A distributed algorithm of group robots applied to maze searching

T. Onitsuka O. Araki
Institute of Pure and Applied Physics
Tokyo University of Science
Tokyo, Japan

T. Tamura
Personal Computer & Network Company
TOSHIBA Corporation
Tokyo, Japan

Abstract

In this study, we proposed a simple algorithm of group robots, which assumes to work independently works on each of the robots. The algorithm uses local distance information without specific centralized control. Usually they move around randomly and individually. Whether the distance between the robots increases beyond a constant value or decreases below another constant switches their behavior into shrink mode or expansion (random search), respectively. In the shrink mode, a robot which is farthest from the others is selected as a leader, and the others go straight toward the leader until the distance decreases below the value. Applying this algorithm to maze searching on computer simulations, we observe interesting properties as follows: (1) The parameter values which switch shrink and expansion modes affect the performance of solving the maze problem, suggesting the important relation between the values and the structural scale of the maze. (2) Proceeding and exploring behaviors with dividing and merging subgroups are self-organized. This leads to better performance an average compared with the results by a set of robots composed of a random searching robot.

1 Introduction

Recently, robotic technologies have been developed such as humanoid robots who can walk using two legs, micromachines with electromagnetic oscillation, and so on. Generally, to make them more intelligent, more complicated algorithms and control systems are needed. However, even an insect can show complicated and intelligent behaviors, which suggests that a group composed of simple and distributed agents may have those functions. In this study, we focus on the algorithm of each simple component, which shows intelligent behaviors as a system. In other words, intelligent behaviors emerge, even though each robot follows the same and simple algorithm. Thus, there is no specific leader who controls the group.

Research approaches inspired by emergent intelligent behaviors of swarms is called Swarm Intelligence (SI)[1][2][3]. SI systems are typically made up of a population of simple agents interacting locally with one another and with their environment. Although there is normally no centralized control structure dictating how individual agents should behave, local interactions between such agents often lead to the emergence of global behavior. Representative examples of SI systems are Ant Colony Optimization (ACO)[4] and Particle Swarm Optimization (PSO)[5][6]. In a swarm of insects or a school of fish, when one finds a desirable path, the rest of the members will follow it in PSO. Assuming an evaluation function over the search space, the vector of velocity of each agent is repeatedly modified depending on the agent's position with global minimum and on the local information. As this calculation proceeds, every agent tends to move toward the position where the objective function has an optimized value. While every agent follows a simple and the same algorithm, the group quickly reaches to optimized position as a group. Our study assumes the situation in which the objective function is not clear. We focus on clustered and searching behaviors such as swarm intelligence without using the objective function over the field.

The purpose of this study is to propose a simple algorithm in each agent distributed when the objective function cannot be used. Each agent can utilize limited local information and communicate each other. The key of the algorithm is the alternative modes of expansion and shrink. Each robot always measures the distance from each other, and notices a leader of the group when the distance increase beyond a constant. Every robot except the leader follows the leader until the distance decreases below another constant value (shrink mode). Then, each robot goes random direction again (expansion mode). While the simulation robots are searching in the maze, we can observe not only the performance of how soon they reach the goal but also emergent behaviors of robustness and adaptability of the system. Our algorithm does not assume centralized control structure. Thus, the system will not stop, even if some of them are lost or broken. And our algorithm is so simple that we can implement it on the hardware robots easily. The most important feature of our algorithm is random searching and a clustered behavior like swarms or a school of fish at the same time. In this study, we did computer simulations in a maze using software robots to show the properties of the algorithm. The parameter values which decide mode switching are shown to be crucial to the performance of the maze searching. Depending on the values, the system was able to reach a goal earlier than random searching.

2 Algorithm

As a first step of our study, we constructed an agent model such as a swarm as simple as possible, whose algorithm is shown as follows(Fig.1): First, each robot moves around randomly without a leader ("expantion mode"(Fig.1-(1)). Each one assumes to go straight in one direction until it reaches a wall, then it reflects to another direction. The reflection angle is randomly selected in order to avoid infinite bouncing in a dead end. When a robot whose distance from the other robots exceeded a constant value of a parameter (d1), the robot becomes a leader (Fig.1-(2)). At this time, at least two robots or more become leader's candidates. The position where the previous "expansion" mode starts in each robot is memorized, and the distances between the position and the current position are compared. Then the robot whose value is largest is selected as a leader. If the distance is also the same, it is randomly selected. This is because we think one which was advanced larger from the previous expansion should be a leader. Once the leader decided, the other robots go straight toward the leader ("shrink mode"(Fig.1-(3))). Only the leader keeps going before as it does, while the other robots changes each direction toward the position of the leader and keep following it at an equal speed until the longest distance between any two robots decreases below another constant parameter (d2). Then, the leader is dismissed and the group's movement returns back to the first free searching (Fig.1-(4)). The velocity of all motions is assumed to be constant except when the direction is changed, when the mode has changed, and while one is following a leader. These procedures are repeated and the leader is selected every time the distance exceeds d1. This algorithm expects that some robots far from the rest may find a new way, which will promote exploration as a group.

