A Hybrid ACO/PSO Control Algorithm for Distributed Swarm Robots

Yan Meng and Ọlọrundamilọla Kazeem
Department of Electrical and Computer Engineering
Stevens Institute of Technology, New Jersey, USA
yan.meng@stevens.edu, okazeem@stevens.edu

Juan C. Muller
Department of Computer Science and Mathematics
New Jersey City University, New Jersey, USA
jmuller@njcu.edu

***Abstract*— In this paper, we present a hybrid Ant Colony Optimization/Particle Swarm Optimization (ACO/PSO) control algorithm for distributed swarm robots, where each robot can only communicate with its neighbors within its communication range. A virtual pheromone mechanism is proposed as the message passing coordination scheme among the robots. This hybrid ACO/PSO architecture adopts the feedback mechanism from environment of ACO and the adaptive interplay among agents of PSO to create a dynamic optimization system, and it is well-suited for a large scale distributed multi-agent system under dynamic environments. Furthermore, a pheromone-edge pair propagation funneling method is developed to reduce the communication overhead among robots. The simulation results concretely demonstrate the robustness, scalability, and individual simplicity of the proposed control architecture in a swarm robot system with real-world constraints.**

I. 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. Such ability is crucial for developing robots in human environments. Swarm robots are often observed to display many of the attributes, such as robustness, adaptability, flexibility, and self-organization, which are typical in collective intelligent system in general.

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 constituent 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. 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.

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.

Payton et al. [14] 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.

The major contribution of this paper is that a SI-based coordination paradigm, i.e., a hybrid Ant Colony Optimization (ACO)/Particle Swarm Optimization (PSO), is proposed to achieve an optimal group behavior for large number of small-scale robots. Each robot 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 [14], inspired by

1-4244-0708-7/07/$20.00 ©2007 IEEE

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 robots. Instead of using infrared signals for transceivers in [14], 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 robots. 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.

II. Problem Statement

The objective of this study is to design a SI-based coordination algorithm for distributed swarm robots, which are supposed to work cooperatively to search for multiple targets in a dynamic unknown environment, and implement predefined tasks on the detected targets. 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.

Let $A=\{a^k\}$ denote a set of swarm agents *a,* where $0 < k \leq N$, k is the swarm *identification number*, and N is the swarm population size. Generally speaking, the ideal vision of the proposed schemes is to obtain the following capabilities and attributes for each agent in a swarm agent system.

- Can interact with other agents, which is a subset of A, if such a subset agents are within the a^k*'s* 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;
- Has infinite "desire" to accomplish a priori swarm wide objectives.

III. A Hybrid ACO/PSO Distributed Control Approach

A. Utility-Based ACO Approach

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 (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.

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.

Let $\tau_{a^k}(t) = \{\tau_{ij}^k(t)\}$ represents a set of agent intensities at time t respective to each pheromone in $p_{a^k}(t)$, where $\tau_{ij}^k(t)$ denotes the agent *intensity*, which 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 with the same target. However, they may also go to other targets with stronger pheromone intensity based on their local decisions. To emulate the pheromone enhancement and elimination procedure in a natural world, $\tau_{ij}^k(t)$ is updated by the following equation:

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

where $0<\rho<1$ 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

$$T_{ij} = \begin{cases} \alpha, \text{if source pheromone} \\ \beta, \text{otherwise} \end{cases} \tag{2}$$

where $0 \le \beta < \alpha \le 1$. 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 $e < 1$.

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

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

$$\mu_{ij}^k(t) = (k_1\omega_{ij}^k(t) - k_2\tau_{ij}^k(t))/R \tag{3}$$

where $\omega_{ij}^k(t)$ and $\tau_{ij}^k(t)$ are target weight and agent intensity, respectively, and R 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. 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.

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 ACO 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.

B. A Hybrid ACO/PSO Approach

To prevent the local optima scenarios in utility-based ACO approach, we turned our attention to another collective intelligence - Particle Swarm Optimization (PSO) [15]. 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 [15] with minor redefinitions of formula variables as follows:

$$v_{ij} = explorative + cognitive + social \qquad (4)$$

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.

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 $\eta_{a^k}(t) = \{\eta_{ij}^{k}(t)\}$ represents a set of visibilities at time t respective to each pheromone in $p_{a^k}(t)$, 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:

$$\eta_{ij}^{k}(t) = r^{k} / d_{ij}^{k}(t) \qquad (5)$$

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 $\eta_{ij}^{k} > 1$, 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.

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

$$v_{ij}^{k}(t+1) = \psi_e * rand_e() * v_{ij}^{k}(t) + \psi_c * rand_c() * (p_c - x_{ij}^{k}(t))$$
$$+ \psi_s * rand_s() * (p_s - x_{ij}^{k}(t)) \qquad (6)$$

where, $\psi_e, \psi_c, and \psi_s$ 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. $p_s = \max(\mu_{ij}^{k}(t))$ represents the global best from the neighbors, and $p_c = \max(\eta_{ij}^{k}(t))$ represents the local cognitive best.

