A Bio-inspired Multi-Robot Coordination Approach

Yan Meng, Ọlọrundamilọla Kazeem and Jing Gan

Department of Electrical and Computer Engineering
Stevens Institute of Technology, Hoboken, NJ 07030

Abstract. In this paper, a bio-inspired stigmergy-based coordination approach is proposed for a distributed multi-robot system. This approach is inspired from the behavior of social insect swarming, where social insect colonies are able to build sophisticated structures and regulate the activities of millions of individuals by endowing each individual with simple rules based on local perception. A virtual pheromone mechanism is proposed as the message passing coordination scheme among the robots. The proposed algorithm has been implemented on embodied robot simulator Player/Stage, and the simulation results show the feasibility, robustness, and scalability of the methods under different dynamic environments with real-world constraints.

1 Introduction

The main challenges for swarm robots are to create intelligent agents that adapt their behaviors based on interaction with the environment and other robots, to become more proficient in their tasks over time, and to adapt to new situations as they occur. Typical problem domains for the study of swarm-based robotic systems include foraging [1], box-pushing [2], aggregation and segregation [3], formation forming [4], cooperative mapping [5], soccer tournaments [6], site preparation [7], sorting [8], and collective construction [9]. All of these systems consist of multiple robots or embodied simulated agents acting autonomously based on their own individual decisions. However, not all of these control architectures are scalable to a large number of robots. For instance, most approaches rely on extensive global communication for cooperation of swarm robots, which may yield stressing communication bottlenecks. Furthermore, the global communication requires high-power onboard transceivers in a large scale environment. However, most swarm robots are only equipped very limited sensing and communication capability.

An alternative paradigm to tackle the scalability issue for swarm robots while maintaining robustness and individual simplicity is through Swarm Intelligence (SI), which is an innovative computational and behavioral metaphor for solving distributed problems by taking its inspiration from the behavior of social insects swarming, flocking, herding, and shoaling phenomena in vertebrates, where social insect colonies are able to build sophisticated structures and regulate the activities of millions of individuals by endowing each individual with simple rules based on local perception.

The abilities of such natural systems appear to transcend the abilities of the con-

Meng Y., Kazeem . and Gan J. (2007).
A Bio-inspired Multi-Robot Coordination Approach.
In *Proceedings of the 3rd International Workshop on Multi-Agent Robotic Systems*, pages 14-23

stituent individual agents. In most biological cases studies so far, robust and coordinated group behavior has been found to be mediated by nothing more than a small set of simple local interactions between individuals, and between individuals and the environment.

Reynold [10] built a computer simulation to model the motion of a flock of birds, called *boids*. He believes the motion of the boids, as a whole, is the result of the actions of each individual member that follow some simple rules. Ward et al. [11] evolved *e-boids*, groups of artificial fish capable of displaying schooling behavior. Spector et al. [12] used a genetic programming to evolve group behaviors for flying agents in a simulated environment. The above mentioned works suggest that artificial evolution can be successfully applied to synthesize effective collective behaviors. And the swarm-bot [13] developed a new robotic system consisting of a swarm of s-bots, mobile robots with the ability to connect to and to disconnect from each other depends on different environments and applications, which is based on behaviors of ant systems. Another swarm intelligence based algorithm, Particle Swarm Optimization (PSO), was proposed by Kennedy and Eberhart [14]. The PSO is a biologically-inspired algorithm motivated by a social analogy, such as flocking, herding, and schooling behavior in animal populations.

Payton et al. [15] proposed pheromone robotics, which was modeled after the chemical insects, such as ants, use to communicate. Instead of spreading a chemical landmark in the environment, they used a virtual pheromone to spread information and create gradients in the information space. By using these virtual pheromones, the robots can send and receive directional communications to each other.

