*24th International Symposium on Automation & Robotics in Construction (ISARC 2007)*
*Construction Automation Group, I.I.T. Madras*

# INITIALIZATION OF ROBOTIC FORMATIONS USING DISCRETE PARTICLE SWARM OPTIMIZATION

**V.T. Ngo**
ARC Centre of Excellence for Autonomous Systems, Faculty of Engineering
University of Technology, Sydney
PO Box 123 Broadway NSW 2007 AUSTRALIA
ngo652@eng.uts.edu.au

**N. M. Kwok and Q. P. Ha**
ARC Centre of Excellence for Autonomous Systems, Faculty of Engineering
University of Technology, Sydney
PO Box 123 Broadway NSW 2007 AUSTRALIA

## ABSTRACT

In this paper, a Discrete Particle Swarm Optimization (DPSO) technique is applied as an optimal motion planning strategy for the initialization of mobile robots to establish some geometrical patterns or formations. Firstly, an optimal assignment of location and orientation for each robot in the formation is performed. For this, a total travelling cost is defined, involving the angular and translational movement by all robots from their initial positions in the workspace to their goal positions. The objective is then to determine the position for each robot in the formation in order to minimize the cost function to form a predefined shape. Once each robot has been assigned with a desired position, a search scheme is implemented to obtain a collision free trajectory for each robot to establish the formation. Simulation results are presented to illustrate the validity of the proposed approach.

## KEY WORDS

Motion Planning, Robotic Formation Initialization, Position Assignment, Discrete Particle Swarm Optimization.

## 1. INTRODUCTION

Robotic formation is defined as the coordination of a group of mobile robots to get into and maintain a desired pattern with a certain geometric shape. Multi-robot coordination has recently received a considerable interest as various applications can be performed faster and more efficiently with multiple robots than with single robots operating independently. Applications requiring multiple robot coordination include mine sweeping [1], moving large objects in a construction site or military scouts [2-4], where each robot needs to be in a particular position corresponding to its sensing capacity or geometrical advantage to perform a given cooperative task. Such problems as initializing the formation, maintaining the formation configuration, avoiding static or moving obstacles in the workspace, and changing formation shapes to fulfil a specific task or to deal with obstacles remain important issues in robotic formation control [5].

Several approaches to solve one or more of the above subtasks can be found in the literature [6-9]. In [10], a directed potential field method was used for motion coordination of robots moving in formations, however, the inherent drawback of potential field method in the cases of local optima was not mentioned. An architecture for robotic formation control was proposed in [5] to solve the four main issues mentioned above. The authors focused on the problem of how to assign the most suitable position for each robot in the formation using a bounded depth-first search with a pruning algorithm to minimize a cost function compromising the level of a follower robot in a formation, the distance and heading deviation of a follower with respect to its leader robot. The search is distributed over every robot in the formation and the result with the lowest cost is chosen. This method is somewhat similar to the work proposed by [11], in which a robot in the group plans its own path independently and a D* search in a coordination diagram is performed to obtain the minimum cost trajectory for each robot in a multi-robot system. Obviously, a search process for this purpose would require a suitable optimisation technique.

Particle Swarm Optimization (PSO), originally developed by Kennedy and Eberhart [12], has proven to be of great potential for optimization applications and has been used successfully in robotics. For example, PSO was used for target searching problems with single and multiple target cases [13]. In [14], a modified PSO algorithm was introduced to find an optimal path for mobile robots. In construction automation, a PSO-based has been recently proposed for the coordination of a group of robotic vehicles [15]. Dealing in a similar application, the Generalized Discrete Particle Swarm Optimization (DPSO), proposed by Clerc [16], was applied for the Travelling Salesman Problem. The author pointed out that DPSO is easily implemented for discrete or combinational problems, particularly when a sufficiently good specialized algorithm is unavailable.

This paper addresses the problem of how to initialize a robotic formation by seeking an optimal position assignment for each robot in the formation. Each robot will be designated a desired position while minimizing the total cost which includes the translational and angular movement of all robots in the formation. The workspace is assumed to be obstacle-free so the shortest path for each robot from its initial position to its assigned position will be a straight line connecting the two positions. As each robot moves to its goal, collision between two or more robots may occur. We propose a search scheme to construct the velocity profile for those robots which may potentially collide with the others.

