International Journal of Innovative Computing, Information and Control
Volume **9**, Number **1**, January **2013**
ICIC International ©2013 ISSN 1349-4198
pp. 305–318

MOTION PLANNING OF SWARM ROBOTS USING POTENTIAL-BASED GENETIC ALGORITHM

Chien-Chou Lin[1], Kun-Cheng Chen[1], Po-Yuan Hsiao[2] and Wei-Ju Chuang[1]

[1]Department of Computer Science and Information Engineering
National Yunlin University of Science and Technology
No. 123, Section 3, University Road, Douliou, Yunlin 64002, Taiwan
{ linchien; g9817704; g9917709 }@yuntech.edu.tw

[2]Department of Computer Science and Information Engineering
Shu-Te University
No. 59, Hengshan Road, Yanchao District, Kaohsiung City 82445, Taiwan
s97639112@stu.edu.tw

Received October 2011; revised March 2012

Abstract. *A potential-based genetic algorithm is proposed for the motion planning of robot swarms. The proposed algorithm consists of a global path planner and a motion planner. The global path planning algorithm plans a trajectory, which the robot swarm should follow, within a Voronoi diagram of the free space. The motion planning algorithm is a genetic algorithm based on artificial potential models. The potential functions are used to keep robots away from obstacles and to keep the robot swarm within a certain distance from each other. Since the proposed approach is a hierarchical algorithm which plans the global path and local motion individually, the robot swarm moves toward the goal by sequentially traversing a sequence of positions along the Voronoi diagram. Therefore, the robot swarm can avoid becoming trapped in local minima.*
Keywords: Swarm robotics, Formation control, Voronoi diagram, Artificial potential field, Genetic algorithm

1. **Introduction.** An increasing number of multi-robot systems have been proposed in recent decades. Swarm robotics is an approach for coordinating multi-robot systems [1-29]. The swarm shares information about the environment and individual members interact with each other. Cooperative behavior may be used to complete tasks, e.g., surveillance, search and rescue. In order to perform these tasks, the robot swarm system should have the capabilities: forming dynamically whenever communication capabilities permit and planning paths for all robots. Therefore, a feasible motion planner should keep every robot connected. Therefore, the proposed algorithm should maintain the connectivity of swarm robots that have limited ranges of communication. The communication network of robots is formed if any two robots in a swarm can connect directly or via a serious connection of other robots.

In general, robot swarm systems can be classified into two basic types: homogeneous systems and heterogeneous systems. In a homogeneous robot system, all robots are the same structure and they have the same capability. Robots can communicate with other robots to share information for decision making. In contrast, in a heterogeneous system, robots have unique functions and play different roles. For example, the swarm may have a central controller robot which plans the path and issues commands to other robots. The present study adopts a heterogeneous robot system.

Most studies on robot swarm cooperation have focused on formation control, which refers to the task of controlling a group of mobile robots to follow a predefined path or trajectory while maintaining the desired formation pattern, as shown in Figure 1. Generally, literals of path planning can be categorized in two basic problem types: holonomic and non-holonomic. In this paper, the focus is on the path planning problem of a holonomic robot swarm. All of the assumptions are the same as those used in other literals of holonomic systems. In practice, a holonomic system should fully stop when changing its direction while following the planned path, whereas a non-holonomic system takes the velocity constraint and the minimum curvature into account when planning a path. Numerous methods have been proposed, which can be roughly categorized into three basic approaches: behavioral, virtual structure, and leader-follower.

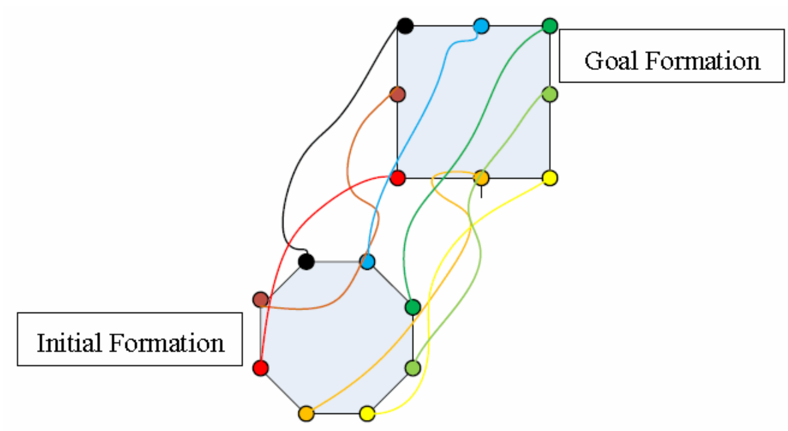


FIGURE 1. A group of mobile robots follows a predefined path or trajectory while maintaining the desired formation pattern.

In virtual structure approaches, the robot swarm is considered as a single rigid robot. A rigid geometric relationship among group members is maintained [1-7]. Therefore, the path planning of a robot swarm can be simplified as the path planning of a rigid robot [1]. The advantage of the virtual structure approach is ease of implementation. However, the approach has low path planning flexibility. In [2,3,5], virtual structure approaches were proposed for coordinating multiple spacecraft. In [7], the algorithm derives scale-parameterized interpolated trajectories for a team of fully actuated mobile robots. The scale parameter controls the distances between robots and minimizes the overall energy consumption due to motion.

For behavior-based approaches [8-17], several desired behaviors, i.e., movement towards goal, obstacle avoidance, collision avoidance, and keeping formation, are defined for each robot to create its trajectory. The planning of robots can be done concurrently. Since each robot is considered individually, it is difficult to guarantee precise formation control. In [15], formation control comprised a sequence of maneuvers between formation patterns. The algorithm consists of three strategies: (i) using relative position information configured in a bidirectional ring topology to maintain the formation, (ii) injecting inter-robot damping via passivity techniques, and (iii) accounting for actuator saturation.

In the leader-follower approach [18-28], the ability of a robot depends on its job. In the swarm, one or a few robots act as leaders which move along predetermined trajectories, and other robots in the group follow while maintaining the desired relative position with respect to the leader. Generally, leader-follower-based robot systems are implemented as centralized systems. However, most leader-follower approaches are not complete algorithms because the safe path, which gives a robot sufficient distance from obstacles and other robots, is difficult to derive. In [28], the centralized leader-follower formation control algorithm uses panoramic vision to maintain the swarm formation.