In this paper, we propose a bio-inspired coordination paradigm to achieve an optimal group behavior for multi-agent systems. Each agent adjusts its movement behavior based on a target utility function, which is defined as the fitness value of moving to different areas using the onboard sensing inputs and shared information through local communication. Similar to [15], inspired by the pheromone drip trail of biological ants, a unique virtual agent-to-agent and agent-to-environment interaction mechanism, i.e. *virtual pheromones,* was proposed as the message passing coordination scheme for the swarm robots. Instead of using infrared signals for transceivers in [15], which requires line of sight to transmit and receive, we use wireless ad hoc network to transmit information and the virtual pheromone structure is designed to be more robust and efficient.

This new meta-heuristic draws on the strengths of two popular SI-based algorithms: Ant Colony Optimization (ACO)'s autocatalytic mechanism and Particle Swarm Optimization (PSO)'s cognitive capabilities through interplay. Basically, two coordination processes among the agents are established in the proposed architecture. One is a modified stigmergy-based ACO algorithm using the distributed virtual pheromones to guide the agents' movements, where each agent has its own virtual pheromone matrix, which can be created, enhanced, evaporated over time, and propagated to its neighboring agents. The other one is interaction-based algorithm, which aims to achieve an optimal global behavior through the interactions among the agents using the PSO-based algorithm.

In our previous work [16], this hybrid algorithm was implemented in a proof-of-concept simulator, where each agent has been simplified to one dot without any sensors installed. The target detection was based on the distance between the agent and the target. Once the distance is within the detection range, it is assumed that the target is detected. To apply this algorithm to the real-world robotic systems, more realistic issues need to be solved. In this paper, we will modify the utility function of the previous proposed hybrid ACO/PSO algorithm to adapt to a more realistic robotic simulation environment. Player/Stage is utilized as our embodied robot simulator. Each robot in Player/Stage is installed with a camera system, a laser range finder, sonar sensor, and wireless communication sensor. The strength of this ACO/PSO coordination architecture lies in the fact that it is truly distributed, self-organized, self-adaptive, and inherently scalable since global control or communication is not required. Each agent makes decisions only based on its local view, and is designed to be simple and sometimes interchangeable, and may be dynamically added or removed without explicit reorganization, making the collective system highly flexible and fault tolerant.

The paper is organized as follows: Section II describes the problem statement. Section III presents the proposed stigmergy-based architecture for distributed swarm robots. Section IV presents the simulation environment and simulation results. To conclude the paper, section V outlines the research conclusion and the future work.

2 Problem Statement

The objective of this study is to design a bio-inspired coordination algorithm for distributed multi-robot systems. To evaluate this coordination algorithm, a multi-target searching task in an unknown dynamic environment is implemented in Player/Stage simulator. The targets can be defined as some predefined tasks need to be processed by the agents in real-world applications, for example, collective construction, resource/garbage detection and collection, people search and rescue, etc.. The goal is to find and process all of the targets as soon as possible. Assume that the agents are simple, and homogeneous, and can be dynamically added or removed without explicit reorganization. Each agent can only communicate with its neighbors. Two agents are defined as neighbors if the distance between them is less than a pre-specified communication range. The agent can only detect the targets within its local sensing range.

3 A Stigmergy-Based Coordination Approach

3.1 Virtual Pheromone as Inter-Agent Communication Mechanism

The ACO algorithm, proposed by Dorigo et al. [13], is essentially a system that simulates the natural behavior of ants, including mechanisms of cooperation and adaptation. The involved agents are steered toward local and global optimization through a mechanism of feedback of simulated pheromones and pheromone intensity processing. It is based on the following ideas. First, each path followed by an ant is associated with a candidate solution for a given problem. Second, when an ant follows a path, the amount of pheromone deposit on that path is proportional to the quality of the corresponding candidate solution for the target problem. Third, when an ant has to choose between two or more paths, the path(s) with a larger amount of pheromone are more attractive to the ant. After some iterations, eventually, the ants will converge to a short path, which is expected to be the optimum or a near-optimum solution for the target problem.

