The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Planar Multi-robot Realizations of Connectivity Graphs Using Genetic Algorithms

Halûk Bayram and H. Işıl Bozma
Intelligent Systems Laboratory, Electrical and Electronic Engineering
Bogazici University, Bebek 34342 Istanbul Turkey

***Abstract*— This paper considers the problem of planar multi-robot realizations of connectivity graphs. A realization is a set of planar positions for a team of robots with a connectivity graph that is identical to an *a priori* given connectivity graph with the additional constraint that it must be feasible. Feasibility means that that the robots must not be overlapping with each other. As the associated mathematical problem is known to be NP-hard, a stochastic approach based on genetic algorithms is proposed. First, a population set based on randomly generated planar and feasible multi-robot positions is generated. Next, a fitness function that measures the similarity of the graph of each member is constructed. Finally, new reproduction operators that enable the evolution of generations are introduced. An extensive statistical study with different number of robots demonstrates that the proposed algorithm can be used to obtain fairly complicated network topologies.**

I. Introduction

This paper considers the problem of planar multi-robot realizations of connectivity graphs. The problem is defined as the generation of robot planar positions whose connectivity graph is identical to an *a priori* given connectivity graph with the additional constraint that the robots must not be overlapping with each other. This problem arises in many applications such as exploration, search, patrolling and collective games (such as soccer) that require automatic positioning of multiple robots with a particular underlying connectivity graph constraint. For example, in multi-robot deployment and coordination tasks with limited communication, robot positions must satisfy the particular connectivity graphs [12], [1]. This paper presents a stochastic approach to this problem and proposes a graph-based genetic algorithm for generating planar multi-robot realizations.

A. Related Literature

There are three related areas: robot networks, disk graphs, and graph drawing. In multi-robot systems, the concept of connectivity graphs has been introduced which imposes various constraints on the relative positions of the robots [10], [9]. For example, connectivity graphs provide a graph-theoretic model for broadcast networks where the radii of the circles correspond to the communication range ρ_c. Interestingly, while many approaches are based on graph based models, the issue of whether an arbitrary graph has a multi-robot realization or not has been mostly overlooked.

Connectivity graphs are known as unit disk graphs in graph theory – which are the intersection graphs of closed disks in the plane where each vertex corresponds to a circle and edge appears between two vertices when the corresponding circles intersect [3]. Of course, the distance unit is not critical since the disks realize the same graph even if the coordinate system is scaled appropriately. Furthermore, unit disk graphs have several alternative definitions that are all equivalent to each other up to a choice of scale factor. Two such alternatives are the intersection graph of equal radius circles or a graph formed from a collection of circles all having the same radius where two circles are connected by an edge if each circle contains the center of the other circle. The set of disks is said to realize the graph [2]. A realization is therefore a mapping of the vertices to points which realize the graph. The recognition problem of unit disk graphs is then posed as: Given a graph, determine if it has a realization [2]. It has been shown that recognizing unit disk graphs is NP-hard. The results are also shown to hold for the disk touching graphs – namely all disks have disjoint interiors.

The NP-hardness of the problem has motivated the development of approximate, potential-based or stochastic approaches – as many problems from different problem areas nevertheless require solutions to the planar realization problem. An inapproximability result has placed a bound on how well the planar coordinates can be derived from the connectivity information alone [8]. In graph drawing, where the goal is to produce aesthetically pleasing drawings of general undirected graphs, one proposed approach is the spring model algorithm. Here, the graph is viewed to be a mechanical collection of rings (the vertices) and connecting springs (the edges) with minimal energy configuration attained when the network graph approaches the goal graph [7]. However, as the spring method is likely to be trapped by local optima, the configurations that are obtained are very poor. In the genetic algorithm TimGA, aesthetic criteria used such as the number of edge crossings, even distribution of nodes, and edge length deviation are utilized [5]. An extension of this work that develops new mutation operators has been proposed in [11]. However, none of these algorithms enforce adherence to a given connectivity graph by considering both edges and non edges (namely vertex pairs that should not have a link between them). The novelty of the proposed approach is to consider the graph realization problem and to propose a genetic algorithm which has proved to be statistically working.

B. States and Graphs

Let $P = \{1, \ldots, p\}$ be the set of robots. Each robot $i \in P$ is associated with the radius $\rho_i \in \mathbb{R}$ and state $b_i \in W \subset \mathbb{R}^2$.