In order to obtain a safe path for swarm robots, the present paper proposes a hierarchical path planning algorithm. The proposed algorithm consists of a global path planner and a motion planner. The global path planning algorithm searches for a path, which the center of the robot swarm should move along, within a Voronoi diagram of the free space. The motion planner is a genetic algorithm based on an artificial potential field. The potential functions are used to keep robots away from obstacles and to keep the robot swarm within a certain distance from each other. With the potential models, the local motion paths derived by the genetic algorithm are safe in terms of being collision free and away from obstacles. The proposed algorithm is a complete algorithm.

The rest of this paper is organized as follows. In Sections 2 and 3, the proposed global and local path planning algorithms are introduced respectively. In Section 4, simulation results and the performance of the algorithm are presented. Finally, conclusions and possible future work directions are given in Section 5.

2. **Global Path Planning.** Due to the coupling of global planning and local planning, many existing algorithms require complex search processes and suffer from local minimum problems. The proposed algorithm consists of a global path planner (GPP) and a motion planner (MP). The former determines the primary movement direction of the swarm robots and the latter derives a configuration of robots. Details of the two planning algorithms are given below.

Global path planning can be considered as a planning problem for a point robot. In Figure 2, a swarm of two robots moves to the goal configuration; the planned path is close to obstacles [30]. In order to obtain a safe path, a Voronoi diagram (VD) is adopted since it is easy to implement and has been shown to work well in many cases.

There are many variants of VD [31-34]. In the present study, a VD consisting of line segments [33] is considered. A VD shows a set of free points which are equidistant to two closest obstacles.

In [34], the VD is constructed using Voronoi vertices and Voronoi arcs. The Voronoi vertices in this case are points equidistant to the closest features of three (or more) polygons. The vertices are connected by continuous chains of Voronoi arcs. An arc may be equidistant to two closest vertices or to two closest obstacle edges or to an obstacle vertex and an obstacle edge.

As shown in Figure 3, all edges and vertices of obstacles are used to construct the VD. The computation complexity is proportional to the total number of features of obstacles. Only a partial VD is used for global path planning for swarm robots. An efficient approach for constructing the partial VD is proposed in this paper.

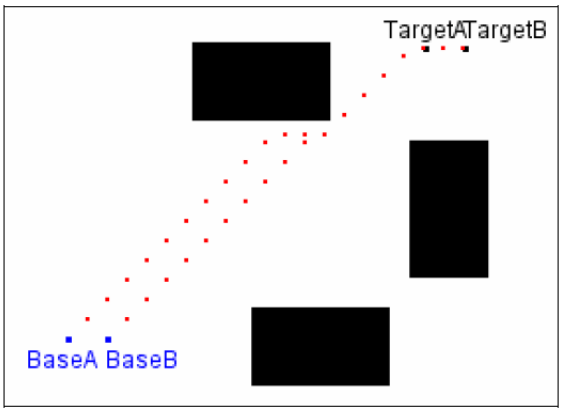


FIGURE 2. Simple path from start to goal with obstacles

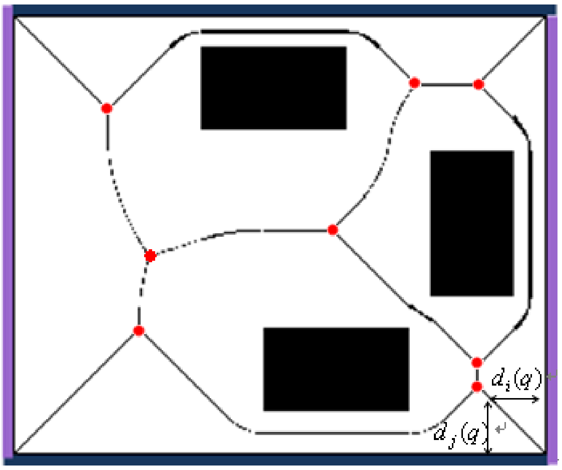


FIGURE 3. Voronoi diagram for Figure 2

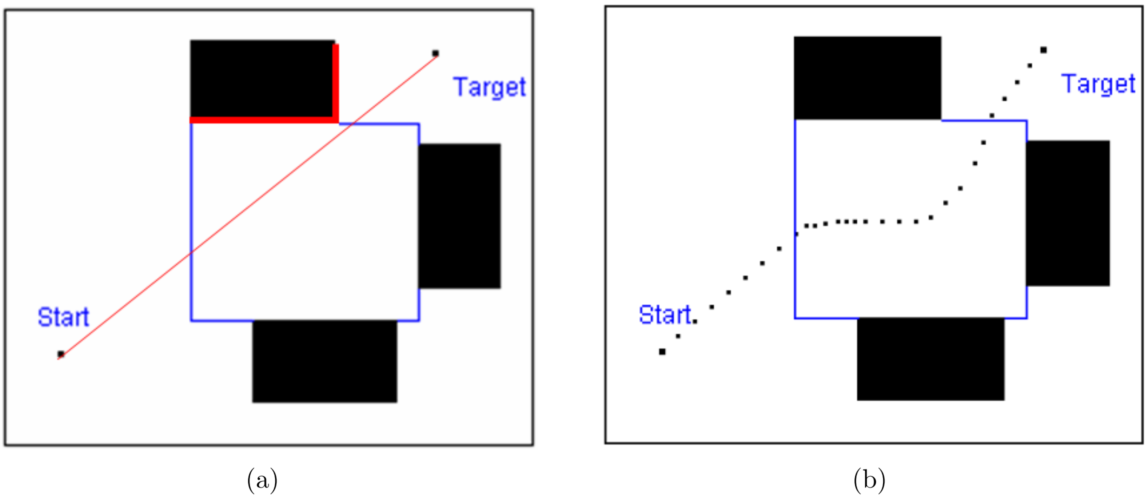


FIGURE 4. Proposed approach where Voronoi vertices are constructed from obstacles near the straight line from start to goal and connected by a Voronoi arc

Unlike approaches which construct the whole Voronoi diagram of the free space and then search for the path, the proposed scheme constructs a partial VD of the region of interest. As shown in Figure 4(a), the proposed approach explores Voronoi vertices constructed from obstacles which are near the straight line from start to goal. Then, the Voronoi vertices are connected by a Voronoi arc which is formed by the nearest edges along the line, as shown in Figure 4(b). The approach significantly reduces the computation complexity. Since a VD is the medial axis of the free space, the global path derived using a VD for swarm robots is the safest path.

