International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012)

Multiple Robots Formation – A Multiobjective Evolution Approach

Genci Capi* and Zulkifli Mohamed

University of Toyama, Gofuku 3190, Toyama, Japan

Abstract

In this paper, we present a new method for multiple robots formation, which means certain geometrical constrains on the relative positions and orientations of the robots throughout their travel. In our method, we apply multiobjective evolutionary computation to generate the neural networks that control the robots to get to the target position relative to the leader robot. The advantage of the proposed algorithm is that in a single run of multiobjective evolution are generated multiple neural controllers. We can select neural networks that control each robot to get to the target position relative to the leader robot. In addition, the robots can switch between neural controllers, therefore creating different geometrical formations. The simulation and experimental results show that the multiobjective-based evolutionary method can be applied effectively for generating neural networks which enable the robots to perform formation tasks.

Keywords: Robot formation, neural networks, evolution.

1. Introduction

Recently, a lot of research is conducted in teaming and cooperation of multiple robots. One important research issue in systems of multiple robots is the motion planning for “formation paradigm”.

Several approaches have been proposed to address the problem of multiple robots formation. The approaches range from leader following [1], [2], virtual structures [3], [4] and virtual leaders [5], [6]. In other works, social potentials [7] and formation constrained functions [8] are used to guide robots into formations.

Neural networks have been also applied to maintaining formations. In [9] a vision-based moving in formation by four mobile robots was presented. One robot, the leader, goes first providing moving plans to the other robots who follow the leading robot. In motion control, for each robot a radial basis function network approximated by learning is used.

This article presents a novel approach for multiple robot formation based on multiobjective evolutionary algorithms (MOEAs) [10]-[12]. Unlike previous methods, in the experiments presented here, the accumulated error between the target and real position of each robot relative to the leader robot is considered as a separate objective function. The nondominated sorting genetic algorithm (NSGA II) [13] is used to generate the Pareto set of neural networks that tradeoff between the separate task performance. MOEAs have been successfully applied to evolve neural networks in which the architectural complexity and performance are co-optimized [14]. MOEAs have also been applied to design feedforward neural networks for multiple task performance [15] and time series forecasting. In addition, Barlow et al. [16] employed the multiobjective genetic programming to evolve controllers for unmanned aerial vehicles.

In this paper, the MOEA is applied for the first time to evolve neural controllers for multiple robots formation. The specific questions we ask in this study are whether MOEAs can successfully generate neural controllers for general stable multiple robots formation tasks; if the evolved neural controllers can be applied for dynamic switching between different formations. In order to answer these questions, in the experiments reported here, we consider the evolution of neural controllers for e-puck robots that have to follow the leader robot in specific relative positions.

* Corresponding author. Tel./Fax: 0081-76-445-6745
E-mail address: capi@eng.u-toyama.ac.jp

1877-7058 © 2012 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
doi:10.1016/j.proeng.2012.07.156
In order to further verify the effectiveness of the proposed method for multiple robots formation, the number of robots is increased. In the proposed method, we evolve the weight connections of the neural controller for each robot. As the number of robots in the formation increases some robots position do not match with the target ones. This is because the target positions spreads in a wide area. Therefore, the initial position of the robot during evolution influences the fitness function. This makes the evolution process difficult. The robustness of the neural controllers evolved in the simulation was also tested in the hardware experiments.

2. Multiobjective Evolutionary Algorithm

In multiobjective optimization problems there are many (possibly conflicting) objectives to be optimized, simultaneously. Therefore, there is no longer a single optimal solution but rather a whole set of possible solutions of equivalent quality. Consider without loss of generality the following multiobjective maximization problem with m decision variables, x parameters and n objectives:

\[ y = f(x) = (f_1(x_1, \ldots, x_m), \ldots, f_n(x_1, \ldots, x_m)) \]

where \( x = (x_1, \ldots, x_m) \in X \), \( y = (y_1, \ldots, y_n) \in Y \) and where x is called decision parameter vector, X parameter space, y objective vector and Y objective space. A decision vector \( a \in X \) is said to dominate a decision vector \( b \in X \) (also written as \( a \preceq b \)) if and only if:

\[ \forall i \in \{1, \ldots, n\} : f_i(a) \geq f_i(b) \land \exists j \in \{1, \ldots, n\} : f_j(a) > f_j(b) \] (2)