Define $b \in \mathbb{R}^{2p}$ to be the robots' state as $b = \sum_{i \in P} b_i \otimes e_i$, where $e_i \in \mathbb{R}^p$ are the unit vectors in $\mathbb{R}^p$ and $\otimes$ is the Kroenecker operator. Let $\delta_{ij} = \| b_i - b_j \|$ denote the robots' pairwise relative distance. Since robots cannot overlap each other, it is required that

$$\forall ij \qquad \delta_{ij} \geq \rho_{ij} \tag{1}$$

where $\rho_{ij} = \rho_i + \rho_j$. Each robot should also stay in workspace that is bounded by radius ρ_0, it is also required that

$$\forall j \in P \qquad \rho_{0j} \geq \| b_j \| \tag{2}$$

where $\rho_{0j} = \rho_0 - \rho_j$. Note that this constraint implies that W is bounded by radius ρ_0. The free robot configuration space $F \subset \mathbb{R}^{2p}$ satisfies Eq.1 and Eq.2 and hence defines the feasible robot positions.

Suppose that the robots have limited communication range ρ_c. A disk-neighborhood of robot i is a closed ball $B_{\rho_c}(b_i)$ of radius $\rho_c >> \rho_i$ around $b_i \in \mathbb{R}^2$. Given ρ_c, any configuration $b \in F$ induces a state-dependent mapping $g : F \to G$. Here $G = \{g' | g' \subseteq K_p\}$ is the set of all possible graphs on P and K_p is the complete graph [6]. The image of the graph map is $g(b) = (\mathcal{P}, E(b))$ is known as the connectivity graph [10]. Here $E(b)$ is the set of edges as defined by the connectivity matrix $A(b) = [a_{ij}(b)]$:

$$E(b) = \{ij \mid a_{ij} = 1\} \tag{3}$$

The connectivity matrix $A(b)$ is defined as follows:

$$a_{ij}(b) = \begin{cases} 1 & \delta_{ij} \leq \rho_c \text{ and } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

C. Problem Statement

Consider a graph function $g : F \to G$. Suppose we are given $g^* \in G$. The set $g^{-1}(g^*) \subset F$ represents the set of robot configurations all having graph g^*. The goal is to find a realization $b \in F$ such that $b \in g^{-1}(g^*)$.

II. General Approach

Our approach is based on genetic algorithms. This Section first gives a broad overview of genetic algorithms. Following, we describe its adaptation to the planar realization problem.

A. Genetic Algorithms

Genetic algorithms are a class of evolutionary methods for determining the optimal classifiers or equivalently an optimal solution [13], [4]. In genetic algorithms, a classifier is represented by a string of genes that is also known as a chromosome. The mapping from a chromosome to the features of a classifier is flexible and depends on the application. In broad overview, genetic algorithms employ stochastic search to evolve the best chromosome. First, a population set of classifiers is constructed. Here each chromosome differs somewhat from the others in the population. Next, a fitness function that evaluates the goodness of each chromosome is constructed. This function is used to compute the score of each chromosome. Following, the classifiers are ranked according to their score and only the fittest are retained. These are then stochastically altered to generate the next generation. There are three primary genetic operators that govern reproduction: replication, crossover and mutation. Replication is mere reproduction. Crossover involves the mating of two different chromosomes via exchanging certain parts. Mutation occurs when the genes change. The overall process is repeated for the succeeding generation. The process is terminated when at least one chromosome has a score that exceeds an a priori specified value.

B. Adaptation to Planar Multi-Robot Realization

In employing genetic algorithms, we must first specify the map from a chromosome to the properties of the classifier. In the proposed approach, each chromosome corresponds to a particular state $b \in F$ and hence a team of p robots that are all located within W. The goal is to generate a state (equivalently a chromosome) that has the given connectivity graph g^*.