In the classical ACO algorithm, the autocatalytic mechanism, i.e. pheromone dropped by agents, is designed as an environmental marker external to agents, which is an indirect agent-to-agent interaction design in nature. In the real world applications using swarm agents, a special pheromone and pheromone detectors need to be designed, and sometimes such physical pheromone is unreliable and easily to be modified under some hazardous environments, such as urban search and rescue. A redefinition of this auto catalyst is necessary.

A *Virtual Pheromone* mechanism is proposed here as a message passing coordination scheme between the agents and the environment and amongst the agents. An agent will build a virtual pheromone data structure whenever it detects a target, and then broadcast this target information to its neighbors through a visual pheromone package. Let $p_{a^k}(t) = \{p_{ij}^k(t)\}$ represents a set of pheromones received by agent *k* at time *t*, where (*i, j*) denotes the 2D global coordinate of the detected target. Each p_{ij}^k has a cache of associated attributes updated per computational iteration. The data structure for the virtual pheromone is defined as follows:

```
  Pheromone structure
  {  Target position;
     Number of target detected;
     The ID of source robot who detects the targets;
     The robot IDs that pheromone has been propagated before
passed to this robot;
     Agent intensity;
     Pheromone interaction intensity;
     Time stamp;}
```

3.2 Target Utility

Basically, each target is associated with different pheromone. Each agent makes its own movement decision based on the parameters of a list of pheromone matrix. Here, let's define target utility and target visibility to explain the decision making procedure of each agent.

First, let $\mu_k(t) = \{\mu_{ij}^k(t)\}$ represents a set of target utilities at time *t*, where $\mu_{ij}^k(t)$ denotes the *target utility* of agent *k* , which is defined as follows:

$$\mu_{ij}^k(t) = (k_1\omega_{ij}^k(t) - k_2\tau_{ij}^k(t))/R \tag{1}$$

where $\omega_{ij}^k(t)$ and $\tau_{ij}^k(t)$ represent target weight and agent intensity, respectively. Let the target weight measures potential target resources available for agent *k* at time *t*. The agent intensity is an indication of the number of agents who will potentially process the corresponding target at location (*i,j*). When we say "potentially", we mean all of the agents who have received the same pheromone information may end up moving to the same target. However, they may also go to other targets with stronger pheromone intensity based on their local decisions.

We can use target intensity to emulate the pheromone enhancement and elimination procedure in natural world, which can be updated by the following equation:

$$\tau_{ij}^{k}(t+1) = \rho * (\tau_{ij}^{k}(t) + T_{ij}^{k}) - (1-\rho) * e * \tau_{ij}^{k}(t) \tag{2}$$

where *0<ρ<1* is the enhancement factor of pheromone intensity. T_{ij}^{k} is the pheromone interaction intensity received from the neighboring agents for a target at (*i,j*), which is defined as

$$T_{ij} = \begin{cases} \alpha, \text{ if source pheromone} \\ \beta, \text{ otherwise} \end{cases} \tag{3}$$

where *0 ≤ β < α ≤ 1*. If an agent discovers a target by itself instead of receiving the information from its neighbors, it is defined as the *source agent*. The source agent then propagates the *source pheromone*, to its neighbors. A propagation agent is a non-source agent, and simply propagates pheromones it received to its neighbors. Basically, T_{ij}^{k} is used for pheromone enhancement. *e* represents the elimination factor. In the ants system, the pheromone will be eliminated over time if it is not being enhanced by the ants, and the elimination procedure usually is slower than the enhancement. When the pheromone trail is totally eliminated, it means that no resource is available through this pheromone trail. To slow down the elimination relative to enhancement, we set $e < 1$.

R denotes local target redundancy, which is defined as the number of the local neighbors who have sent the pheromones referring to the same target at (*i, j*) to agent *k*. k_1 and k_2 are constant factors which are used to adjust the weights of target weight and agent intensity parameters.

