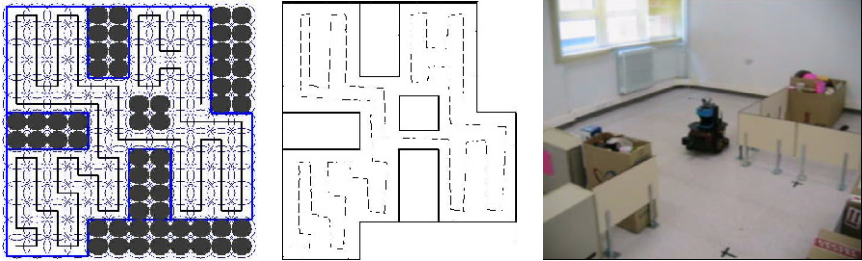
Env#1 denotes the environment in Fig.7 without obstacle, Env#2 denotes a two-disk obstacle in the middle, Env#3 denotes two one-disk obstacles placed at the room entrances, Env#4 denotes four-disks in bottom-left room, and Env#5 denotes five-disk obstacle blocking hallway as in Fig. 8. Figures of the environments #1-#4 could be reached from the web site [12]. Table 1 show that the proposed method has less repeatedly covered disk compared to BSA. Therefore, the robot may finish the coverage task in a shorter time and consuming less energy.



### 3.2 Experiments Using Pioneer 3-DX Mobile Robot

The proposed algorithm is applied to sensor-based coverage of an indoor environment using Pioneer 3-DX robot. The robot has an onboard P3-800 computer with Linux OS. The sensors on the robot are: SICK LMS 200 laser range finder, sonar ring sensors, camera, and compass. Aria ARLN software module is used for the localization purpose [13]. SICK LMS 200 laser range sensor is used for the sensor based coverage task. The sensor has normally a range of 50 meters, but for experimental purposes the range is restricted to 40cm with software.

An additional four-disk size obstacle is placed in the middle of the layout (Fig.7.b) to create a more challenging environment as in Fig. 9.a. The path of the proposed method is given in the Fig. 9.a is followed by the robot as given in Fig. 9.b. This figure is drawn using log values of the robot during the experiment. Fig. 9.c shows snapshot of the recorded video during the movement of the robot. The full video of the live performance of the robot is recorded, and can be reached from following web site: [www.ai-robotlab.ogu.edu.tr](http://www.ai-robotlab.ogu.edu.tr) [12].



**Fig. 9.** a) Planned path using the proposed methods, b) The path followed by Pioneer-3 DX mobile robot in the laboratory, c) Snapshot of the recorded video

## 4 Conclusions

Sensor based coverage path planning for a mobile robot is one of the recent research issues. Among several approaches proposed, a few articles have addressed the use of the genetic algorithms for MRCPP problem. This article proposes prioritized movement patterns to generate efficient rectilinear coverage paths. They enable GA to find optimal or near-optimal solutions. Comparison with one of the existing methods in the literature, the proposed method resulted in 10% to 17.5% improvement in traveled distance over BSA for a set of challenging test environments with less repeated coverage. The results we obtained so far have been rather promising and efficient. The approach is yet to be studied for multiple robots cases.

## Acknowledgment

This work is supported by the Scientific and Technical Research Council of Turkey (TUBITAK), Contract No 107E064, project title: “Mobile Robot Route Planning for Complete Coverage of Dynamic Indoor Environments”.

## References

1. Choset, H.: Coverage for robotics - A survey of recent results. *Annals of Mathematics and Artificial Intelligence* 31, 113–126 (2001)
2. Guo, Y., Balakrishnan, M.: Complete coverage control for nonholonomic mobile robots in dynamic environments. In: *Proceedings of 2006 IEEE International Conference on Robotics and Automation*, pp. 1704–1709 (2006)
3. Acar, E.U., Choset, H., Lee, J.Y.: Sensor-based coverage with extended range detectors. *IEEE Transactions on Robotics* 22(1), 189–198 (2006)
4. Gonzalez, E., Alvarez, O., Diaz, Y., Parra, C., Bustacara, C.: BSA: A Complete Coverage Algorithm. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2040–2044 (2005)
5. Gabriely, Y., Rimon, E.: Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence* 31, 77–98 (2001)
6. Ryerson, A.E.F., Zhang, Q.: Vehicle Path Planning for Complete Field Coverage Using Genetic Algorithms. *Agricultural Engineering International: the CIGR E journal* IX, 1–11 (2007) (Manuscript, ATOE-07-014)
7. Jimenez, P.A., Shirinzadeh, B., Nicholson, A., Alici, G.: Optimal area covering using genetic algorithms. In: *Proceedings of the 2007 IEEE /ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1–5 (2007)
8. Kapanoglu, M., Ozkan, M., Yazici, A., Parlaktuna, O.: A Genetic Algorithm with Orientation Compasses for Single or Multi-Robot Coverage Path Planning. In: *Proceedings of 6th International Symposium on Intelligent & Manufacturing Systems*, pp. 668–678 (2008)
9. Goldberg, D.E.: *Genetic Algorithm in search, optimization, and Machine Learning*. Addison-Wesley, Reading (1989)
10. Castillo, O., Trujillo, L., Melin, P.: Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots. *Soft Computing* 11(3), 269–279 (2007)
11. Erinc, G., Caprin, S.: A genetic algorithm for nonholonomic motion planning. In: *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pp. 1843–1849 (2007)
12. Artificial Intelligence and Robotic laboratory, Eskisehir Osmangazi University, Turkey, <http://www.ai-robotlab.ogu.edu.tr>
13. Aria- ARNL software module (2009), <http://robots.mobilerobots.com>