An initial population set $S(0) \subset F$ is constructed where the cardinality $|S(0)|$ is set a priori to N_P. Next, a selection process initiates a new generation $S(k) \subset F$ where $k \in Z^+$ is the generation number. The fitness of each chromosome in the current population is evaluated based on a fitness function f and the members of this population set are ranked accordingly. The population $S(k)$ is via selecting members from $S(k-1)$ randomly with probability depending on the relative rank value of the individuals [13]. In this manner, population members having higher fitness are chosen more than those having lower fitness values. These members of the population are to be used in the stochastic alteration that follows. Two primary genetic operators govern reproduction: crossover and mutation. This is followed by replacement where the population is changed via replacing the children with the parents. Elitism is applied to keep the best chromosome in new population. If there is no improvement in the elite chromosome after a predefined number of generations N_E, newly generated randomly realizations are used to replace the p_A percent of the population while keeping the best p_B percent of the population. The process is halted when fitness of a generation reaches a desired level or when the number of generations exceeds a given value N_G.

III. Fitness Function

Given a member $b \in S(k)$ of any population, the fitness function should measure the similarity between its graph $g(b)$ and the goal g^*. This is equivalent to finding $b \in F$ in such a way that its adjacency matrix $A(b)$ is the same as that $A = [a_{ij}]$ of the given goal graph g^*. Recalling that $\delta_{ij} = \| b_i - b_j \|$, the fitness function $f : F \to \mathbb{R}^{\geq 0}$ encodes the following measures.

First, the similarity of adjacency matrix $A(b)$ of a given realization b with that of the goal A is measured. For this, both the edges and no edges of $A(b)$ must be compared with the corresponding entities of A. If the goal graph has an edge ij and hence $a_{ij} = 1$, then $\rho_{ij} \leq \delta_{ij} \leq \rho_c$ which is measured

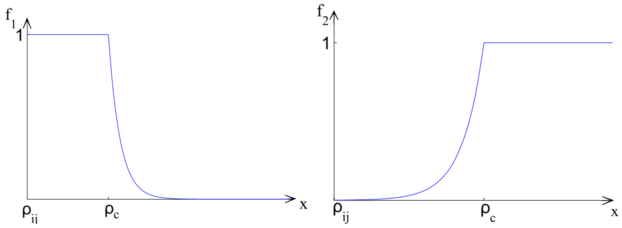

Fig. 1. Left: f_1 function; Right: f_2 function.

by the function $f_1 : R^{\geq 0} \to [0, 1]$ defined as:

$$f_1(x) = \begin{cases} 1 & x \leq \rho_c \\ e^{-(x-\rho_c)} & x > \rho_c \end{cases} \tag{5}$$

Similarly, if the goal graph does not have the edge ij, then $a_{ij} = 0$ which means that $\delta_{ij} > \rho_c$ which can be measured by the function $f_2 : R^{\geq 0} \to [0, 1]$ defined as:

$$f_2(x) = \begin{cases} 1 & x > \rho_c \\ e^{(x-\rho_{ij})}/e^{(\rho_c-\rho_{ij})} & x \leq \rho_c \end{cases} \tag{6}$$

Hence, the term $\sum_{ij} a_{ij} f_1(\delta_{ij}) + (1 - a_{ij}) f_2(\delta_{ij})$ varies 0 in case of being completely different and $p(p-1)$ in case of being completely identical.

Next, we also measure the amount of dissimilarity between the adjacency matrix $A(b)$ of b and that A of the goal. It is composed of two terms. For all missing edges in $A(b)$ that are present in A, the function $f_3 : R^{\geq 0} \to R$ measures the distance each robot pair need to approach in order to come within each others' communication range ρ_c: δ_{ij} using:

$$f_3(x) = x - \rho_c \tag{7}$$

Similarly, for all the superfluous edges present in $A(b)$ that should not be present with respect to A, the function $f_4 : R^{\geq 0} \to R$ measures how close is the corresponding robot pair from moving within each others' neighborhood as:

$$f_4(x) = 2\rho_o(1 - x/\rho_c) \tag{8}$$

The term $1 + \sum_{\substack{ij \notin E(b) \\ ij \in g^*}} f_3(\delta_{ij}) + \sum_{\substack{ij \in E(b) \\ ij \notin g^*}} f_4(\delta_{ij})$ varies between 1 when the target connectivity graph is realized and being a large number in case of being different.

Hence, the fitness function can be constructed as the ratio of these two terms as:

$$f(b) = \frac{\sum_{ij} a_{ij} f_1(\delta_{ij}) + (1 - a_{ij}) f_2(\delta_{ij})}{1 + \sum_{\substack{ij \notin E(b) \\ ij \in g^*}} f_3(\delta_{ij}) + \sum_{\substack{ij \in E(b) \\ ij \notin g^*}} f_4(\delta_{ij})} \tag{9}$$

IV. Reproduction

Two primary genetic operators govern reproduction: crossover and mutation.