NSGAII was employed to evolve the neural controller where the weight connections are encoded as real numbers. In [17], the authors compared the NSGAII with four other multiobjective evolutionary algorithms using two test problems. The NSGAII performed better than the others did, showing that it can be successfully used to find multiple Pareto-optimal solutions. In NSGAII, before selection is performed, the population is ranked on the basis of domination using Pareto ranking.

3. Multiobjective Evolutionary Algorithm

3.1. Formation task

The developed system is shown in Fig. 1(a). The robots have to get to their position relative to the leader robot and keep the geometrical formation throughout their travel. Because the robots initial positions are different from the target ones, the robots have to move fast to reach to their position relative to the leader robot. The entire environment is a rectangle of surrounded by walls. The individual life time of each robot is 500 time steps, where each time step lasts 0.1s. During this time the leader robot moves with a constant velocity of 0.1m/s. The first task consists of two robots following the leader robot forming a triangle of a predetermined shape.

![Formation task](image)

Fig. 1. Formation task: (a) Developed system; (b) Formation task.
3.2. Neural Architecture

We implemented a simple feed-forward neural controller with 3, 2 and 2 units in the input, hidden and output layers, respectively. The reason that we selected a simple robot is that we are interested to conduct evolution of neural controllers in the real hardware. The inputs of the neural controller are the angle ($A_{\text{Leader}}$) and distance ($D_{\text{Leader}}$) of the E-puck robot relative to the leader robot and orientation of the E-puck robot (Fig. 1(b)). The egocentric angle to the leader robot varies from 0 to 1 where 0 corresponds to 45° to the right and 1 is 45° to the left of the leader robot. Random noise, uniformly distributed in the range of +/- 5% of sensor readings, has been added to the angle of the E-puck robot. Because the distance to the E-puck robot during the experiments is determined based on the number of pixels, the random noise in simulations is considered in the range of +/- 10%. The hidden and output units use sigmoid activation function:

$$y_i = \frac{1}{1 + e^{-x_i}}$$  \hspace{1cm} (3)

where the incoming activation for node $i$ is:

$$x_i = \sum_j w_{ij} y_j$$  \hspace{1cm} (4)

and $j$ ranges over nodes with weights into node $i$.

The output units directly control the right and left wheel angular velocities where 0 corresponds to no motion and 1 corresponds to full-speed forward rotation. The maximum forward velocity is considered to be 0.1 m/s.

![Fig. 2. Nondominated optimal solutions of different generations.](image1)

![Fig. 3. Simulation and experimental results (2 robots).](image2)
3.3. Evolution

For any evolutionary computation technique, a chromosome representation is needed to describe each individual in the population. The genome of every individual of the population encodes the weight connections of the neural controller. The connection weights range from -5 to 5. The target distance \(d_t\) between the leader and E-puck robots is considered 0.3m, while the target angle +/-15 degree. During evolution each neural of the population controls a single robot motion and at the end of its lifetime the fitness function for each target position is calculated. In order to minimize the difference between the target and real position relative to the leader robot, the fitness \(f_j\) of each robot in the formation (\(j=1\)–number of robots in the formation), is calculated as follows:

\[
 f_j = \max_{st}\sum_{i=1}^{max_{st}} \left( |d_i^r - d_i^t| + |\theta_i^r - \theta_i^t| \right)
\]  

(5)

where \(max_{st}\) is the maximum number of steps, \(d_i^r\) and \(d_i^t\) are the real and the target distance, and \(\theta_i^r\) and \(\theta_i^t\) are the real and target angle. If an individual happens to get out of the visual field, hit the leader robot or the wall, the trial is terminated and a low fitness is assigned. Therefore, such individuals will have a low probability to survive. The following genetic parameters are used: \(N_{ge}=30\), \(N_{pop}=100\), \(\sigma_{shared}=0.4\).