3. **Local Motion Planning.** The obtained global path can be sampled as a series of positions, denoted as $(q_1, q_2, q_3, \ldots, q_n)$, which the center of the robot swarm should follow. For each two adjacent positions q_i and q_{i+1}, the GA-based local motion planner plans a set of collision-free paths for robots.

The GA searches for optimal solutions over the whole solution space and thus avoids being trapped in local minima. To search for the optimal positions for robots by a GA-based algorithm, the coordinate displacements of the robot swarm are encoded into one chromosome, denoted as $((x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k))$. The chromosome is then constructed as shown in Figure 5. The proposed evolutionary algorithm is described below.

Algorithm Non-random Initial GA for Swarm Robots

Begin
  $\boldsymbol{i = 1}$**; /* Initialize intermediate goal */**
  $\boldsymbol{t = 0}$**; /* Initialize generations */**
  Randomly generate initial population $\boldsymbol{P_i(t)}$;
  fitness$(\boldsymbol{P_i(t)})$;
  repeat until $(\boldsymbol{q_i = q_n})$ **Do**
    $\boldsymbol{P_{i+1}(t) = P_i(t)}$;
    repeat until (reach $\boldsymbol{q_i}$**) Do**
    select $\boldsymbol{P_i(t+l)}$ **from** $\boldsymbol{P_i(t)}$;
    crossover$(\boldsymbol{P_i(t+l)})$;
    mutate $(\boldsymbol{P_i(t+l)})$;
    fitness$(\boldsymbol{P_i(t+l)})$;
    $\boldsymbol{t = t + 1}$**;**
    end
    $\boldsymbol{i = i + 1}$**;**
  end
End

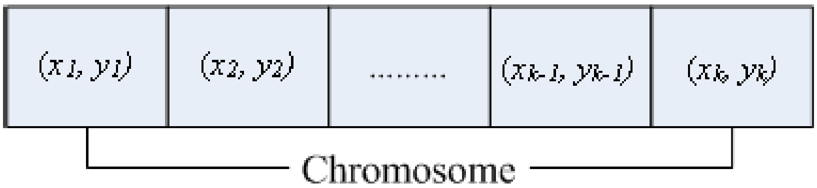


FIGURE 5. Configuration of robot swarm represented as a chromosome with genes as coordinate displacements

The basic procedure of the proposed algorithm can be simplified as the following five steps:

i. Step 0: Initialization of population. Population is initialized for q_i. If $i = 0$, the initial population is generated randomly; otherwise, the initial population is the offspring of q_{i-1}.
ii. Step 1: Selection. Population is sorted by fitness and the top 10% of chromosomes are preserved for crossover and mutation.
iii. Step 2: Crossover. The crossover operation generates new offspring from two parent chromosomes.
iv. Step 3: Mutation. The mutation operation locally adjusts genes with different probabilities to generate new offspring.
v. Step 4: Evaluation of population. If the optimal solution is found, $i = i+1$ is set Step 0 is repeated to begin a new evolution for the next intermediate goal q_{i+1}; otherwise, Step 1 is repeated to begin the next generation.

The implementation details of the five steps are shown below.

A. Step 0: Initialization of Population

The population, $P_1(0)$, of the first intermediate goal is generated randomly. The initial populations, $(P_i(0), i > 1)$, of other intermediate goals are partially obtained from the last generation of the preceding intermediate goal and partially randomly generated. Since these initial populations are eugenic and inherit from ancestors, the evolution time is reduced.

B. Step 1: Selection

The aim of selection is to preserve the optimal chromosomes and abandon the suboptimal chromosomes. Generally, selection is performed according to the fitness of every chromosome, where the fitness evaluation of the GA is an objective function for chromosomes. Generally, there are several types of selection: roulette, tournament, best, random, top percent. In the present study, the top percent scheme is adopted. The top 10 percent of the population is reserved as the next generation's population and others are selected randomly.

C. Crossover Operator

The reproduction operators, including crossover and mutation, generate new offsprings for next generation. Crossover is performed between two selected chromosomes, called parents, by exchanging parts of their genomes to form two new chromosomes, called offspring. The most popular types of crossover operation are one-point, two-point, uniform, and blending. In the present study, since the ith gene of a chromosome represents the position of robot i, the crossover operator exchanges similarly positioned genes of a pair of chromosomes, as shown in Figure 6.

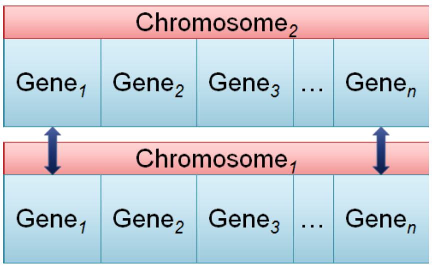


FIGURE 6. Crossover operator where similarly positioned genes of a pair of chromosomes are exchanged

D. Mutation Operator

The mutation operator changes an arbitrary bit in a genetic sequence with some probability. Mutation maintains genetic diversity from one generation of a population of chromosomes to the next while attempting to avoid local minima.

E. Evaluation Criteria

Generally, selection is conducted according to the fitness of every chromosome, where the fitness evaluation of the GA is an objective function for chromosomes. The trivial definition of a fitness function is the distance between the robots and the goal. Therefore, the fitness function of a configuration can be defined as:

$$V_q = \sum_{i=1}^{k} D_q^i \tag{1}$$

where D_q^i is the distance between robot i and the intermediate goal of the swarm center, q. The distance can be simplified as:

$$\begin{aligned} D_q^i &= \sqrt{(T_x - B_x^i)^2 + (T_y - B_y^i)^2} \\ &\approx \left|T_x - B_x^i\right| + \left|T_y - B_y^i\right| \end{aligned} \tag{2}$$

where (T_x, T_y) is the coordinator of the intermediate goal and (B_x, B_y) is the coordinator of robot i. An optimal chromosome is one that represents a configuration of swarm robots that is collision-free and reaches the goal. Motions that lead to collisions with obstacles or other robots should be removed. Thus, the fitness function is defined as:

$$V_q = f_{collide}(q) \times \left(\sum_{i=1}^{k} D_q^i + \rho U_{rep}(q) \right) \tag{3}$$

where ρ is a constant and $U_{rep}(q)$ is the repulsive potential of swarm robots from obstacles. When a configuration leads to a collision with obstacles, the collision function, $f_{collide}(q)$, is equal to $V_{\max}$, which is a penalty; otherwise, it is equal to 1.