A. Crossover

For crossover reproduction, first, a mating pool is generated via considering each robot $i \in P$ and adding it to the mating pool with probability p_C. If the number of robots in this set turns out to be an odd number, a randomly selected robot is removed from the pool. The crossover operator considers each consecutive two members $b^-, b'^- \in S(k)$ of the mating pool and replaces them by two offsprings b^+ and b'^+. The superscripts $-$ and $+$ indicate each chromosome before and after the crossover respectively. Three types of crossover operators are considered: Single, multi-point and square.

1) Single Crossover: In the single point crossover, the location of a randomly selected robot i is swapped between the parents. That is to say, $b_i^+ = b_i'^-$ and $b_i'^+ = b_i^-$. All the links are adjusted accordingly. If the crossover leads to an unfeasible configuration for any of the resulting offsprings, the offsprings are not added to the population $S(k)$. Instead, the crossover is repeated with another robot $l \neq i$. If a feasible configuration is not available after trying all the indices, the crossover operation is not applied to this pair.

2) Multi Point Crossover: Multi-point crossover is similar to single point crossover. Here, the parents are spliced into two groups via choosing a robot index i randomly. The offsprings are generated via exchanging one of the groups formed. b^+ and b'^+ are defined as follows for $\forall j \in P$:

$$b_j^+ = \begin{cases} b_j'^- & \text{if } 1 \leq j \leq i \\ b_j^- & \text{if } i < j \leq p \end{cases} \tag{10}$$

$$b_j'^+ = \begin{cases} b_j^- & \text{if } 1 \leq j \leq i \\ b_j'^- & \text{if } i < j \leq p \end{cases} \tag{11}$$

If the resulting operation leads to an unfeasible configuration for any of the offsprings, again the offsprings are not added to the population set $S(k)$. Instead, another robot index $l \neq i$ is selected. This is repeated until either both of the offsprings have feasible configurations or all the indices are depleted. In case of failure to generate offsprings having feasible configurations, the pointwise crossover is not done for this pair.

3) Square Crossover: In square crossover, two offsprings are generated via exchanging a small set of robots between the two parents. The exchange is based on two square regions $C^-, C'^- \subset W$ both centered at the same location $c^- \in W$ in W, having the same edge length D_C. These square regions are selected in a manner such that there exists at least two robots $i, j \in \mathcal{P}$ with $b_i \in C^-$ and $b_j' \in C'^-$. Hence, each square contains at least one robot from one parent. Let M and M' denote the index set of robots in C^- and C'^- respectively as:

$$\begin{aligned} M &= \left\{ i \in P | b_i^- \in C^- \right\} \\ M' &= \left\{ i \in P | b_i'^- \in C'^- \right\} \end{aligned}$$

Following, two offsprings b^+ and b'^+ are generated by first duplicating each parent exactly, exchanging robots inside C^- and C'^- while the locations of the rest of the robots are kept unchanged as much as possible. Furthermore, the centers of C^- and C'^- are moved to a new randomly selected location $c^+ \in W$ in the offsprings' workspace while ensuring that there is no collision. Of course, all the robots in the offsprings