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

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

C. Finite State Machine of the Swarm 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.

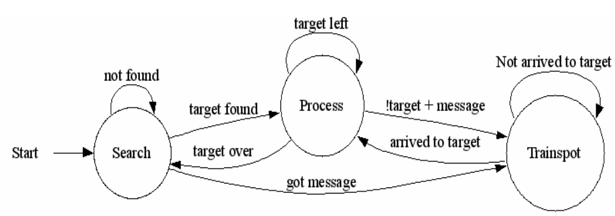


Fig.1. The finite state machine of the swarming agent

IV. A PROPAGATION FUNNELING METHOD OF AGENT-TO-AGENT INTERACTION

As the population of the swarm agents increases, the interaction and communication among the agents becomes expensive since the communication overhead may increase exponentially. Without a structured noise reducing protocol, the proposed hybrid meta-heuristic architecture may become infeasible in a large scale multi-agent system. To reduce the communication propagation, a pheromone-edge-pair (PEP) propagation funneling method is proposed here.

First, let's define

$$G_{a^k}(t) = \{listof(p_{ij}, e_{kn})\},$$
$$\text{where} \quad G_{a^k}(t) \le G_N, 0 < k \le N, 0 < n \le N \quad (8)$$

where $G_{a^k}(t)$ represents a list of all pheromone-edge pairs obtained by agent k from its neighboring agent n ($0 < n < N$) at time t, where N is the swarm size. p_{ij} denotes the pheromone at coordinate (i, j), and G_N represents the maximum number of agent-to-agent interaction connections allowed for the swarm of size N. In other words, G_N represents the situation where all agents are within interaction rage of each another. e_{kn} represents the edge interaction of a^k connecting to a^n , where k and n are swarm identification number, and $k \neq n$ (i.e. no self interactions).

On a Cartesian plane C, a maximal graph G_C of agent-to-agent interactions over edges of pheromones is defined. Even for a G_C configuration of agents, the explosion of agent-to-agent interaction would unlikely happen considering random search behavior, given physical boundaries of agents, swarm population size, limited local range of interaction, and number of pheromones in propagation. This funneling method dictates an out-propagation of distinct pheromones, p_{ij}, over the given edges, e_{kn}, from a target coordinate, which defines pheromone-edge pairs, $\{p_{ij}, e_{kn}\}$. This data structure dictates valid and invalid interaction states to be received by an agent as the following four cases:

1) *same pheromone—different edge (valid)*
2) *different pheromone—same edge (valid)*
3) *different pheromone—different edge(valid)*
4) *same pheromone—same edge (invalid)*

The reason we set case 1 as valid state is because we need to track the redundancy number in (3) of potential agents who may approach to the same target in the future. Case 4 may happen when the propagation agent receives the same pheromone from difference neighbors, and then it sends the pheromone to the same agent multiple times.

As agents send and receive virtual pheromones, the associated pheromone-edge pairs are archived in $G_{a^k}(t)$. For every future interaction, the interaction archive must be parsed and checked for valid states of interaction pairs. If it is a valid state, the agent stores the received pheromones into $G_{a^k}(t)$. Otherwise, it would ignore the received pheromones.

The PEP propagation funneling method significantly reduce the communication overhead of the swarm system, especially when the swarm size is increased to a large scale.

V. THE FRAMEWORK OF THE HYBRID METHOD

The framework of the hybrid algorithm will be described in this section. At time $t=0$, the agents, a^k ($0 < k \le N$), are randomly distributed over the searching environment, the pheromone list for each agent is initialized as empty, and the associated pheromone attributes $\eta_{ij}^k(0), \tau_{ij}^k(0)$, and $\mu_{ij}^k(0)$ are set to zero. η_{ij}^k represents the agent's propensity to cognition or greedy localizing behavior. τ_{ij}^k represents the number of agents that would potentially adopt a target of the associated pheromone, and μ_{ij} represents an agent's disposition to socialize and adopt a non-greedy global behavior.

Over the iterations to optimization, swarm agents perform space transition from one position to another, which are related to transitions in behavior space. An agent continues performing randomized space transitions unless either an agent directly acquires a target (becoming the source agent of a pheromone), or indirectly acquires a target, i.e. receives a pheromone(s) from its neighbors (becoming the propagation agent). Under this situation, cognitive factor and social factor would not be considered. When $\{p_{ij}^k\}$ is not empty, it means that one or more targets has been detected either by the agent itself or other agents, the cognitive and social factors would be taken into considerations for optimal global behaviors.