4. Results

All the robots have a wireless communication with the control PC. The leader robot is equipped with a wireless camera used to calculate the distance, angle and direction of the robots. We first discuss the best solutions obtained from the MOEA, for a simple formation mechanism where two robots have to be positioned relative to the leader robot. Fig. 4 shows the nondominated optimal front for generations 1, 10, and 30, averaged for five different runs of MOEA. During the first 30 generations there is a great improvement of the quality and distribution of nondominated optimal solutions.

The nondominated optimal front of generation 30 has a clear tradeoff between the two objective functions (Fig. 2). The extreme solutions represent the best Neural Networks that control each robot to get to the target position relative to the leader robot and keep this throughout the motion.

Simulation and experimental results where the E-puck robots form a triangle with the leader robot are shown in Fig. 3. Initially, the E-puck robots position is different from the target ones. Therefore, the robots move quickly to get into the desired position relative to the leader robot.

Fig. 4. Robot 1 neural controller.
The Hinton diagram of Robot 1 NN weight connections (Fig. 4(a)) show that initially, the hidden unit 1 (H1) is fully activated. Due to positive connection with the LeftMotor unit and negative connection with the RightMotor unit, the LeftMotor unit is nearly fully activated and the RightMotor unit is deactivated (Fig. 4(b)). Therefore, the Robot 1 first rotates clockwise to reach the target position relative to the leader robot. Initially, the Robot 1 is positioned in the front of the leader robot (ang=0.5) and as the robot moves the angle converge to the target one. When the robot direction is not headed toward the leader robot (Dir=-1), the Robot 1 makes a quick right turn. Then it follows the leader robot keeping the relative position.

The Robot 2, initially is not headed toward the leader robot (Dir=-1). Due to a strong negative connection between the \( H_2 \) with the \( L_{\text{wheel}} \) unit, as shown in Fig. 5(a), the left wheel essentially stops moving while the right wheel continues to rotate with nearly the maximum velocity (Fig. 5(b)). Therefore, the Robot 2 rotates counterclockwise to move to the target position. Then the Robot 2 strategy is to move slowly clockwise until the leader robot is not in front of the robot. Due to the discontinuity in activation of \( \text{Dir}_{\text{action}} \), the E-puck robot turns quickly counterclockwise (the \( L_{\text{wheel}} \) unit is fully activated).
We implemented the evolved optimal neural controllers on the real hardware of the E-puck robots. The distance and angle of the E-puck robots are calculated using the visual sensor. Because the distance is calculated based on the number of pixels, while the angle based on the blob position, there are some differences between the simulated and real robot performance. For example, the distance and angle input units have some discontinuity in their activation. However, despite these differences, the robots still performed the formation task well.

An important advantage of applying MOEAs to evolve neural controllers for multiple robot formation is the ease with which the number of robots may be increased. In the following, we present the results where another robot target position is added in front of the target robot. However, the robot positions are not the same with the target ones (Fig. 6). For example, Robot 3, after reaching the target position, continues to move forward. As the number of robots in the formation increases some robots position do not match with the target ones. This is because the target positions spreads in a wide area. Therefore, the initial position of the robot during evolution influences the fitness function. For example, even if good individual (NN) of the population can get a bad fitness if the distance between the initial and the target position is large and vice-versa. This problem can be solved by increasing the number of robots during evolution considering as fitness function the average fitness of all robots. The initial positions of multiple robots can be designed based on the target positions, reducing the effect of the initial and target positions in the fitness function.

Simultaneous evolution of multiple neural networks controlling the robots to reach the target position relative to the leader robot, makes it possible that robots switch between neural controllers resulting in different formations. Fig. 7 shows two robots switching among three evolved neural controllers. First, two robots are controlled by the neural networks that control them to get to the target 1 and target 2 position relative to the leader robot. Then, while the Robot 1 continues to be controlled by the same NN, the Robot 2 switches to another neural controller. Therefore, the Robot 2 moves to another target position relative to the leader robot.