The potential $U_{rep}(q)$ can be calculated analytically [11,23] as:

$$U_{rep}(q) = \sum_{i=1}^{k} U_{rep}^i(q) \tag{4}$$

$U_{rep}^i(q)$ is the repulsive potential of robot i from the nearest obstacle.

$$U_{rep}^i(q) = \begin{cases} \frac{1}{2}\eta \left(\frac{1}{Disp(i)} - \frac{1}{Q^*} \right)^2, & Disp(i) \le Q^* \\ 0, & Disp(i) > Q^* \end{cases} \tag{5}$$

where Q^* is the minimum distance from obstacles and η is a gain of the repulsive gradient. $Disp(i)$ is the distance between robot i and the closest obstacle.

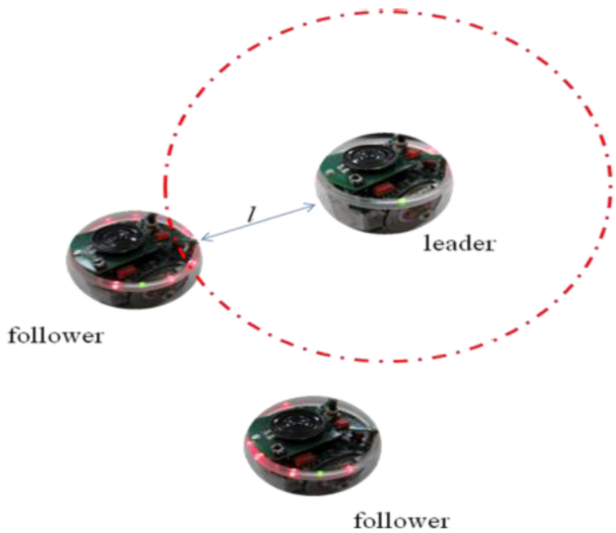


FIGURE 7. Swarm cohesion considered as a spring equation

For swarm cohesion, a robot in the swarm should keep a certain distance from the swarm center and not stray far from other robots. Thus, a spring function is adopted as a repulsive/attractive potential function in the fitness function. The fitness function can be rewritten as:

$$V_q = f_{collide}(q) \times \left(\sum_{i=1}^{k} D_q^i + \rho U_{rep}(q) + \frac{1}{2}\kappa \sum_{i=1}^{n} X_i^2 \right) \tag{6}$$

with

$$X_i^2 = (l - l_i)^2 \tag{7}$$

where l_i is the distance between robot i and the nearest neighbor robot and l is the safe distance which should be kept between robots. The desired formation of the swarm robots can be maintained using the spring function. The selection stage of the GA is performed according to the fitness function in (6).

4. **Results.** The GPP of the proposed algorithm was implemented using a modified Voronoi algorithm. The MP of the proposed algorithm was implemented using the JGAP API, an open source Java-based genetic algorithm and genetic programming package. The simulations were run on a PC with a Core2 Duo 1.7-GHz CPU running the Windows XP operating system. The population was 100 and the maximum number of generations was set to 400. 10% of the initial population was non-random in the simulations. The probabilities of mutation and crossover were both 10%. The safe distances, l and Q^*, were set to 7 pixels and 5 pixels, respectively.

Figure 8 shows an example of a 3-robot swarm. The initial positions and the goal positions are shown in the bottom-left and upper-right corners of Figure 8, respectively. The configurations are represented as triangles. The formation of swarm robots is maintained from start to goal, but the size of the triangles varies with respect to the environment.

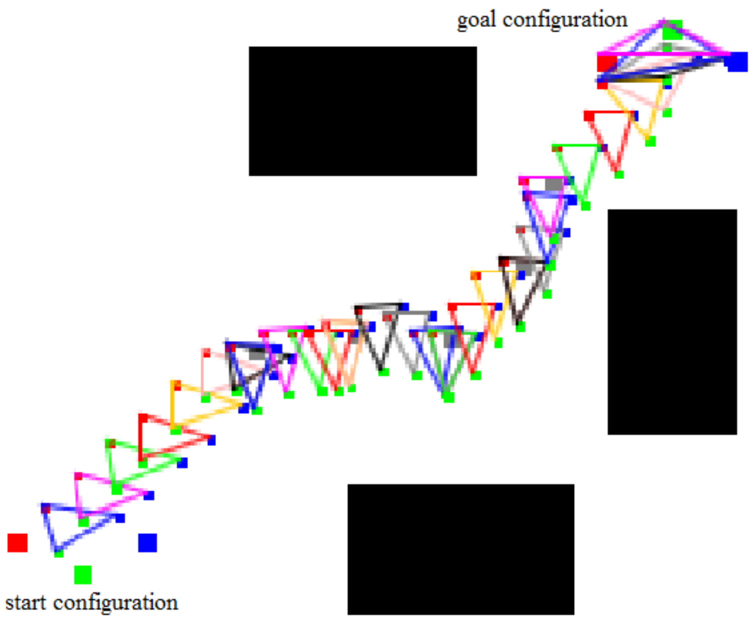


FIGURE 8. Configurations are represented as triangles.