The rest of the paper is organized as follows. In Section 2, a brief review of DPSO is presented. The proposed robotic formation initialization approach is described in Section 3. Section 4 gives some simulation results to demonstrate the validity of the approach. A conclusion is given in Section 5 of the paper.

## 2. DISCRETE PARTICLE SWARM OPTIMIZATION

In Particle Swarm Optimization, a set of moving particles are initially thrown into the search space. Each particle, having a position and a velocity, knows its own position and the fitness function to evaluate its solution quality. Each particle randomly searches through the problem space by updating its own memory with its position and the social information gathered from other particles. At each time step, the movement of a particle is a compromise of three behaviours: to move randomly (its own way), to go towards its best previous position (memory), and to go towards its neighbour (global) best position. This evolutionary selection is described by the following equations for the $i^{th}$ particle:

$$\begin{cases} v_{i,new} = c_1 \otimes v_{i,old} + c_2 \otimes \left(p_{i,b} - p_{i,old}\right) + c_3 \otimes \left(g_b - p_{i,old}\right) \\ p_{i,new} = p_{i,old} \oplus v_{i,new} \end{cases} \tag{1}$$

where

| | |
|---|---|
| $v_{i,new}$ | New velocity calculated for the $i^{th}$ particle |
| $v_{i,old}$ | Velocity of the $i^{th}$ particle from the previous iteration |
| $p_{i,new}$ | New position calculated for the $i^{th}$ particle |
| $p_{i,old}$ | Position of the $i^{th}$ particle from the previous iteration |
| $p_{i,b}$ | Best position of the $i^{th}$ particle so far |
| $g_b$ | Best position of from the neighbour so far |
| $c_1, c_2, c_3$ | Social/cognitive confidence coefficients |

The PSO algorithm described above is known as traditional PSO and works well only in continuous domains. This is a limitation of the classical PSO because many applications are set in a space featuring discrete variables. Naturally, Discrete Particle Swarm Optimization (DPSO) is developed to tackle those discrete optimization problems. It differs from the traditional PSO in the sense that its particles do not represent points in the $n$-dimensional Euclidean space [17]. In this paper, it represents, in a discrete nature, a combination of selected position assignments for mobile robots in a formation. Equation (1) is also used to update particle's velocity and position in the DPSO. The particle position and velocity encoding varies from one specific problem to another and will be presented in Section 3 for the case of position assignment.

# 3. ROBOTIC FORMATION INITIALIZATION APPROACH

## 3.1 Premises and Problem Statement

The focus here is on the optimal position assignment and trajectory generation for robots in a desired formation, other issues such as path planning, choice of the leader robot in a formation or motion control are beyond the scope of this paper. We assume that the planner knows the desired position of a formation and the environment for initialization is obstacle-free. Each robot is assumed to move at constant speeds and be able to switch instantaneously between a fixed speed and halting, which is typical in multi-robot motion planing [11]. Therefore, the shortest path for a robot would be a straight line connecting its initial position and its desired position in the formation. The formation initialization problem for $N$ robots is now defined as follows: *Given the position of the leader robot and a desired formation configuration, assign the desired position for each robot in the group and find a set of velocity profiles for each robot* $R_i (i = 1,..., N-1)$ *to move from its initial position to its desired position without colliding with each other, while minimizing the total time for all robots to establish the formation.*

## 3.2 Formation Initialization Algorithm

We divide the problem into two sub problems. Firstly, DPSO is applied to find the optimal position assignment for each robot. Each robot is assigned a desired position in a formation so that total time required by all robots to form the formation is minimized. Once each robot has known its assigned position, the next step is to perform a geometric checking to find a priority scheme by which a robot has to wait for others in the case of a potential collision with another robot and the total waiting time for all robots will be minimized.

### 3.2.1 Position Assignment

Let us consider $N$ robots initially scattering in the workplace. One robot has been chosen to be the reference robot and is called the "leader robot" while the rest are called follower robots in this paper. The selection of the leader robot may be based on its geographical advantage in the whole group or its powerful sensing capacity. From the initial position of the selected leader robot together with the required formation configuration, desired positions of all follower robots in the formation are calculated. If there are $N$ robots, including the leader, to enter a formation in which each desired position in the formation is indexed as $P_i(x_i, y_i, \theta_i), i = 1..N-1$, or just $P_i$ for convenience, and each robot excluding the leader has a unique identification (ID) as $R_1,..., R_{N-1}$ or just $1, ..., N-1$ then each particle position and velocity are encoded as follows.