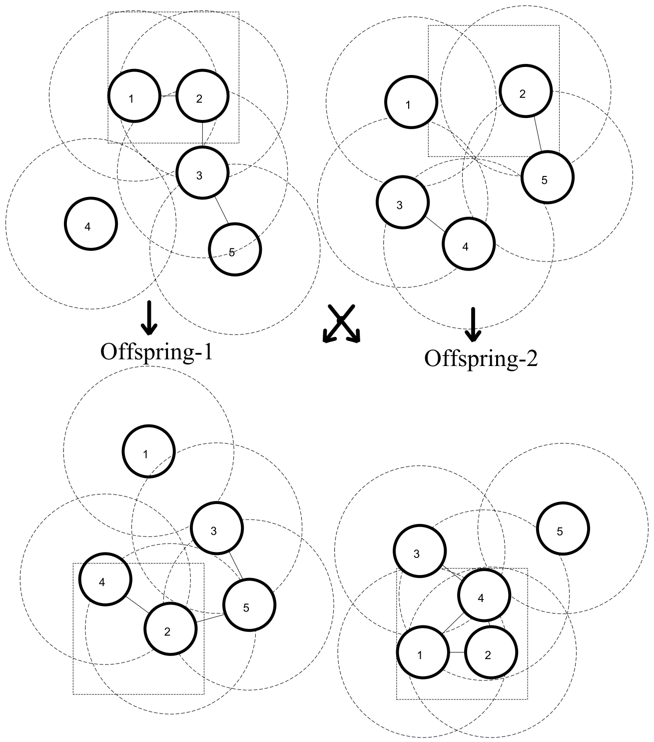

Fig. 2. Square crossover. In each graph, the robots are shown with dark circles and the communication ranges are shown with the dotted circles. If two robots's centers are within each others' communication range, a link is established as shown by the lines connecting their centers. The dotted circles indicate the squares selected. The two offsprings are generated via exchanging the robots.

having indices identical to those in M and M' are removed. The offsprings b^+ and $b_i'^+$ are defined as follows:

$$b_i^+ = \begin{cases} b_i'^- - c^- + c^+ & \text{if } i \in M' \\ b_i^- & \text{if } i \notin M' \end{cases} \tag{12}$$

$$b_i'^+ = \begin{cases} b_i^- - c^- + c^+ & \text{if } i \in M \\ b_i'^- & \text{if } i \notin M \end{cases} \tag{13}$$

A sample square crossover is as shown in Fig. 2. Here, C^- contains robots 1 and 2 and hence $M = \{1, 2\}$. Similarly, C'^- contains robot 2 and hence $M' = \{2\}$. Next, C^- and C'^- are moved to a new center. The two offsprings are generated via copying all the robots while exchanging those in M and M' respectively. The operation is completed after removing all the robots having identical indices with those in M and M'.

B. Mutation

The mutation operator is used to increase the variability of the population by perturbing each robots' position in a graph with a given probability. The algorithm uses two different kinds of mutation operators – robot and link mutations. At each iteration k, all the members of the population $S(k)$ are considered and only one type mutation is selected with probability p_R for robot mutation and p_L for link mutation.

If robot mutation is selected, robot mutation is applied on all the robots b_i in the given sample. There are two alternative operators depending on the restriction on the mutated location of each robot. The new position can be perturbed largely or slightly which correspond to the two types of operators. Each is selected with probabilities p_{R1} and p_{R2} respectively. For example, in Fig.3(top), the position of robot 3 is mutated with a large perturbation whereas in Fig.3(bottom), the same robot undergoes a slight perturbation.

If link mutation is selected, the mutations are applied on the links. There are three alternative operators and each is applied with probability p_{L1}, p_{L2} and p_{L3} respectively. First, a robot with one link only – known as leaf robot – is rotated through a random angle while perturbing the link length also slightly without breaking the link as shown in Fig. 4(top) for the link between robots 2 and 3. A second type of link mutation is where a randomly selected link is moved to a new location in W by keeping its length and direction. An example is presented in in Fig. 4(center) where the link between robots 2 and 3 is translated. The third type of mutation is where a randomly selected robot that also does not have any links is forced to be connected to a randomly chosen nearby robot. An example is as seen in Fig. 4(bottom) where robot 1 is made to establish a link with robot 2.

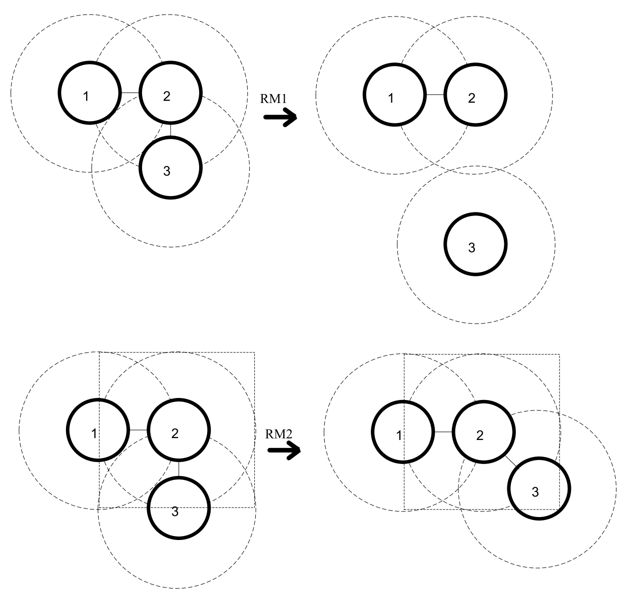

Fig. 3. Robot mutations: (Top) Translation within the workspace; (Bottom) Translation within a small area.