Generally speaking, the higher the target utility is, the more attractive the corresponding target is to the agent. More specifically, when the target weight is greater than the agent intensity, it means that there are more tasks need to be processed (or there are more resources left) in this target. Therefore, the benefit of moving to this target would be higher in terms of the global optimization. If the agent intensity is greater than the target weight, it means that there will be more potential agents (globally) moving to this target, which may lead to the less available tasks (or resources) left in the future. Therefore, the benefit of moving to this target would be less in terms of the global optimization. With the local redundancy, we are trying to prevent the scenarios that all of the agents within a local neighbor move to the same target instead of exploring new targets elsewhere.

3.3 Target Visibility

Initially, the agents are randomly distributed in the searching environment, where multiple targets with different sizes and some static obstacles are randomly dispersed within the environment. At each iteration, if each agent adjusts its behavior based only on the target utility, it may lead the agent to be very greedy in terms of the agents' behaviors, since the agents would rather move to the target with higher utility than explore new areas. This greedy behavior of the agents may easily lead to local optima.

To prevent the local optima scenarios in utility-based approach mainly based on ACO, we have to take into consideration of target visibility. Let $\eta_k(t) = \{\eta_{ij}^k(t)\}$ represents a set of visibilities at time *t*, where $\eta_{ij}^k(t)$ denotes the target visibility for agent *k* in terms of target at location (*i, j*), which is defined by the following equation:

$$\eta_{ij}^k(t) = r^k / d_{ij}^k(t) \tag{4}$$

where r^k represents the local detection range of agent *k*, and the $d_{ij}^k(t)$ represents the distance between the agent *k* and the target at location (*i, j*). If $\eta_{ij}^k > 1$, we set $\eta_{ij}^k = 1$. When the target visibility is higher, it means the distance between the target and the agent is smaller, it would be more benefit to move to this target due to its less cost compared to moving to the far-away target under the same environmental condition.

3.4 Agent Behavior Control

Now the question is how to integrate the target utility and target visibility into an efficient fitness function to guide the movement behaviors of each agent. To tackle this issue, we turned our attention to another collective intelligence - Particle Swarm Optimization (PSO). The PSO algorithm is population-based: a set of potential solutions evolves to approach a convenient solution (or set of solutions) for a problem. The social metaphor that led to this algorithm can be summarized as follows: the individuals that are part of a society hold an opinion that is part of a "belief space" (the search space) shared by every possible individual. Individuals may modify this "opinion state" based on three factors: (1) The knowledge of the environment (explorative factor); (2) The individual's previous history of states (cognitive factor); (3) The previous history of states of the individual's neighborhood (social factor).

A direct PSO adoption to swarm agents would be difficult, because swarm agents may be blinded over in reference to global concerns without any feedback. However, the PSO algorithm is a decision processor for annealing premature convergence of particles in swarm situations. Thus, a new optimization technique specifically tailored to the application of swarm agents is proposed in this paper. This new meta-heuristics draws on the strengths of both systems: ACO's autocatalytic mechanism through environment and PSO's cognitive capabilities through interplay among agents. In this hybrid method, the agents make their movement decisions not only based on the target utility defined in (3), but also on their movement inertia and their own past experiences, which would provide more opportunities to explore new areas.

Basically, the PSO algorithm can be represented as in (5), which is derived from the classical PSO algorithm [14] with minor redefinitions of formula variables as follows:

$$v_{ij} = \textit{explorative} + \textit{cognitive} + \textit{social} \tag{5}$$

where v_{ij} is the velocity of a agent. To determine which behavior is adopted by agent *k* of the swarm, the velocity, $v_{ij}^k(t)$ has to be decided first. If the received pheromone intensity is high, the agent would increase the weight of social factor, and decrease the weight of cognitive factor. On the other hand, if the local visibility is of significant to the agent, then the velocity of the agent would prefer the cognitive factor to the social factor. Furthermore, at any given time, the velocity of the agent would leave some spaces for the exploration of new areas no matter what. Therefore, the basic idea is to propel towards a probabilistic median, where explorative factor, cognitive factor (local agent respective views), and social factor (global swarm wide views) are considered simultaneously and try to merge these three factors into consistent behaviors for each agent. The exploration factor can be easily emulated by random movement.

