A Swarm Intelligence Based Coordination Algorithm for Distributed Multi-Agent Systems

Yan Meng and Ọlọrundamilọla Kazeem
Department of Electrical and Computer Engineering
Stevens Institute of Technology, New Jersey, USA
Email:yan.meng@stevens.edu, okazeem@stevens.edu
Phone: +1 201 216 5496

Juan C. Muller
Department of Computer Science and Mathematics
New Jersey City University, New Jersey, USA
jmuller@njcu.edu
Phone: +1 201 216 5496

Abstract - *This paper presents a synergy of Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) into a novel hybrid coordination algorithm for distributed multi-agent systems. The intended multi-agent systems are composed of relatively simple, expendable agents with highly decentralized, self-organized behaviors; which as a whole achieve global optimization over a set task. Basically, two coordination processes among the agents will be established. One is a stigmergy-based algorithm using the distributed virtual pheromones to guide the agents' movement, the other one is interaction-based algorithm, where a global maximum of the attribute values can be obtained through the interaction between the agents. The simulation results demonstrate that the proposed hybrid swarm intelligence based architecture is feasible, efficient, and robust to coordinate a simulated distributed multi-agent system.*

1. INTRODUCTION

Recently, multi-agent systems have been employed to various domains in the real-world applications, such as 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, some of these approaches are either cannot scale to large numbers or fragile to dynamic environment. To develop a self-organized and self-adaptive multi-agent system is a challenging task.

Recently, swarm intelligence (SI) has attracted extensive attentions to tackle the scalability issue for multi-agent systems while maintaining system robustness and individual simplicity. SI 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. 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. The 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.

Flocks of migrating birds and schools of fish are familiar examples of spatial self-organized patterns formed by living organisms through social foraging. Such aggregation patterns are observed not only in colonies of single-cell bacteria, social insects like ants and termites, as well as in colonies of multi-cellular vertebrates like birds and fish, but also in human societies [10]. Wasps, bees, ants, and termites all make effective use of their environment and resources by displaying collective "swarm" intelligence. For example, termite colonies build nests with a complexity far beyond the comprehension of the individual termite, while ant colonies dynamically allocate labor to various vital tasks such as foraging or defense without any central decision-making ability [11][12]. Slime mould is another perfect example. These are very simple cellular organisms with limited mobile and sensory capabilities, but in times of food shortage they aggregate to form a mobile slug capable of transporting the assembled individuals to a new feeding area. Should food shortage persist, they then form into a fruiting body that disperses their spores using the wind, thus ensuring the survival of the colony [12][13]. The signals that each ant sends out to other ants, by laying down chemical trails of pheromones, enable the ant community as a whole to find the most abundant food sources. Wilson [14] showed that ants emit specific pheromones and identified the chemicals, the glands that emitted them and even the fixed action responses to each of the various pheromones. It was found that pheromones comprises a medium for communication among the ants, allowing fixed action collaboration, the result of which is a group behavior that is adaptive where the individual's behaviors are not.

Based on the above observations and studies in biological systems, the bio-inspired swarm intelligence should provide such a property for a system that the collective behaviors of simple entities interacting locally with their environment cause coherent functional global patterns to emerge. The SI-based approaches emphasize self-organization, distributedness, parallelism, and exploitation of direct (peer-to-peer) or indirect (via the environment) local communication mechanisms among relatively simple agents.

SI provides a basis with which it is possible to solve distributed problems without centralized control or the provision of a global model. And more and more researchers investigate such SI-based approaches in various multi-agent systems. Reynold [15] 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. [16] evolved *e-boids*, groups of artificial fish capable of displaying schooling behavior. Spector et al. [17] 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. Payton et al. [18] 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 agents can send and receive directional communications to each other.

Among many successful bio-inspired swarm intelligence based computational paradigms, two well-known approaches are known as ACO (Ant Colony Optimization algorithm [19]) and PSO (Particle Swarm Optimization [20]). In this paper, a hybrid ACO/PSO collective coordination paradigm is proposed for a distributed multi-agent system. 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 [18], 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,* is proposed as the message passing coordination scheme for the swarm agents. Instead of using infrared signals for transceivers in [18], 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: ACO's autocatalytic mechanism and 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.

The strength of the proposed hybrid 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 hybrid ACO/PSO control architecture for distributed swarm agents. Section IV describes a pheromone edge pair propagation funneling method to reduce the communication propagation. Section V details the framework and structure of the hybrid algorithm. Section VI presents the simulation environment and simulation results. To conclude the paper, section VII outlines the research conclusion and the future work.