The hybrid ACO/PSO algorithm is summarized as follows:

At t = 0, N agents are randomly distributed in a 2D searching plane, which is a subset of the Cartesian plane;
For each agent, set the pheromone list as empty $\{p_{ij}^k\} = \{\}$;
For k = 1 to N
  Set $\psi_c = \psi_s = 0$;
  Compute $v_{ij}^k(t+1)$ *using (6)*
  Compute $x_{ij}^k(t+1)$ *using (7)*
Endfor

While (not all of the tasks on the targets have been finished || the iteration number is less than a preset threshold)
  For k = 1 to N
  do
    If (target is not found) and (pheromone(s) is not received)
      Set $\psi_e \neq 0$;
      Set $\psi_c = 0$;
      Set $\psi_s = 0$;
      Compute $v_{ij}^k(t+1)$ *using (6);*
      Compute $x_{ij}^k(t+1)$ *using (7);*
    Else if (target is found by agent itself)
      Set $\psi_e = 0$;
      Set $\psi_c \neq 0$;
      Set $\psi_s = 0$;
      Compute $v_{ij}^k(t+1)$ *using (6);*
      Compute $x_{ij}^k(t+1)$ *using (7);*
    Else (pheromone(s) is received)

Update $\{p_{ij}^k\}$*, pheromone list is not empty;*
For $m = 1$ *to size of* $\{p_{ij}^k\}$
Calculate T_{ij} *using (2);*
Calculate $\tau_{ij}(t)$ *using (1);*
Calculate $\eta_{ij}(t)$*using(5);*
if $\eta_{ij}(t) > 1.0$*, then* $\eta_{ij}(t) = 1.0$*;*
Calculate $\mu_{ij}(t)$ *using (3);*
EndFor
Set $\psi_e \neq 0$*;*
Set $\psi_c \neq 0$*;*
Set $\psi_s \neq 0$*;*
Compute $v_{ij}^k(t+1)$ *using (6);*
Compute $x_{ij}^k(t+1)$ *using (7);*
Endfor
Set $t = t + 1$

VI. 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 infrarastructures 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.

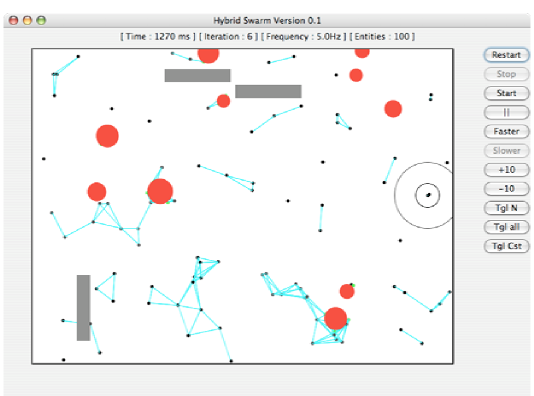
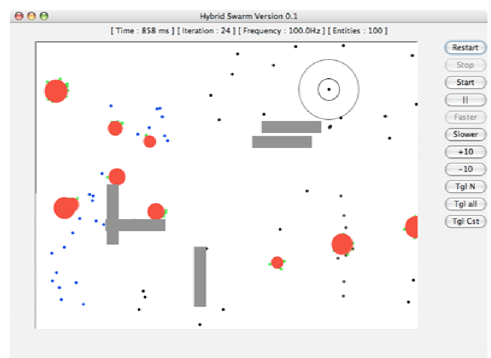


Fig. 2. A screenshot of the simulation of swarming agents

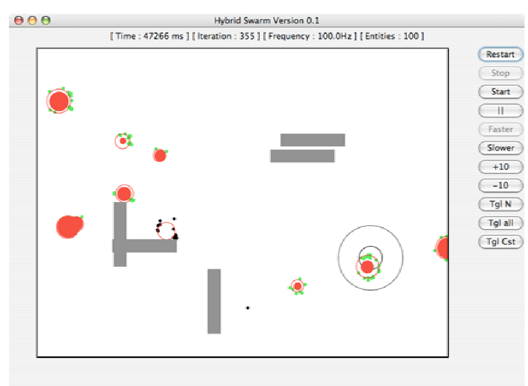
The frequency, which indicates how long each computational iteration takes, and the swarm size are displayed on the top of the simulator, which can be easily reconfigured with this user-friendly interface. For example, the simulation can be stopped, paused, and restarted at any time. The number of agents 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 target utility value for utility-based ACO (UB-ACO) method, or three factors (exploration, cognitive, and social) together for hybrid ACO/PSO method.