The challenge part is how to define local best (cognitive factor) and global best (social factor). One straight forward method is to select the highest target visibility from a list of available targets as the local best. If only one target is on the list, then this target would be the local best. The easy way to select global best is to select the highest target utility from a list of available targets. If only one target is on the list, then this target would be the global best.

Instead of defining a fitness function, for a robot system, the robot velocity vector including both magnitude and direction would be a better representation to control the movement behavior. Based on the above discussion and PSO algorithm, each agent would control its movement behaviors by following this equation:

$$\begin{aligned} v_{ij}^{k}(t+1) &= \psi_e * rand_e(\,) * v_{ij}^{k}(t) + \psi_c * rand_c(\,) * (p_c - x_{ij}^{k}(t)) \\ &+ \psi_s * rand_s(\,) * (p_s - x_{ij}^{k}(t)) \end{aligned} \tag{6}$$

where, $\psi_e, \psi_c, and \psi_s$ represent the propensity constraint factors for explosive, cognitive, and social behaviors, respectively, $0 \le rand_{\Theta}() < 1$ where $\Theta = e, c,$ or $s,$ and $x_{ij}^{k}(t)$ represents the position of agent *k* at time *t*. $p_s = \max(\mu_{ij}^{k}(t))$ represents the global best from the neighbors, and $p_c = \max(\eta_{ij}^{k}(t))$ represents the local cognitive best. The position of each agent *k* at time *t+1* can be updated by

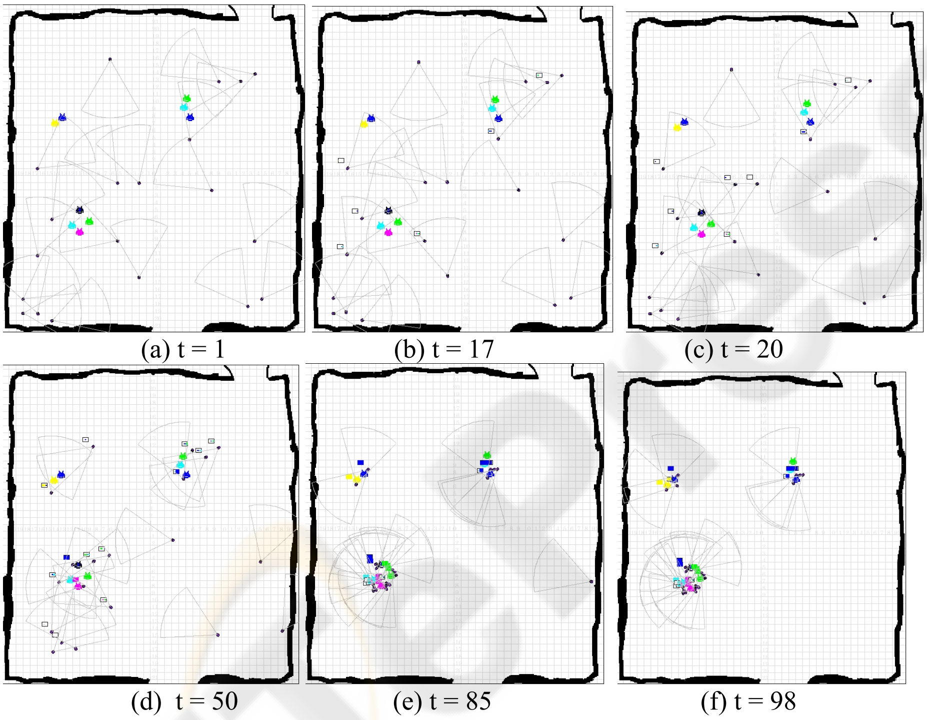
$$x_{ij}^{k}(t+1) = x_{ij}^{k}(t) + v_{ij}^{k}(t+1)\,. \tag{7}$$

4 Simulation Results

To evaluate the performance of the proposed stigmergy-based algorithm in a distributed swarm agent system, we implement this algorithm on a Player/Stage robot simulator. As shown in Fig. 1, the environment is an open space with 20 homogeneous mobile robots. Each robot is equipped with a camera system to detect and track targets, a laser range finder to measure the distance between the target and itself, a sonar sensor to avoid obstacles (i.e. both static obstacles and mobile obstacles, such as other agents), and a wireless communication card to communicate with other agents.