V. Simulations

In this section, we present simulation results from running the proposed algorithm for varying connectivity graphs. The values of all the parameters are set as presented in Table-I. The type of the crossover operation is determined statistically as follows: We made 100 simulations with a high number of robots, identical parameter set and adjacency matrix and with different $S(0)$ using only one of crossover operators. The performance of each operator is assessed based on the percentage of realizations found in these simulations which turn out to be 62, 65 and 70 percent for the single-point, multi-point and square crossover operators respectively. Hence,

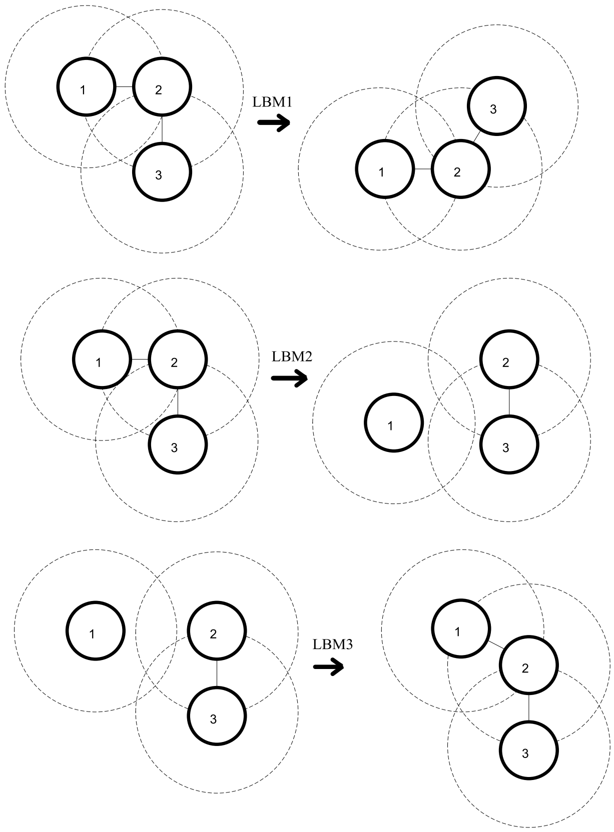

Fig. 4. Link-based mutations: (Top) Rotation based; (Center) Translation based; (Bottom) Connectivity based.

in the remaining simulations, the only type of crossover operator used is the square crossover.

We start with $p = 8$ since an example of an unrealizable graph is the star $K_{1,7}$ with one central node connected to seven leaves as given by A_1. It is well known this graph does not have a realization since using geometry, it can be seen that if each of seven unit disks touches a common unit disk, some two of the remaining seven disks must touch each other. A_2 is a realizable version of this graph. The algorithm does not generate a realization for A_1. For A_2, a sample graph is shown in Fig.5.

$$A_1 = \begin{bmatrix} 0&1&1&1&1&1&1&1 \\ 1&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0&1&1&1&1&1&1&1 \\ 1&0&1&0&0&0&0&1 \\ 1&1&0&1&0&0&0&0 \\ 1&0&1&0&1&0&0&0 \\ 1&0&0&1&0&1&0&0 \\ 1&0&0&0&1&0&1&0 \\ 1&0&0&0&0&1&0&1 \\ 1&1&0&0&0&0&1&0 \end{bmatrix}$$

Next, we run the algorithm for $p = 30$ robots and seek realizations of a graph having degree 3 with a simple connectivity structure shown in Fig. 6. Due to size, the corresponding adjacency matrix is not provided in the paper. We run 100 simulations all with different initial population set $S(0)$. Six sample realizations generated by the algorithm are as shown in Fig.7.