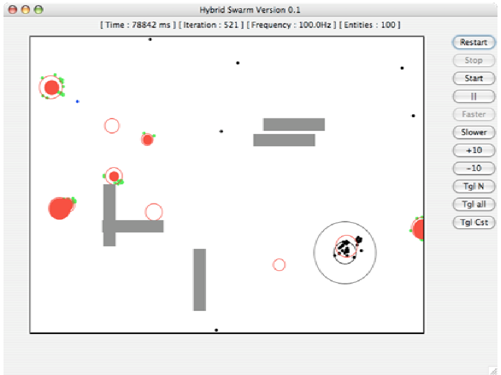
Fig. 3 shows a set of sequential snapshots of the simulation using the hybrid method. Initially, the agents are randomly searching for targets, as shown in Fig. 3(a). Once the targets have been detected, the agents who detected the targets send virtual pheromone to their neighbors. Each agent makes its 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, as shown in Fig. 3(b). After a target is finished by agents, it would disappear and the associated agents would be dispersed and search for new unfinished targets, as shown in Fig. 3(c), and Fig. 3(d), until all of the targets have been finished, as shown in Fig. 3(e). This simulation was run on an Apple Mac OSX 10.4 Tiger computer with a PPC at 1.0GHz and 768M RAM.



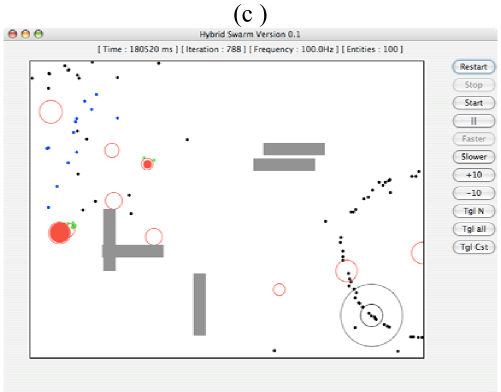
(a)



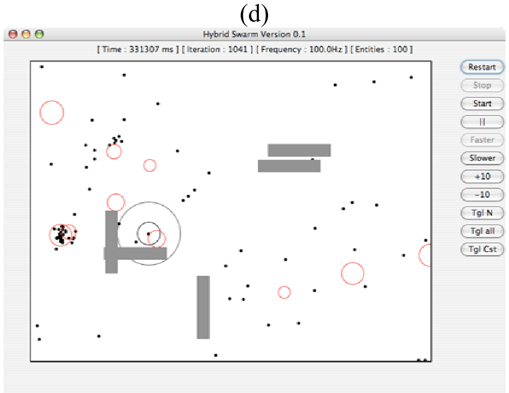
(b)



(c)



(d)



(e)

Fig.3. Sequential snapshots of a simulation with 100 agents, 10 targets and 5 static obstacles using hybrid ACO/PSO approach.

As we know, 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. 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 both UB-ACO and 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 both methods, 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 Fig. 4 and Fig.5, where the processing time means the iteration numbers need for each simulation.

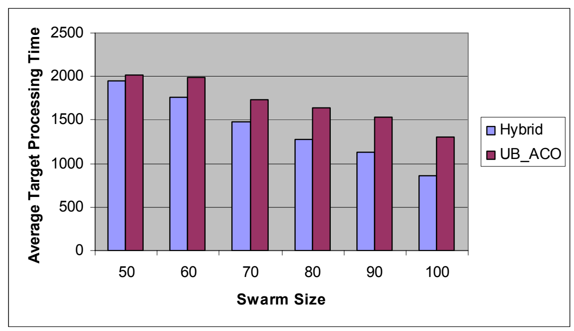


Fig. 4. The average target processing time of hybrid and utility-based ACO methods in terms of different swarm sizes from 50 to 100. Unit of time is iteration.

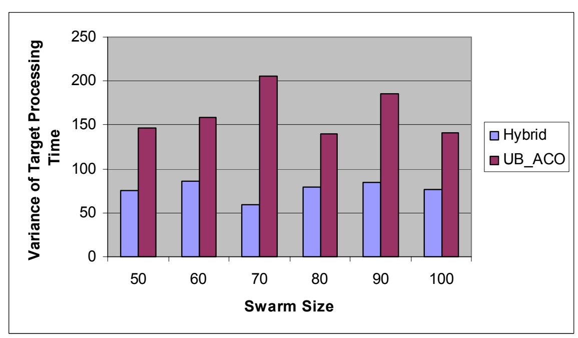


Fig. 5. The variance of target processing time of hybrid and utility-based ACO methods in terms of different swarm sizes from 50 to 100. Unit of time is iteration.

It is observed from Fig. 4 that when the swarm size is 50, there is very small difference in the average targets processing time between the utility-based ACO method and hybrid method.

When the swarm size is increased, it can be seen that the hybrid method outperform the utility-based ACO method in a significant way, especially in the case of 100 agents. The reason behind this observation is because the agents using UB-ACO 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 new 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, it can be seen from Fig. 5 that the hybrid method is more stable and consistent than the UB-ACO method.

VII. 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. These 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 at Stevens Institute of Technology, New Jersey, USA.

References

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

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