5. Conclusions

This paper has experimentally investigated the effectiveness of applying MOEAs to address the multiple robot formation problem. In particular, it was demonstrated that in a single run of the MOEA, robust neural controllers for each robot to get to the target position relative to the leader robot and keep it throughout the motion. The robustness of evolved neural controllers was also tested on the real hardware.

For future work, we will address two possible extensions. First, we plan to increase the number of robots during evolution considering as fitness function the average fitness of all robots. The second possible extension is to develop neural controllers for formation task in more dynamic environments.

References

[1] Parker, L. E., 1998. Alliance: An architecture for fault tolerant multirobot cooperation, IEEE Transactions on Robotics and Automation, 14/2, pp. 220-40.
[2] Desai, J. P., Ostrowski, J. P., Kumar, V., 2001. Modeling and control of formations of nonholonomic mobile robots, IEEE Transactions on Robotics and Automation, 17/6, pp. 905-908.
[3] Beard, R. W., Lawton, J., Hadaegh, F. Y., 2001, A feedback architecture for formation control, IEEE Transactions on Control Systems Technology, 9/6, pp. 777-790.
[4] Lewis, M. A., Tan, K., 1997. High precision formation control of mobile robots using virtual structures, Autonomous Robots, 4, pp. 387-403.
[5] Leonard, N. E., Fiorelli, E., 2001. "Virtual leaders, artificial potentials and coordinated control of groups," in Proceedings of the IEEE Conference on Decision and Control, Orlando, Florida, pp. 2968-2973.
[6] P. Ögren, E. Fiorelli, and N. E. Leonard, 2002. “Formations with a mission: Stable coordination of vehicle group maneuvers,” Proceedings of 15th International Symposium on Mathematical Theory of Networks and Systems.
[7] Balch, T., Hybinette, M., 2000. “Social Potentials for Scalable Multi Robot Formations,” Proc. IEEE International Conference on Robotics and Automation, pp. 73–80.
[8] Egerstedt M., Hu, X., 2001. Formation constrained multi-agent control, IEEE Transactions on Robotics and Automation, 17/6, pp. 947–951.
[9] Hirota, K., Kuwabara, T., Ishida, K., Miyano, A., Ohsawa, T., Takeuchi, W., Yubazaki, N., Ohtani, M., 1995. “Robots moving in formation by using neural network and radial basis functions”, Proceedings of the 1995 International Conference on Fuzzy Systems, vol. 5, pp. 91–94.
[10] Coello, C. A., Van Veldhuizen, D. A., Lamont, G. B., 2002. Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers, New York.
[11] Deb, K., 2001. Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons, Chichester, UK.
[12] Fonseca, C.M., Fleming, P.J., 1995. An overview of evolutionary algorithms in multiobjective optimization, Evolutionary Computation, 3, pp. 1-16.
[13] Tran, K. D., 2009. An Improved Non-dominated Sorting Genetic Algorithm-II (ANSGA-II) with adaptable parameters, Intl. Jour. of Intelligent Systems Technologies and Applications, 7/4, pp. 347-369.
[14] Abbass, H. A., 2001. A Memetic Pareto Evolutionary Approach to Artificial Neural Networks, The Australian Joint Conf. on Artificial Intelligence, Adelaide, Lecture Notes in Artificial Intelligence LNAI 2256, Springer-Verlag, 1-12.
[15] Capi, G., 2007. Multiobjective Evolution of Neural Controllers and Task Complexity, IEEE Transactions on Robotics, 23/6, 1225-1234.
[16] Barlow, G. J., C. Oh, K., Grant, E., 2004. "Incremental evolution of autonomous controllers for unmanned aerial vehicles using multi-objective genetic programming. Proceedings. of IEEE Int. Conf. on Cybernetics and Intelligent Systems (CIS).
[17] Dias A. H. F., De Vasconcelos, J. A., 2002. Multiobjective genetic algorithms applied to solve optimization problems, IEEE Transactions on Magnetic, 38/2, pp. 1133-1136.