TABLE I

PARAMETERS OF GA

Parameters	Symbol	Value
Workspace radius	ρ_0	50
Robot radius	ρ_i	0.5
Connectivity range	ρ_c	20
Population size	N_P	60
Max. number of generation	N_G	10000
Crossover probability	p_C	0.3
Crossover square size	D_C	15
RM probability	p_R	0.5
RM1 probability	p_{R1}	0.2
RM2 probability	p_{R2}	0.2
Mutation square size	D_M	5
LBM probability	p_L	0.5
LBM1 probability	p_{L1}	0.2
LBM2 probability	p_{L2}	0.3
LBM3 probability	p_{L3}	0.3
Constancy generation for elite	N_E	1000
Random new addition percentage	p_A	33
Best population percentage	p_B	10

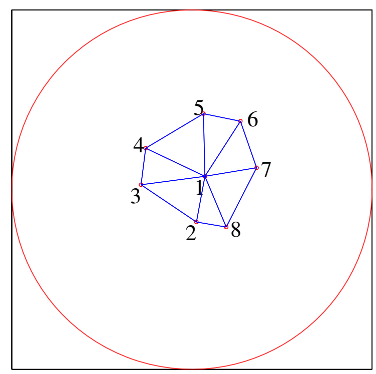

Fig. 5. A sample graph realization for 8 robots given A_2.

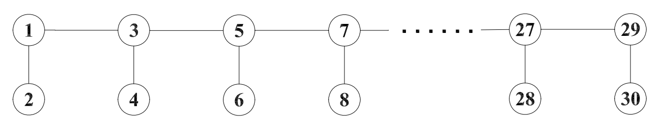

Fig. 6. A simple graph for 30 robots.

Finally, we run the algorithm again for $p = 30$ robots, but this time the realizations are of a graph having a more complicated connectivity as shown in Fig.8. Again 100 simulations with different initial population set $S(0)$ are made. Six sample realizations generated by the algorithm are as shown in Fig.9.

VI. CONCLUSION

This paper considers the problem of planar multi-robot realizations of connectivity graphs. A realization is a set of robot locations in the planar workspace having a connectivity graph that is identical to an *a priori* given connectivity graph with the additional constraint it must be feasible. As the associated mathematical problem is known to be NP-hard, a stochastic approach based on genetic algorithms is proposed. Here, a population set is generated based on randomly generated feasible planar multi-robot positions. Each member

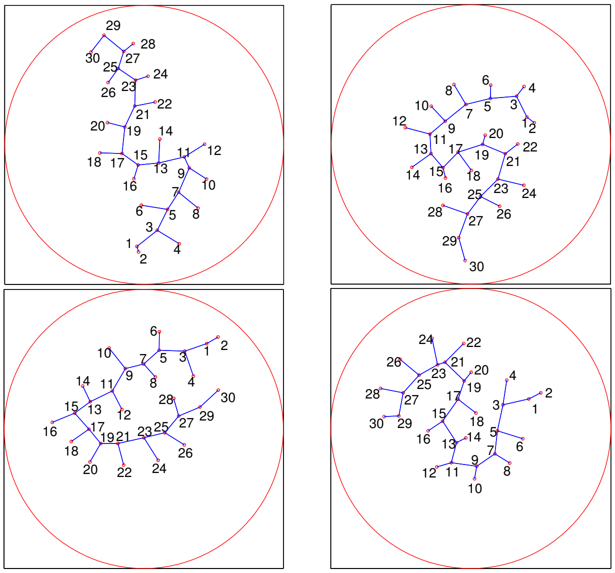

Fig. 7. Sample 30 robot realizations for a simple connectivity structure.

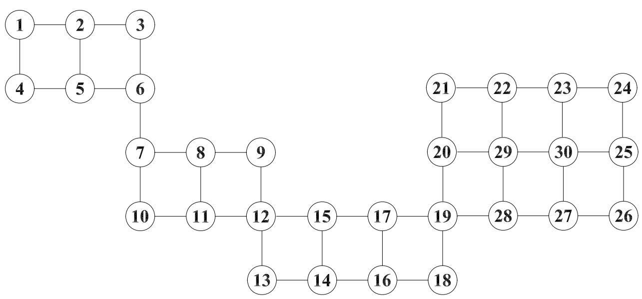

Fig. 8. A complicated connectivity graph for 30 robots.

in this set is then evaluated using a novel fitness function that measures the similarity of its connectivity graph with the given connectivity graph. New mutation operators that enable the evolution of generations are introduced. An extensive statistical study with different number of robots demonstrates that the proposed algorithm can be used to obtain realizations for fairly complicated connectivity graphs.