- Particle position $p_i = (x_1, x_2,..., x_{N-1})$, with $x_j \in \{1,.., N-1\}$ means that robot $R_{x_i}$ is assigned to the desired position $P_{x_i}$ in the formation.

*Objective function*

As stated above, the criteria for the position assignment problem is to minimize the total distance traversed by all robots to form the formation. To ease the computational burden of the planner as well as the implementation complexity in trajectory tracking for each robot, the trajectory of each robot when moving from an initial position to its desired position is separated into three phases. They are: (i) spinning on its wheels to align with the straight line connecting its initial position to its desired position; (ii) following that line, and (iii) spinning on its wheels to align with the formation orientation. Suppose the $i^{th}$ follower robot does not have to wait for other robots in a potential collision or there is no propensity of collision, then the time needed to reach its goal is

$$T_i = \omega_i^{-1}\alpha_{i,1} + \omega_i^{-1}\alpha_{i,2} + v_i^{-1}d_i\,, \qquad (2)$$

where $\alpha_{i,1}$, $\alpha_{i,2}$ and $d_i$ are respectively angular and translational displacement corresponding to each phase. The objective function is chosen as:

$$\min_{v_i,\ \omega_i} (F = \sum_{i=1}^{N-1} T_i) \quad (3)$$
$$s.t.\ \text{safety}$$

The actual total time executed by a follower robot to reach a desired position may be different from (2) as it may have to stop and wait for other robots if inter-vehicle collision is to happen. As each robot's angular $(\alpha_{i,1} + \alpha_{i,2})$ and translational $(d_i)$ movement are assigned by the desired formation shape, a solution found with a minimum cost defined by (3) will guarantee that the position assignment is time-optimal subject to the safety condition of inter-robot collision avoidance, which will be described in the next section.

### 3.2.2 Motion Planning

As mentioned, the cost function used in DPSO algorithm given in (3) should be subject to the collision-free condition. This constraint is resolved in this paper via a motion planner to construct a suitable trajectory for each robot where its safety will be preserved. For the safety purpose, the physical robot is located as a circle with radius $r_{safe} = r + r_{margin}$, where $r$ is the distance from the center of the robot to the its furthest point and $r_{margin}$ is the marginal clearance around it [18]. With the segmentation of a robot's trajectory corresponding to three phases as stated above, the boundary of a robot path will be rectangular which facilitates the geometrical collision checking.

To safely reach a target for a robot in the group, a quick geometrical check is performed to identify robots which potentially collide with each other. Those robots whose paths do not geometrically collide with others will be excluded from the search operation and their velocity profiles are constructed straightforwardly. For those robots whose geometrical paths cross the other paths may or may not collide with each, depending on whether or not they reach the same point at the same time, their velocity profiles are obtained from the search by adopting further the following inter-vehicle collision avoidance strategy:

- For robot $R_i$, calculate the time instant and duration it crosses the other robots' paths as

$$CT_i = [(t_0^1,\ t_1^1, index_1), (t_0^2,\ t_1^2, index_2), \cdots, (t_0^j,\ t_1^j, index_j)]$$

which means robot $R_i$ crosses and occupies the path of robot $R_{index_j}$ during $(t_0^j,\ t_1^j)$. Once a robot crosses another robot's path, it is included in the inter-vehicle collision avoidance checking even if it does not reach the crossing point at the same time with the other robot. Thus, $CT_i$ will change over time as explained later.

- The search result for robot $R_i$ is of the form $VP_i = [b_1, b_2, \cdots, b_{M_i}]$, where $b_i \in \{0, 1\}$ and $M_i$ is the number of time steps in the search outcome for robot $R_i$. This will depend on the length of the robot path, robot velocity, and the duration of each time step. Robot $R_i$ will follow its planned path with a constant velocity during the $i^{th}$ time slot if $b_i = 1$; otherwise, it stops. When $b_i = 1$, the velocity of robot $R_i$ may be $(v_i = 0, \omega_i = const)$ if it is spinning on wheels, or $(v_i = const, \omega_i = 0)$ if it is following the straight line to its desired position.