2. A Hybrid ACO/PSO Distributed Control Approach

2.1 Assumptions of Distributed Multi-Agent Systems

The objective of this study is to design a SI-based coordination algorithm for distributed multi-agent agents, which are supposed to work cooperatively to search for multiple targets in a dynamic unknown environment, and implement some predefined tasks on the detected targets. These predefined tasks can be defined as tasks depending on different applications, for example, collecting and moving tasks in collective construction, detection and collection tasks in resource/garbage detection and collection, people detection and resource delivery tasks in people search and rescue, etc.. The goal is to find and process all of the targets within a minimum time. Assume that the agents are simple, and homogeneous, and can be dynamically added to or removed from the team 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.

Generally speaking, the ideal vision of the proposed schemes is to obtain the following capabilities and attributes for each agent in a multi-agent system.

- Can interact with other agents within its interaction range;
- Can differentiate and identify swarm members from non-members or targets, and knows how to interact with environment;
- Has a priori state behavior cache, experience archives, and learning structures;
- Interacts only valid information of archives into swarm propagation, i.e. has noise reduction buffers and protocols;

- Can perform simple computational and analysis processes.

2.2 The Approach

The ACO algorithm, proposed by Dorigo et al. [19], 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 (a2a) 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.

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 virtual pheromone data structure whenever it detects a target, and then broadcast this target information to its neighbors through visual pheromone.

Let $p_k(t)=\{p_{ij}^k(t)\}$ 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.

Let $\xi_k(t)=\{\xi_{ij}^k(t)\}$ represents a set of agent intensities at time *t* respective to each pheromone in $p_k(t)$, where $\xi_{ij}^k(t)$ denotes the agent *intensity*, which is an indication of the number of agents who have received the same pheromone. at location *(i,j)*. Here "potentially" means that those agents who have received the pheromone at same location will have the intension to move to the same target. However, they may also choose other targets based on their own utility functions. To emulate the pheromone enhancement and elimination procedure in a natural world, $\xi_{ij}^k(t)$ can be updated by the following equation:

$$\xi_{ij}^k(t+1)=\rho*(\xi_{ij}^k(t)+\tau_{ij}^k)-(1-\rho)*e*\xi_{ij}^k(t) \qquad (1)$$

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

$$\tau_{ij}^k=\begin{cases}\alpha, \text{if source pheromone}\\ \beta, \text{otherwise}\end{cases} \qquad (2)$$

where $0 \le \beta < \alpha \le 1$. The source agents refer to the agents who detect the target by themselves. 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, τ_{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 $e<1$.

Let $\omega_k(t)=\{\omega_{ij}^k(t)\}$ represents a set of potential target weights respective to each pheromone in $p_k(t)$, where $\omega_{ij}^k(t)$ denotes the *target weight,* which measures potential target resources available for agent *k* at time *t*.

Finally, Let $u_k(t)=\{u_{ij}^k(t)\}$ represents a set of target utilities at time *t* respective to each pheromone in $p_k(t)$, where $u_{ij}^k(t)$ denotes the *target utility* of agent *k* , which is defined as follows:

$$u_{ij}^k(t)=\frac{1}{\lambda}(c_1\omega_{ij}^k(t)-c_2\xi_{ij}^k(t)) \qquad (3)$$

where $\omega_{ij}^k(t)$ and $\xi_{ij}^k(t)$ are target weight and agent intensity, respectively, and λ is the 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*. c_1 and c_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.

The agents are randomly distributed in the searching environment initially, where multiple targets with different sizes and some static obstacles are randomly dispersed within the environment. At each iteration, each agent adjusts its behavior based on the target utility. This utility-based approach is 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, we turned our attention to another collective intelligence - Particle Swarm Optimization (PSO) [20]. The PSO is a biologically-inspired algorithm motivated by a social analogy, such as flocking, herding, and schooling behavior in animal populations.

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.

The PSO algorithm can be represented as in (4), which is derived from the classical PSO algorithm [20] with minor redefinitions of formula variables as follows:

$$agent\_velocity = explorative + cognitive + social \tag{4}$$

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.

Basically, the above mentioned utility-based ACO approach is the social activities among the agents, where the agents propagate the pheromone information to its neighbors, which would be a perfect match to estimate the social factor in the PSO algorithm.

In terms of cognitive factor in the PSO algorithm, it is based on the local view of each agent, which can be represented by the target visibility. Let $\gamma_k(t) = \{\gamma_{ij}^k(t)\}$ represents a set of visibilities at time t respective to each pheromone in $p_k(t)$, where $\gamma_{ij}^k(t)$ denotes the target visibility for agent k in terms of target at location (i, j), which is defined by the following equation:

$$\gamma_{ij}^k(t) = r^k / d_{ij}^k(t) \tag{5}$$

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 $\gamma_{ij}^k > 1$, we set $\gamma_{ij}^k = 1$. 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.