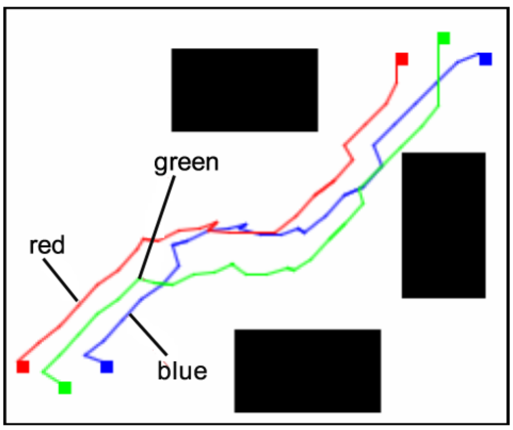


FIGURE 9. Three trajectories for three-robot swarm example ($Q^* = 5$)

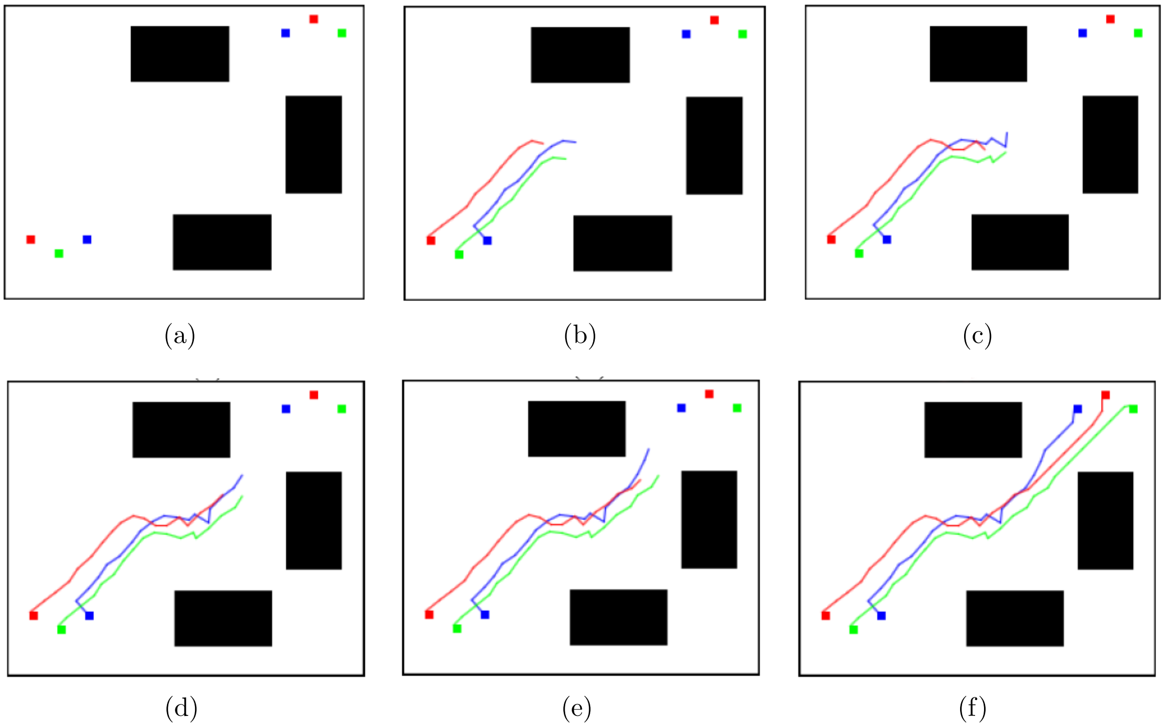


FIGURE 10. Three trajectories for three-robot swarm example ($Q^* = 7$)

Figure 9 shows the planned trajectories for the three robots. Two of the paths (blue and red) are safe. The unsafe path can be made safe by increasing Q^*. Figure 10 shows the simulation results for $Q^* = 7$. The new green path is away from the obstacle.

In Figure 9 and Figure 10, the three final trajectories for the robots intersect each other. However, the paths are collision-free since the robots pass the intersections at different times. In Figure 11, the intersections have been removed from the trajectories. Since the three trajectories do not intersect, the planned trajectories are collision-free paths.

Figure 12 shows the initial configuration and final trajectories for a four-robot swarm moving in a narrow passage. The swarm robots traverse along the path of the VD. The simulation took 5.8688 seconds to plan the 53-configuration collision-free path. The computation time depends on the complexity of the path.

Figure 13 shows the initial configuration and final trajectories obtained for a five-robot swarm. The simulation took 12.2237 seconds to plan the 125-configuration collision-free path. Since the swarm has more robots and more DOFs, the chromosomes are longer. For a given population size, it is more difficult to find a feasible configuration for larger swarms. Therefore, the obtained path is more winding.

Figure 14 shows the initial configuration and the final trajectories for an eight-robot swarm. The simulation took 65.9157 seconds to plan the 249-configuration collision-free path. The larger a robot swarm is, the more complex the computation is. In this example, the swarm takes more configurations for each step because the swarm robots have to locally adjust to get a set of collision-free paths. It is obviously that the obtained paths are more winding than previous examples. The path can be improved by increasing the population size, which will also reduce the number of obtained configurations. Figure 15 shows the fitness value of configurations for various maximum population sizes. In Figure 15(b), the simulation with a population of 1000 is the first to reach the goal. And, the simulation with a population of 100 takes the most configurations to reach the