- Collision between any two robots, if occurs, will take place only one time due to the feature of a robot path. Hence, once the collision issue has been resolved, these two robots are collision-free with each other so either one will be released from the safety check with respect to the other. If a robot has no more potential collision with others, it will be released from the inter-robot collision checking procedure and it will safely follow its path.

- Deadlock may occur if robot $R_i$ crosses the target of robot $R_j$ [15]. In order to avoid this deadlock, $R_j$ must wait until $R_i$ passes its target.

- If a robot reaches the crossing area before its counter-part robot and no deadlock occurs when crossing the area first, then it will have a higher priority than its counter-part. If deadlock occurs, it has to wait for its counter-part to cross first and therefore, acting in a lower priority.

- At each time step $i^{th}$, the current time is compared with the time in $CT_i$:

  1. If $R_i$ does not reach any crossing area, set $b_i = 1$.

2. If $R_i$ is at a crossing area:

   2.1 If deadlock occurs when it moves, set $b_i = 0$.

   2.2 If no deadlock occurs and its counter- part has not reached this crossing area, set $b_i = 1$

   2.3 If no deadlock occurs and its counter-part has reached this crossing area:

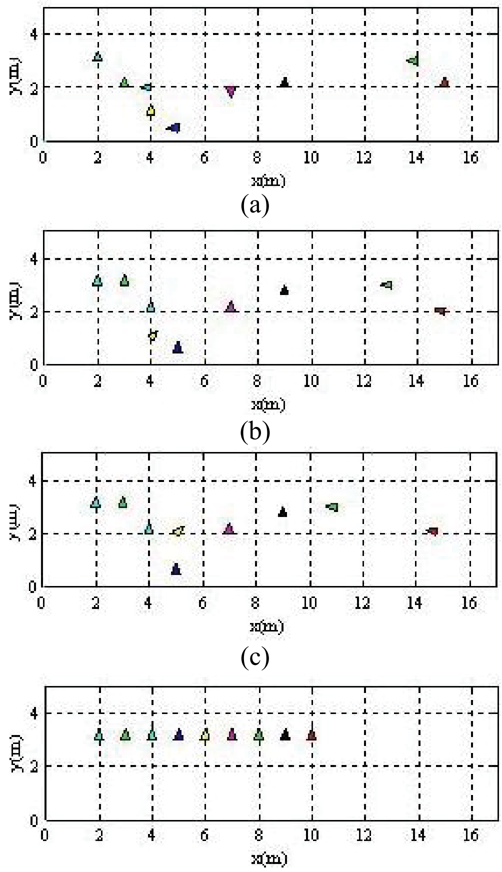
   - If it has entered this area before its counter-part then it will continue moving, i.e., set, $b_i = 1$.
   - If a counter-part has entered this area before the considered robot $R_i$, then check the collision potential between the two robots. If both of them move to the next step, then set $b_i$ accordingly, i.e., $b_i = 1$ if no collision exists and $b_i = 0$ otherwise.
   - If $R_i$ has passed the crossing area, then release $R_i$ and its counter-part from the checking procedure in future steps.

After all the search results have been found, desired velocity profiles or safe trajectories for all robots in the formation will be constructed straightforwardly.

## 4. SIMULATION RESULTS

The proposed method for robotic formation initialization using DPSO and the behavioural collision avoidance strategy is applied to simulations for different type of formation configurations with different number of robots. Due to limited space, the simulation results for a case of 9 robots to form a line formation are presented.

Figure 1a through 1d show the snapshots for robot locations over time. The robots from left to right and top to bottom are the leader robot, the initial location of follower robots $R_1, R_4, R_6, R_5, R_7, R_8$, and $R_2$, respectively. The proposed algorithm gives the best position as $P_{best} = [1, 4, 3, 6, 5, 8 ,7, 2]$.




(a)

(b)

(c)

(d)

**Figure 1. Example of Nine Robots to Form a Line Formation. Snapshots of the Formation Over Time.**

## 5. CONCLUSION

A discrete particle swarm optimization (DPSO) algorithm has been proposed for scheduling optimally the position ass ignment to initialize a formation of vehicles, which can be deployed for construction automation purposes. For the sake of simulation, the fitness function to implement DPSO comprises the total time required for all vehicles to establish the formation, subject to the collision avoidance condition. Simulation results are included to verify the approach effectiveness. Inpractice, further criteria such as sensing capacity and geometrical advantages can also be incorporated in this multi-objective optimization technique.