The exploration factor can be easily emulated by random movement. The detailed velocity is updated as follows:

$$v_{ij}^k(t+1) = \phi_e * rand_e() * v_{ij}^k(t) + \phi_c * rand_c() * (p_c - x_{ij}^k(t)) + \phi_s * rand_s() * (p_s - x_{ij}^k(t)) \tag{6}$$

where, $\phi_e, \phi_c, and \phi_s$ represent the propensity constraint factors for explorave, 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. $p_s = \max(u_{ij}^k(t))$ represents the global best from the neighbors, and $p_c = \max(\gamma_{ij}^k(t))$ represents the local cognitive best.

The position adopted by agent k at time $t+1$ is updated by

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

2.3 States of the Agents

To summarize the overall behavior of each agent, the finite state machine of the agent is defined in Fig. 1. Basically, each agent has three states: search, process, and transport.

Initially, the agent randomly searches for the targets, which is at search state. When the target is detected by the agent through its local observation, the agent changes its state to process state, where the agent works on the targets depending on what kind of tasks the target represents. When the agent finishes the task on the target, it goes back to search state again for new targets. If the agent receives the target information from its neighbors and the pheromone intensity is strong enough, the agent changes to transport state, in other words, the agent is moving to the target. Once it arrives at the target, its state is changed to process. Once one target is finished, all of the agents who have landed to this target would disperse again to search for new targets.

Fig.1. The finite state machine of the swarming agent

3. The Simulation Results

To evaluate the performance of the proposed utility-based ACO and hybrid ACO/PSO algorithm in a distributed swarm agent system, we build a virtual simulation environment written in Java language. The basic simulation infrastructures are shown in Fig.2. The parameter constraints are defined as follows: the searching environment is a 2D area with 640 x 480 pixels. The local communication radius of each agent is set up as 30 pixels, and the target visibility range is set up as 10 pixels. The agents are represented by the black dots, where the aqua links connecting the dots indicate that the agents are within the local communication range, and they can exchange the pheromone information with their neighbors. The targets are represented by the different size of red dots, and the static obstacles are represented by grey rectangles.

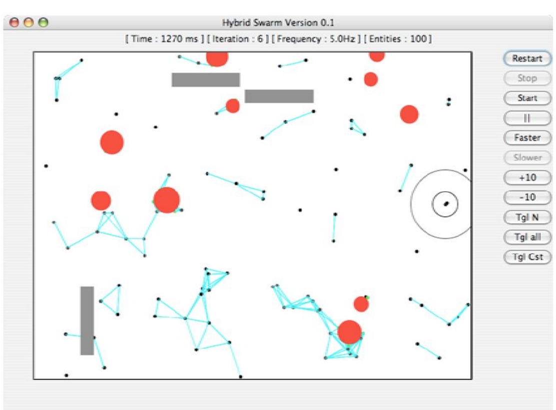

Fig. 2. A screenshot of the simulation of swarming agents

The frequency, which indicates how long each computational iteration takes, and the swarm size are displayed on the top of the simulator, which can be easily reconfigured with this user-friendly interface. For example, the simulation can be stopped, paused, and restarted at any time. The number of agents and the frequency can be increased or decreased dynamically. You have an option to show the local communication link or not. All these actions can be conducted by clicking the corresponding icons on the right side of the simulator.

This virtual simulation environment is set up as a highly dynamic system. When the restart icon is clicked, all of the agents, targets, and obstacles are randomly distributed in the environment, and the agents start moving around to search for the targets. Each agent adjusts its behavior based on the three factors (exploration, cognitive, and social) together for hybrid ACO/PSO method.

Basically, the simulation procedure is as followings: initially, the agents are randomly searching for targets. Once the targets have been detected, the agents who detected the targets send virtual pheromone to their neighbors. Each agent makes its own decision based on the proposed hybrid algorithm. Once an agent arrives at a target, it starts processing the target, which leads to the target size become smaller. After a target is finished by agents, it would disappear and the associated agents would be dispersed and search for new unfinished targets, until all of the targets have been finished. This simulation was run on an Apple Mac OSX 10.4 Tiger computer with a PPC at 1.0GHz and 768M RAM.

As we know, global path planning is very time consuming, especially for swarm agents 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. Whenever an agent detects a static obstacle within a predefined distance, it would turn right at 45 degree and move forward until the obstacle is beyond the predefined distance, then the agent move toward its original destination. If another mobile agent is detected within a predefined distance, the agent stops until other agents move away beyond a predefined distance, then it continues its movement.