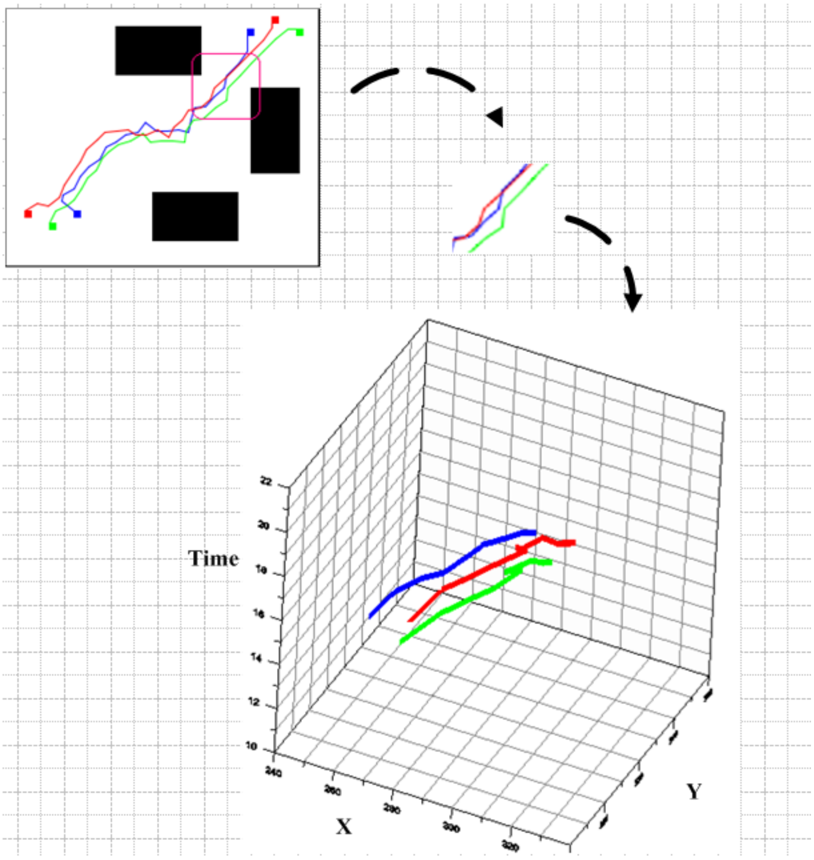
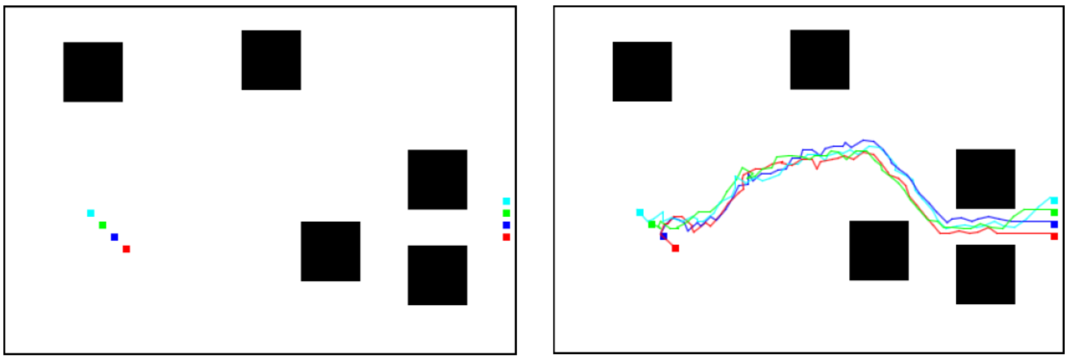


FIGURE 11. Trajectories without intersections

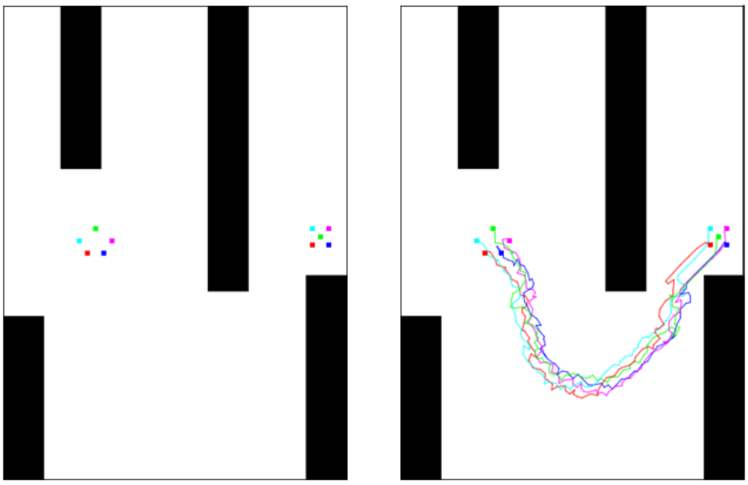


(a) (b)

FIGURE 12. Four-robot swarm example: (a) initial configuration and (b) final trajectories

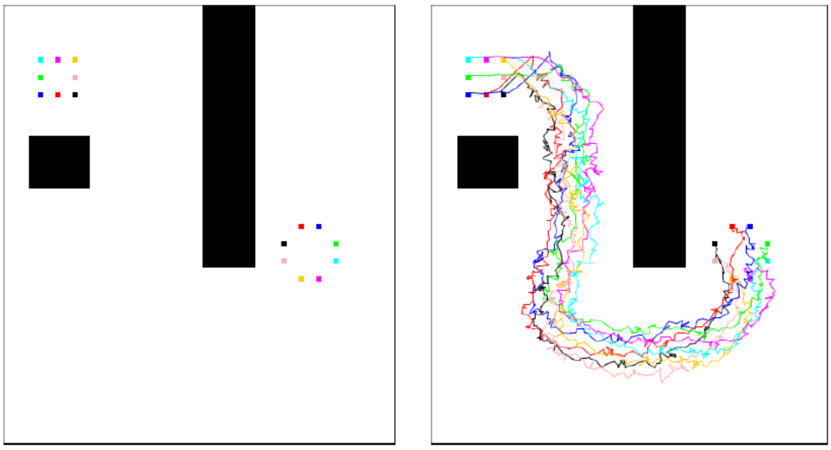
goal in Figure 15(c). The size of the maximum population also affects the speed of the optimization of configurations.

5. **Conclusions.** A potential-based genetic algorithm was proposed for the formation control of swarm robots. The proposed algorithm consists of a global path planner (GPP) and a motion planner (MP). The GPP searches for a path, which the swarm robots should follow, from the start to the goal within a Voronoi diagram of the workspace. The MP is a GA-based planner based on an artificial potential field. The repulsion keeps robots away from obstacles and the spring function maintains the robot swarm within



(a) (b)

FIGURE 13. Five-robot swarm example: (a) initial configuration and (b) final trajectories