As shown in Fig.1, the searching environment is a rectangle area, where several targets with different colors and sizes are randomly distributed in the environment, and grey dots represent the robots. The arc shape in front of each robot represents the field of view of the vision system on each robot. The communication range is set up as the same range of the vision but using a circle instead of an arc. Whenever the robots are within other agent's communication range, they would exchange the information between them.

Initially, the agents are randomly searching for targets, as shown in Fig. 1(a) at t =1. Once a robot detects a target, it would propagate the pheromone of this target to its neighbors, as shown in Fig. 1(b), where a small rectangle indicates that the vision system on the associated robot has detected the targets. After receiving a pheromone message, robots make their own decisions where to move on next time step based on the proposed algorithm, as shown in Fig. 1(c) and Fig. 1(d). Sometimes, a robot may get trapped in a corner or boundary line trying to avoid obstacles. It may take long time before it can get out, as shown in Fig. 1(e). The simulation stops when all of the targets being found and processed, as shown in Fig. 1(f).



(a) t = 1 (b) t = 17 (c) t = 20

(d) t = 50 (e) t = 85 (f) t = 98

Fig.1. 20 robots searching for randomly distributed targets in an open space on a player/stage simulator at t = 1, 17. 20, 50, 65, and 82 time steps.

Generally, global path planning is very time consuming, especially for swarm robots where each agent may have to replan its global path very frequently due to the constant agent-to-agent collision. Dynamic mobile agent avoidance is another challenging task, which is not our focus in this paper. Therefore, to speed up the searching procedure in simulation, a simple path planning method is conducted. Once an agent makes its decision according to the proposed algorithms, it will set the selected target as its destination point, and move toward the target. Since there may have static obstacles and mobile obstacles (i.e. other agents) on its way to the destination, an obstacle avoidance algorithm is necessary. Here, an adaptive vector algorithm is applied for obstacle avoidance.

To evaluate the performance of the proposed method, utility-based method is also implemented, where each agent makes its own movement decision only based on the target utility defined in (1). Three cases of different target distributions are conducted, where in each case 10 targets are distributed in the environment with fixed positions. Then, we start running the simulations with the swarm size of 20 using both methods, each method runs 35 times to obtain the mean and variance values in time steps. One time step represents the time that all of the agents need to make their movement decisions once sequentially. Some statistics results of comparison of these two methods are shown in Table 1.

Table 1. Simulation comparison of two methods.

	Mean/Variance (case 1)	Mean/variance (case 2)	Mean/Variance (case 3)
Utility-based	174/30	121/21	183/16
Hybrid	102/17	94/10	98/12

It is observed from Table 1 that the hybrid method outperforms utility-based method. The reason behind this observation is because the agents using utility-based method are extremely greedy and would always try to achieve the best utility. Therefore, they would rather move to detected targets with highest utilities than exploring areas for new targets. On the other hand, the hybrid method not only considers the target utility, but also consider the exploration (i.e. inertia factor), and its own past experiences. This exploration tendency would lead the agents using the hybrid method to be more dispersed for different targets, which may result in efficient searching results. When the agent receives the pheromone information of multiple targets, it would make decision whether to pick the target or explore to a new area, or if multiple targets are available, which one to pick so that the global optimization performance can be achieved. Furthermore, the hybrid method is more stable and consistent than the utility-based method from the variance values in Table 1. It is also observed that the proposed method is very robust to different target distributions in the environment.