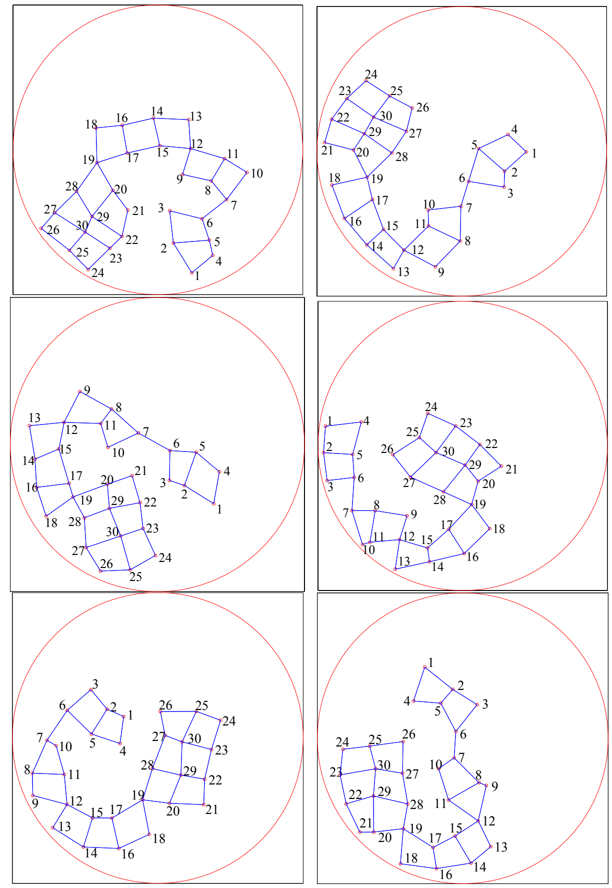

Fig. 9. Sample 30 robot realizations for the graph of Fig. 8.

ACKNOWLEDGMENT

This work has been supported by Bogazici University BAP Projects 09HA201D, 5169 and Tubitak Project MAG 107M240.

REFERENCES

[1] Bayram, H. and H.I. Bozma, "Multi-robot Navigation with Limited Communication - Deterministic vs Game-Theoretic Networks", *Proc. IEEE/RSJ Int. Conf. on Int. Robots and Systems*, 2010.

[2] Breu, H. and D.G. Kirkpatrick, "Unit disk graph recognition is NP-hard", *Computational Geometry*, vol. 9, pp. 3-25, 1998.

[3] Clark, Brent N., C. Colbourn, D.S. , "Unit disk graphs", *Discrete Math.*, vol. 86, no. 1-3, 165-177, 1990.

[4] Duda, R.O., P.E.Hart, D.G.Stork. Pattern Classification, Second Edition, John Wiley and Sons, Inc, 2001.

[5] Eloranta, T. and E. Makinen, "TimGA: A Genetic Algorithm for Drawing Undirected Graphs", *Divulgaciones Matematicas*, vol. 9, no. 2, pp. 155-171, 2001.

[6] Meshabi, M, "On State-Dependent Dynamic Graphs and Their Controllability Properties", *IEEE Trans. on Autom. Control*, vol. 50, no. 3, pp.387-392, 2005.

[7] Kamada, T, S. Kawai, "An algorithm for drawing general undirected graphs", *Information Processing Letters*, vol.31, pp.7-15, 1989.

[8] Kuhn, F. , T. Moscibroda and R. Wattenhofer, "Unit disk graph approximation", *Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications(DIAL-M)*, pp. 17-23, 2004.

[9] Muhammad, A. and M. Egerstedt,"Connectivity Graphs as Models of Local Interactions", *IEEE Conference on Decision and Control*, pp.124-129, 2004.

[10] Saber, R. and R. M. Murray, " Flocking with Obstacle Avoidance: Cooperation with Limited Communication in Mobile Networks", *Proc. 42nd IEEE Conference on Decision and Control*, pp.2022-2028, 2003.

[11] Zhang, Q. H. Liu, W. Zhang, Y. Guo, "Drawing Undirected Graphs with Genetic Algorithms", *Lecture notes in computer science*, vol. 3612, pp.28-36, 2005.

[12] Sahin, C.S., E. Urrea, M.U. Uyar, M. Conner,I. Hokelek, G.Bertoli and C. Pizzo, "Self-deployment of Mobile Agents in Manets for Military Applications", *Army Science Conference*, 2008.

[13] S.N.Sivanandam and S.N.Deepa, Introduction to Genetic Algorithms, Springer, 2008.