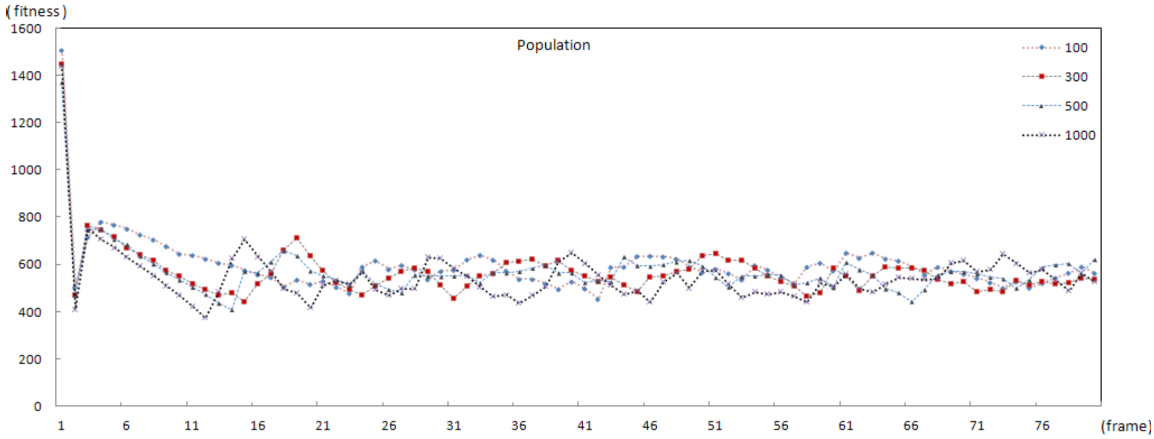


(a) (b)

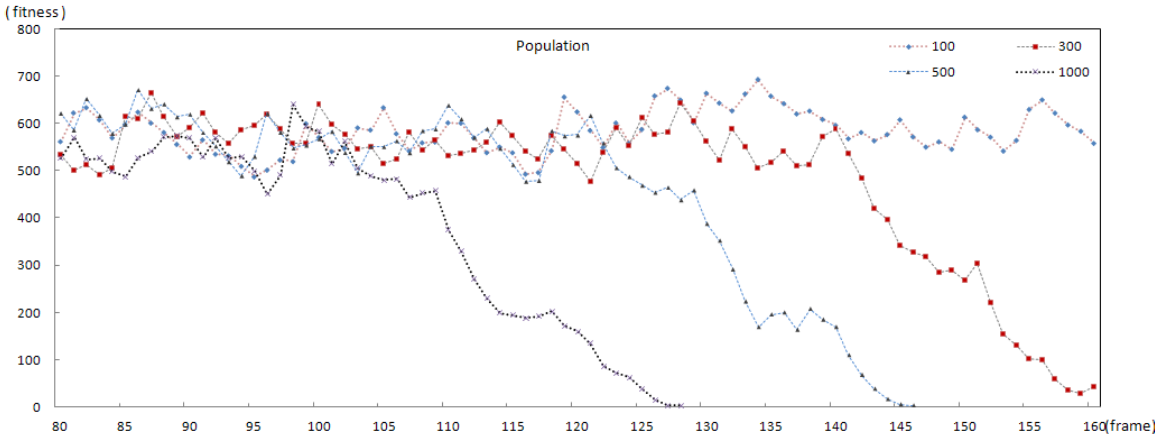
FIGURE 14. Eight-robot swarm example: (a) initial configuration and (b) final trajectories

a certain distance from each other. Since the GA searches for an optimal configuration which has lower potential, the obtained paths are safe. Since the proposed approach is a hierarchical algorithm which plans the global path and local motion individually, the robot swarm moves toward the goal by sequentially traversing a sequence of positions along the Voronoi diagram. Therefore, the robot swarm can avoid becoming trapped in local minima. Simulation results demonstrated that the proposed algorithm can plan collision-free paths for swarm robots. As for the computation complexity, because chromosomes are composed of the coordinate displacements of the robot swarm, the search space is downsized significantly. Therefore, an optimization solution can be found in a reasonable time.

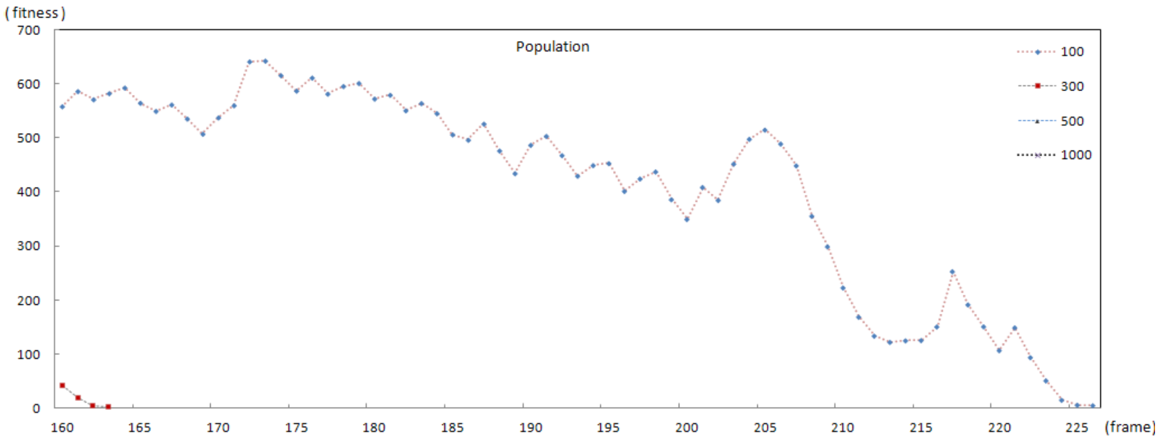
The path planning problem for swarm robots was considered for 2-D workspaces. The proposed algorithm can be extended to 3-D workspaces without significant modification. For example, the gene of a robot can be represented as (x, y, z).



(a)



(b)



(c)

FIGURE 15. Fitness value of configurations for various population sizes

In future works, we will focus on smoothing the planned paths to reduce redundant movements, especially for 8-robot swarms. Some constraints may be considered in the planning stage, e.g., a minimum curvature constraint. With this modification, the proposed algorithm should be more efficient.

Acknowledgment. This work was supported by the National Science Council of Taiwan under grants NSC 98-2220-E-224-009 and NSC 99-2220-E-224-007.

REFERENCES

[1] M. A. Lewis and K.-H. Tan, High precision formation control of mobile robots using virtual structures, *Autonomous Robots*, vol.4, no.4, pp.387-403, 1997.

[2] R. W. Beard and F. Y. Hadaegh, Constellation templates: An approach to autonomous formation flying, *Proc. of World Automation Congress*, Alaska, pp.177.1-177.6, 1998.

[3] A. W. Beard, J. Lawton and F. Y. Hadaegh, A coordination architecture for spacecraft formation control, *IEEE Transactions on Control Systems Technology*, vol.9, no.6, pp.777-790, 2001.