5 Conclusion and Future Work

A bio-inspired stigmergy-based algorithm is proposed for a distributed multi-agent system. By using natural metaphors, inherent parallelism, stochastic nature, adaptivity, and positive feedback, the proposed method is truly distributed, self-organized, self-adaptive, and inherently scalable since there is no global control or communication, and be able to address the complex problems under dynamic environments.

However, there are still some unsolved issues remained. For example, the communication overhead among agents is extensive, which will consume too much power of limited on-board battery, especially for a large scale swarm agent system. Furthermore, it is difficult to predict the swarm performance according to a particular metric or analyze further possible optimization margins and intrinsic limitations of these approaches from an engineering point of view. Our future work will tackle these issues and mainly focus on developing a dynamic swarm model to allow the swarm agents to achieve the target global goal and expected performance.

Reference

1. M.J.B. Krieger, J. B. Billeter, and L. Keller, "Ant-like Task Allocation and Recruitment in Cooperative Robots," *Nature*, Vol. 406, 2000, pp. 992-995.
2. M. J. Mataric, M. Nilsson, and K. T. Simsarian, "Cooperative Multi-robot Box-Pushing", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1995.
3. A. Martinoli, A. J. Ijspeert, and F. Mondada, "Understanding Collective Aggregation Mechanisms: From Probabilistic Modeling to Experiments with Real Robots", *Robotics and Autonomous Systems*, Vol. 29, pp. 51-63, 1999.
4. T. Balch and R. C. Arkin, "Behavior-based Formation Control for Multi-robot Teams", IEEE Trans. on Robotics and Automation, 1999.
5. B. Yamauchi, "Decentralized Coordination for Multi-robot Exploration", *Robotics and Autonomous Systems*, Vol. 29, No. 1, pp. 111-118, 1999.
6. T. Weigel, J. S. Gutmann, m. Dietl, A. Kleiner, and B. Nebel, "CS Freiburg: Coordinating Robots for Successful Soccer Playing", Special Issue on Advances in Multi-Robot Systems, T. Arai, E. Pagello, and L. E. Parker, Editors, *IEEE Trans. on Robotics and Automation*, Vol. 18, No.5, pp. 685-699, 2002.
7. C.A.C. Parker and H. Zhang, "Collective Robotic Site Preparation". *Adaptive Behavior*. Vol.14, No. 1, 2006, pp. 5-19.
8. O.E. Holland and C. Melhuish, "Stigmergy, Self-Organization, and Sorting in Collective Robotics", *Artificial Life*, Vol. 5, pp. 173-202, 1999.
9. R. L. Stewart and R. A. Russell. "A Distributed Feedback Mechanism to Regulate Wall Construction by a Robotic Swarm". *Adaptive Behavior*. 14(1):21-51, 2006.
10. C. W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model", *Computer Graphics*, 21(4), July 1987, pp. 25-34.
11. C. R. Ward, F. Gobet, and G. Kendall. "Evolving collective behavior in an artificial ecology". *Artificial Life*, Vol. 7, No. 2, 2001, pp.191-209.
12. L. Spector, J. Klein, C. Perry, and M.D. Feinstein. "Emergence of collective behavior in evolving populations of flying agents". In *E. Cantu-Paz et. al. editor. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003)*, Berlin, Germany, 2003, pp.61-73.
13. M. Dorigo, V. Maniezzo, and A.Colorni, "Ant System: Optimization by a Colony of Cooperating Agents," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 26, No. 1, February 1996.
14. J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *IEEE Conference on Neural Networks*, Proceedings 1995.
15. D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee, "Pheromone Robotics", *Autonomous Robots*, Vol. 11, No. 3, November, 2001.
16. Y. Meng, O. Kazeem, and J. Muller, "A Hybrid ACO/PSO Control Algorithm for Distributed Swarm Robots", *2007 IEEE Swarm Intelligence Symposium*, April 2007, Hawaii, USA.