To obtain the statistic performance of hybrid ACO/PSO methods, we implemented the following experiments. 10 targets are distributed in the environment with fixed positions for all the simulations, as well as the obstacles. Then, we start running the simulations with the swarm size of 50 using hybrid method, each method runs 35 times to obtain the mean and variance values. The same process is repeated for the swarm size of 60, 70, 80, 90, and 100. Since the running speeds of simulations may differ from one computer to another, the performance measurement is defined as the number of iterations. One iteration represents the time that all of the agents need to make their movement decisions once sequentially. The experimental results are shown in Table 1, where the processing time means the iteration numbers need for each simulation.

It is observed from Table 1 that the smaller the swarm size, the more time the agents need to take to finish the tasks. With the swarm size increases, the system efficiency increases while maintaining the similar robustness with smaller swarm size. The underlying reason for this is because 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.

TABLE 1: Statistics of simulation results

	60	70	80	90	100
Mean	1950	1759	1478	1278	1125
Variance	76	86	59	79	85

4. CONCLUSION AND FUTURE WORK

A hybrid ACO/PSO control schemes are proposed for the distributed swarm agents. The main characteristics of the proposed Swarm Intelligence (SI) based approaches are the use of natural metaphors, inherent parallelism, stochastic nature, adaptivitiy, and the use of positive feedback. This SI-based architecture are 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.

While the proposed SI-based approaches have the advantages in system robustness, scalability, and individual simplicity, however, the communication overhead is still a critical issue, which needs to be improved, 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.

ACKNOWLEDGMENT

This work was supported by NSF Research Education for Undergraduate program under contract number 0452235.

REFERENCES

[1] M.J.B. Krieger, J. B. Billeter, and L. Keller, "Ant-like Task Allocation and Recruitment in Cooperative Agents," *Nature*, Vol. 406, 2000, pp. 992-995.

[2] M. J. Mataric, M. Nilsson, and K. T. Simsarian, "Cooperative Multi-agent Box-Pushing", *IEEE/RSJ International Conference on Intelligent Agents and Systems*, 1995.

[3] A. Martinoli, A. J. Ijspeert, and F. Mondada, "Understanding Collective Aggregation Mechanisms: From Probabilistic Modeling to Experiments with Real Agents", *Agentics and Autonomous Systems*, Vol. 29, pp. 51-63, 1999.

[4] T. Balch and R. C. Arkin, "Behavior-based Formation Control for Multi-agent Teams", IEEE Trans. on Agentics and Automation, 1999.

[5] B. Yamauchi, "Decentralized Coordination for Multi-agent Exploration", *Agentics 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 Agents for Successful Soccer Playing", Special Issue on Advances in Multi-Agent Systems, T. Arai, E. Pagello, and L. E. Parker, Editors, *IEEE Trans. on Agentics and Automation*, Vol. 18, No.5, pp. 685-699, 2002.

[7] C.A.C. Parker and H. Zhang, Collective Agentic 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 Agentics", *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 agentic swarm. *Adaptive Behavior*. 14(1):21-51, 2006.

[10] D. Chowdhury, K. Nishinari, and A. Schadschneider, "Self-organized patterns and traffic flow in colonies of organisms from bacteria and social insects to vertebrates", *special issue on Pattern Formation, in Phase Transitions*, Taylor and Francis, Vol 77, 601, 2004.

[11] E. Bonabeau, G. Theralauz, J.L. Deneubourg, and S. Camazine, "Self-organization in social insects", *Trends in Ecology and Evolution*, 12(5), 188-193, 1997.

[12] H. Williams, Spatial organization of a homogeneous agent population using diffusive signaling and role differentiation, *MsC Evolutionary and Adaptive Systems Thesis*, COGS, University of Sussex, UK, Sept. 2002.

[13] A. F. M. Maree, and P. Hogeweg, "How amoeboids self-organize into a fruiting body: multicellullar coordination in dictyostelium discoideum", *PNAS*, Vol. 98, No. 7, pp. 3879-3883, 2001.

[14] E. O. Wilson, *The Insect Societies*, Cambridge, MA, Belknap Press, 1971.

[15] C. W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model", *Computer Graphics*, 21(4), July 1987, pp. 25-34.

[16] 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.

[17] 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. Ed. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003)*, Berlin, Germany, 2003, pp.61-73.

[18] D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee, Pheromone Robotics, *Autonomous Robots*, Vol. 11, No. 3, November, 2001.

[19] 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.

[20] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *IEEE Conference on Neural Networks*, Proceedings 1995.