[4] T. Eren, P. N. Belhumeur and A. S. Morse, Closing ranks in vehicle formations based rigidity, *Proc. of IEEE Conference Decision Control,* Las Vegas, pp.2959-2964, 2002.

[5] R. Wei and R. W. Beard, A decentralized scheme for spacecraft formation flying via the virtual structure approach, *Proc. of Conference the American Control*, Denver, Colorado, pp.1746-1751, 2003.

[6] Y.-S. Zhang, F. B. Bastani and I-L. Yen, Decentralized path planner for multi-robot systems, *IEEE International Conference on Tools with Artificial Intelligence*, pp.171-175, 2009.

[7] C. Belta and V. Kumar, Optimal motion generation for groups of robots: A geometric approach, *Journal of Mechanical Design*, vol.126, pp.63-70, 2004.

[8] Q. Chen and J. Y. S. Luh, Coordination and control of a group of small mobile robots, *Proc. of IEEE International Conference on Robotics and Automation*, pp.2315-2320, 1994.

[9] K. Sugihara and I. Suzuki, Distributed algorithms for formation of geometric patterns with many mobile robots, *Journal of Robotic Systems,* vol.13, no.3, pp.127-139, 1996.

[10] T. Balch and R. C. Arkin, Behavior-based formation control for multi-robot teams, *IEEE Transactions on Robotics and Automation*, vol.14, pp.926-939, 1998.

[11] M. Schneider-Fontan and M. J. Mataric, Territorial multi-robot task division, *IEEE Transactions on Robotics and Automation*, vol.14, pp.815-822, 1998.

[12] L. E. Parker, ALLIANCE: An architecture for fault-tolerant multi-robot cooperation, *IEEE Transactions on Robotics and Automation*, vol.14, pp.220-240, 1998.

[13] M. Veloso, P. Stone and K. Han, The CMUnited-97 robotic soccer team: Perception and multi-agent control, *Robotics and Autonomous Systems*, vol.29, pp.133-143, 1999.

[14] S. Monteiro and E. Bicho, A dynamical systems approach to behavior-based formation control, *Proc. of IEEE International Conference on Robotics Automation*, pp.2606-2611, 2002.

[15] J. R. T. Lawton, R. W. Beard and B. J. Young, A decentralized approach to formation maneuvers, *IEEE Transactions on Robotics and Automation*, vol.19, no.6, pp.933-941, 2003.

[16] V. Gazi, Swarm aggregations using artificial potentials and sliding-mode control, *IEEE Transactions on Robotics*, vol.21, no.6, pp.1208-1214, 2005.

[17] G. Baldassarre, V. Trianni, M. Bonani, F. Mondada, M. Dorigo and S. Nolfi, Self-organized coordinated motion in groups of physically connected robots, *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol.37, no.1, pp.224-239, 2007.

[18] M. Egerstedt and X. Hu, Formation constrained multi-agent control, *IEEE Transactions on Robotics and Automation*, vol.17, no.6, pp.947-951, 2001.

[19] H. G. Tanner, G. J. Pappas and V. Kumar, Leader-to-formation stability, *IEEE Transactions on Robotics and Automation*, vol.20, no.3, pp.443-455, 2004.

[20] W. Ren and R. W. Beard, Consensus seeking in multiagent systems under dynamically changing interaction topologies, *IEEE Transactions on Automatic Control*, vol.50, no.5, pp.655-661, 2005.

[21] Y. Q. Chen and Z. Wang, Formation control: a review and a new consideration, *Proc. of IEEE/RSJ International Conference Intelligent Agents and Systems*, pp.3181-3186, 2005.

[22] W. Burgard, M. Moors, C. Stachniss and F. E. Schneider, Coordinated multi-robot exploration, *IEEE Transactions on Robotics*, vol.21, no.3, pp.376-386, 2005.

[23] L. Moreau, Stability of multiagent systems with time-dependent communication links, *IEEE Transactions on Automatic Control*, vol.50, no.2, pp.169-182, 2005.

[24] S. Kloder and S. Hutchinson, Path planning for permutation-invariant multi-robot formations, *IEEE Transactions on Robotics*, vol.22, no.4, pp.650-665, 2006.

[25] L. Vig and J. A. Adams, Multi-robot coalition formation, *IEEE Transactions on Robotics*, vol.22, no.4, pp.637-649, 2006.

[26] R. Olfati-Saber, Flocking for multi-agent dynamic systems: Algorithms and theory, *IEEE Transactions on Automatic Control*, vol.51, no.3, pp.401-420, 2006.

[27] J. Shao, G. Xie and L. Wang, Leader-following formation control of multiple mobile vehicles, *IET Control Theory Appl.*, vol.1, no.2, pp.545-552, 2007.

[28] G. L. Mariottini, F. Morbidi, D. Prattichizzo, G. J. Pappas and K. Daniilidis, Leader-follower formations: Uncalibrated vision-based localization and control, *Proc. of IEEE International Conference on Robotics Automation*, pp.2403-2408, 2007.
[29] M. Kloetzer and C. Belta, Temporal logic planning and control of robotic swarms by hierarchical abstractions, *IEEE Transactions on Robotics*, vol.23, no.2, pp.320-330, 2007.
[30] G. Erinc and S. Carpin, A genetic algorithm for nonholonomic motion planning, *Proc. of IEEE International Conference on Robotics and Automation*, Roma, Italy, pp.1843-1849, 2007.
[31] O. Takahashi and R. J. Schilling, Motion planning in a plane using generalized Voronoi diagrams, *IEEE Transactions on Robotics and Automation*, vol.5, no.2, 1989.
[32] R. Mahkovic and T. Slivnik, Generalized local Voronoi diagram of visible region, *Proc. of IEEE International Conference on Robotics Automation*, Leuven, Belgium, pp.349-355, 1998.
[33] H. Choset, K. M. Lynch, S. Hutchinson, G. kantor, W. Burgard, L. E. Kavraki and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, Cambridge, 2005.
[34] R. Wein, J. P. van den Berg and D. Halperin, The visibility – Voronoi complex and its applications, *Computational Geometry*, vol.36, pp.66-87, 2007.