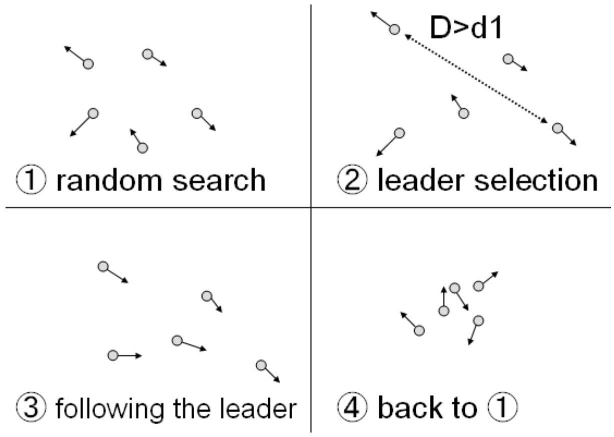

Figure 1: Schematic behaviors of the algorithm

Although an obstacle or walls in the maze may also split the group into some subgroups during searching, there is no problem because the leader is selected in each subgroup. As a result, two or more groups might coexist and continue searching, which will lead to an efficient search. When the leader is decided, the robots those follow the leader are limited in the range that the distance from the leader is less than the value of the third parameter (d3). This limitation is needed so that the effect of the robot whose position is too far away on the selection of a next leader becomes weaker. This may partially avoid back-and-forth motion of each group.

3 Computer simulations

3.1 Method

To examine properties of the above-mentioned algorithm, we did computer simulations using software robots in a simple maze. We observed behaviors of robots (N=20) in the maze (Fig.2), which start at the position in the corner and search around until they find the goal in another corner. In addition, we measured the time (number of procedural steps) spent by all of the robots from the start to the goal. The result is averaged over 100 trials with different random seeds for each combination of parameter values. When a robot reaches the goal, it assumes to stay there without becoming a leader.

The size and the width of walls of the maze is shown as follows:map size = 500x500 dot, and wall width =

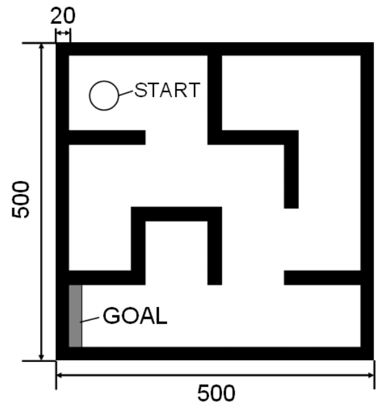

Figure 2: map

20 dot. Parameter values are as follows unless otherwise indicated:d1 = 60 - 140 (10 interval), d2 = 50, d3 = d1+25, Number of robots = 20, and Speed of robots = 6 dot/step.

3.2 Time to the goal

Figure 3 shows the average number of steps until all robots reach the goal as the value of d1 changes from 60 to 140 with fixed value of d2 and d3. The result suggests that there exists an optimal value around 110 for d1 (When d1=110, the mean value indicates 4299 steps). When d2 changes with fixed d1 and d3, the average steps are shown in Fig. 4 (d1=100,d3=150). The effect of d2 on the attainment is not so clear as d1.

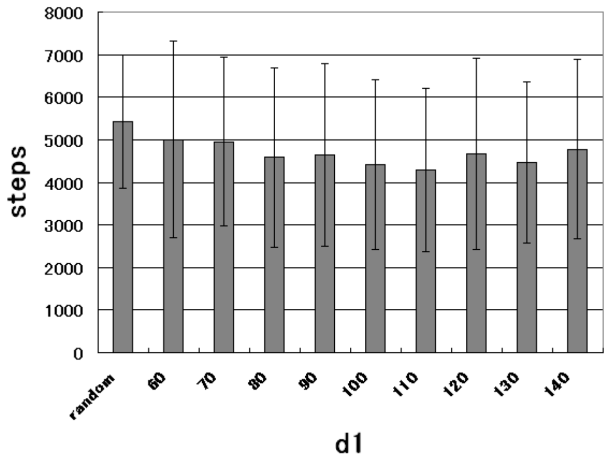

Figure 3: Average steps to the goal (d2 = 50, d3 = d1 + 25)

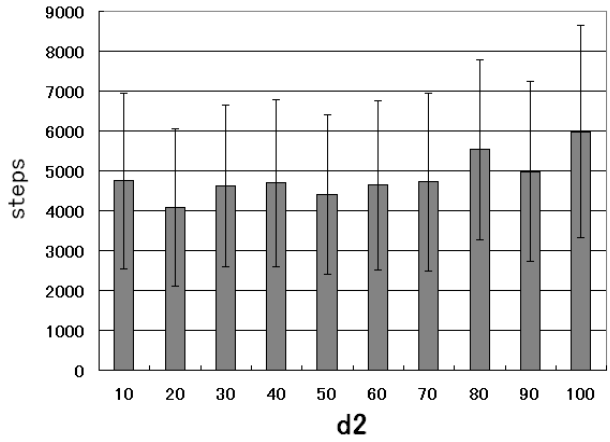

Figure 4: Average steps to the goal (d1 = 100, d3 = 150)