# 6. ACKNOWLEDGMENTS

This work is supported by the Vietnam Ministry of Education and Training and also by the ARC Centre of Excellence program, funded by the Australian Research Council (ARC) and the New South Wales State Government.


# 7. REFERENCES


[1] A. J. Healey, "Application of formation control for multi-vehicle robotic minesweeping," in Proc 40th IEEE Conference on Decision and Control, Orlando, Florida, Dec. 2001, pp. 1497-1502.

[2] B. Donald, L.Gariepy, and D.Rus, "Distributed manipulation of multiple objects using ropes," in Proc IEEE International Conference on Robotics and Automation, San Francisco, CA, 2000, pp. 450-457.

[3] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," Autonomous Robots, vol. 8, pp. 325-344, 2000.

[4] J. Feddema, C. Lewis, and D. Schoenwald, "Decentralized control of cooperative robotic vehicles: Theory and application," IEEE Transactions on Robotics and Automation, vol. 18, pp. 852-864, 2002.

[5] M. Lemay, F. Michaud, Letourneau.D, and J.-M. Valin, "Autonomous Initialization of robotic formations," in Proc IEEE Int. Conf. on Robotics and Automation, New Orleans, LA, 2004, pp. 3018-3023.

[6] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Control of changes in formation for a team of mobile robots," in Proc IEEE Int. Conference on Robotics and Automation, Detroit, MI, May 1999, pp. 1556-1561.

[7] Y. Hao and S. K. Agrawal, "Planning and control of UGV formations in a dynamic environment: A practical framework with experiments," in Proc IEEE Int. Conference on Robotics and Automation, Taipei, Taiwan, Sep. 2003, pp. 1209-1214.

[8] H. Yamaguchia and T. Arai, "Distributed and autonomous control method for generating shape of multiple mobile robot group," in Proc IEEE International Conference on Robotics and Automation, San Diego, CA, 1994, pp. 800-807.

[9] B. Erkin, S. Onur, and S. Erol, "A review: Pattern formation and adaptation in multi-robot systems," Robotics Institute - Carnegie Mellon University, Pittsburgh CMU-RI-TR-03-43 2003.

[10] F. E. Schneider and D. Wildermuth, "A potential field based approach to multi robot formation navigation," in Proc IEEE International Conference on Robotic, Intelligent Systems and Signal Processing, Changsha, China, Oct.2003, pp. 680-685.

[11] Y. Guo and L. E. Parker, "A distributed and optimal motion planning approach for multiple mobile robots," in Proc IEEE International Conference on Robotics and Automation, Washington, DC, May, 2002, pp. 2612 -2619.

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc IEEE International Conference on Neural Networks, Dec. 1995, pp. 1942-1948.

[13] S. Doctor and D. K. Venayagamoorthy, "Unmanned vehicle navigation using swarm intelligence," in Proc IEEE International Conference on Intelligent Sensing and Information Processing, 2004, pp. 249-253.

[14] Y. Qin, D. Sun, N. Li, and Y. Cen, "Path planning for mobile robot using the particle swarm optimization with mutation operator," in Proc 3rd Intl. Conf. on Machine Learning & Cybernetics, Shanghai, China, Aug. 2004, pp. 2473-2478.

[15] N. M. Kwok, Q. P. Ha, V. T. Ngo, and S.M. Hong, “Particle Swarm Optimization-based Coordination of a Group of Construction Vehicles,” in Proc. Int. Sym. Automation and Robotics in Construction, Tokyo, Japan, Oct. 2006, pp. 840-845.

[16] M. Clerc, "Binary particle swarm optimisers: toolbox, derivations, and mathematical insights," in http://clerc.maurice.free.fr/pso/, accessed Feb 2005.

[17] E. S. Correa, A. A. Freitas, and C. G. Johnson, "A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set," in Proc Genetic and Evolutionary Computation Conference, 2006, pp. 35-42.

[18] A. D. Nguyen, V. T. Ngo, N. M. Kwok, and Q. P. Ha, "Formation Initialization with Virtual-Head Robot Tracking and Three-Point l-l Control," International Transactions on Systems Science and Applications, vol. 2, no. 4, pp. 405-412, 2007.