For comparison, we tried another set of simulations using a swarm of robots, which do not follow our algorithm but follow just individual random searching. They do not expand or shrink together. Each robot goes straight, reflects at the wall to a random direction, and stops when it reaches the goal. In this case, it takes 5425 steps on the average for all of them to reach the goal. Thus our algorithm shows better performance an average.

These results show that the value of the parameters, especially d1, affects the performance of solving the maze problem, which suggests the relationship between the values and the scale of the maze. When d1 is smaller than the width of a dead end, it will be difficult to escape from there. To escape there and explore other fields, d1 should be as large as the size. When d1 is much larger than the size of the maze, a leader is not be selected and emergent behaviors are random like a Brownian motion. Thus, matching of the parameter values are needed for the scales of the maze (width of the roads, for example) to improve the performance. Adaptation or dynamical fitting of the parameter values are one of our future problems.

3.3 Decentralized search

We observed interesting behaviors during the maze simulations. Two of them are shown in these subsections.

One mass of robots at first divides a few times at the walls as the time proceeds. Then each group explores here and there independently(Fig 5). This is one of the most interesting features of SI, decentralized search.

Acquired information during searching about position or evaluation, which is not communicated in current version, would be useful to make the behavior more intelligent.

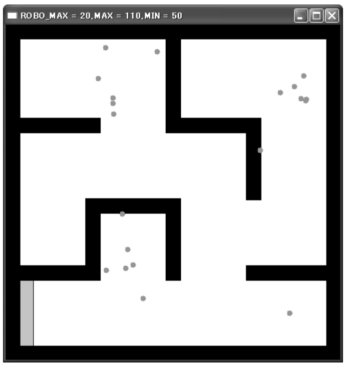

Figure 5: Decentralized search

3.4 Self-organized adaptation

Self-organized adaptational behaviors to circumstances are also observed. On the straight way they pass through there together, each explores independently in a closed way, and they return backward together. This is due to the shrink and expansion mechanism, which regularly corresponds to the searching behavior in the maze.

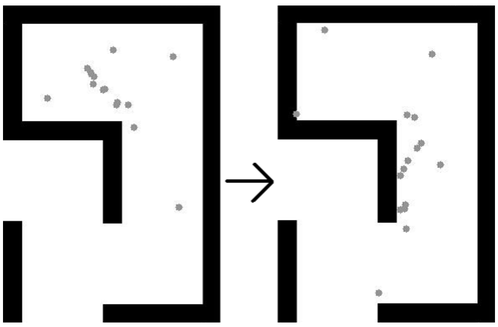

Figure 6: Escape from blind alley

Decision policy of choosing a remote leader promotes getting out from closed fields. Another important parameter d2 tends to limit the closest distance between the robots. While following the leader, each distance rarely decreases less than d2 because the speed of the leader and the others are the same. When the leader encounters a wall, which may be in a deadlock, the robot will be dismissed from the leader with the distance condition satisfied.

The behaviors such as proceeding and exploring in the maze emerge in spite of the simple algorithm using only local information i.e. distance from each other. This behavior can be regarded as emergent SI. Application to another problem, making the algorithm more general and distributed, and hardware implementation are also our future works.

4 Conclusion

In this study, we proposed a simple algorithm of group robots, which independently works on each of the robots. The algorithm uses local distance information without specific centralized control. Applying this algorithm to maze searching on computer simulations, we observe interesting properties as follows:
(1) The values of the parameters, especially d1, affect the performance of solving the maze problem, suggesting the imporatn relation between the values and the structural scale of the maze.
(2) Proceeding and exploring behaviors with deviding and merging subgroups are self-organized. This may lead to better performance compared with the results by a set of robots composed of a random searching robot.

References

[1] E. Bonabeau & M. Dorigo & G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford Univ Press, 1999.

[2] J. F. Kennedy & R. Eberhart & Y. Shi, *Swarm Intelligence*, Morgan Kaufmann Pub., 2001.

[3] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, 2006.

[4] M. Dorigo & T. Stutzle, *Ant Colony Optimization*, MIT Press, 2004.

[5] J. Kennedy & R. Eberhart, "Particle swarm optimization," *Proc. of the IEEE Int. Conf. on Neural Networks*, Piscataway, NJ, pp. 1942-1948, 1995.

[6] M. Clerc & J. Kennedy, "The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space," *IEEE Transactions on Evolutionary Computation*,Vol. 6, pp. 58-73, 2002.