LIVS: Local Interaction via Virtual Stigmergy Coordination in Distributed Search and Collective Cleanup

Yan Meng and Jing Gan, Member, IEEE

Abstract—Distributed coordination is critical for a multi-robot system in hazardous waste cleanup under a dynamic environment. To achieve higher efficiency as well as robustness, a bio-inspired Local Interaction via Virtual Stigmergy (LIVS) coordination approach is proposed in this paper. This new meta-heuristic integrates two mechanisms - stigmergy-based autocatalytic mechanism and Particle Swarm Optimization cognitive capabilities through local interaction — into one efficient approach. The proposed LIVS algorithm has been implemented on both self-developed simulator and embodied robot simulator Player/Stage in a searching task. The simulation results demonstrate the feasibility, robustness, and scalability of the methods with real-world constraints.

I. INTRODUCTION

SEARCHING targets/objects in an unknown dynamic environment using multi-robot systems has various real-world applications, such as hazardous waste cleanup, urban search and rescue, surveillance systems, and monitoring in military combat environments.

Distributed multi-robot systems demand group coherence (robots need to have the incentive to work together faithfully) and group competence (robots need to know how to work together well). The distributed coordination is desirable for multi-robot systems under a dynamic environment due to its robustness, flexibility, and reliability. Therefore, the main challenge for multi-robot systems is to develop intelligent robots that can adapt their behaviors based on interaction with the environment and other robots, become more proficient in their tasks over time, and 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 robots acting autonomously based on their own individual decisions.

One of the major issues for multi-robot systems, especially for large scale systems, is the communication intensity. Most of the available multi-robot systems rely on extensive global communication for cooperation of swarm robots, which may yield stressing communication bottlenecks. Furthermore, it is difficult, if not impossible, for each simple robot to maintain global communication with very limited on-board power and communication capability under a dynamic environment.

Therefore, more researchers turned their attention to the biological systems for inspirations. Reynolds [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 genetic programming to evolve group behaviors for flying robots in a simulated environment. The swarm-bot was developed as a new robotic system [13] 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. 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 stigmergy robotics, which was modeled after the chemical insects, such as ants, use to communicate. Pugh and Martonioni [16] proposed a group learning algorithm using Particle Swarm Optimization for a multi-robot system. Werfel and Nagpal [17] proposed an extended stigmergy by increasing the capabilities of environmental elements in swarm robots to automatically assemble solid structures of square building blocks in two dimensions according to a high-level user-specific design.

In this paper, a novel bio-inspired coordination paradigm is proposed to achieve an optimal group behavior for multi-robot systems, which is based on Local Interaction via Visual Stigmergy (LIVS). Basically, two coordination processes among the robots are established in the proposed architecture. One is a virtual stigmergy based algorithm to guide the robots’ movements for targets, where each robot has its own virtual stigmergy matrix that can be created, enhanced, evaporated, and propagated to its neighboring robots. The other one is Particle Swarm Optimization (PSO)’s cognitive capabilities through local interaction, which aims to achieve the balance for each robot between the exploration and exploitation through the interactions among the robots.
The paper is organized as follows: Section II describes the problem statement. The proposed LIVS scheme is discussed in Section III. Section IV describes a propagation funneling method to reduce the inter-robot communication. Section V presents the simulation results using the self-developed simulator and Player/Stage simulator. To conclude the paper, Section VI outlines the research conclusions and the future work.

II. PROBLEM STATEMENT

The objective of this study is to design an efficient and robust distributed coordination algorithm for multi-robot systems, especially for a hazardous waste cleanup task under a dynamic environment. The targets can be defined as any kind of predefined hazardous waste. It is assumed that the searching area is bounded and robots can detect the hazardous materials using special on-board sensors, such as camera systems or other sensors. The robot can only detect the targets within its local sensing range. Once a robot detects a target, it cleans up the target assuming the cleanup time is proportional to the size of target. Assume that the robots are simple, and homogeneous. Each robot can only communicate with its neighbors. Two robots are defined as neighbors if the distance between them is less than a pre-specified communication range. The goal is to detect and cleanup all of the targets within the searching area as soon as possible.

III. LOCAL INTERACTION VIA VIRTUAL STIGMERGY (LIVS) COORDINATION APPROACH

A. Virtual Stigmergy Mechanism

Stigmergy is a class of mechanisms that mediate animal-animal interactions through artifacts or via indirect communication, providing a kind of environmental synergy, information gathered from work in progress, distributed incremental learning and memory among the society. To emulate stigmergy-based communication in swarm robots, special stigmergy materials and associated detectors need to be designed, and most of the time such chemical/physical stigmergy is unreliable and easily modified under some hazardous environments, such as urban search and rescue. A modification of this autocatalyst is necessary. Similar to [15], a unique virtual robot-to-robot interaction mechanism, i.e. virtual stigmergy, is proposed as the message passing coordination scheme in this paper. Instead of using infrared signals for transceivers in [15], which requires line of sight to transmit and receive, ZigBee is used, which is smaller, cheaper, and power efficient compared to 802.11b wireless card.

Each robot in the environment is associated with one unique stigmergy, which can be enhanced or evaporated over time to adapt to a dynamic environment. Initially, each robot creates its own virtual stigmergy matrix, which installs all stigmergy information associated with different targets. Whenever a robot detects a target, it would update its own stigmergy matrix and broadcast this target information to its neighbors through a visual stigmergy package.

B. Target Utility and Target Visibility

In order to design a stigmergy-based behavior, inspired by Chialvo and Millonas in [18], a simple model to estimate the stigmergy weighting function is defined as

$$w_{ij}^k(t) = \left(1 + \frac{\tau_{ij}^k(t)}{1 + \tau_{ij}^k(t)/\delta^k}\right)^\lambda .$$

This function measures the relative probability of robot $k$ moving to a target at $(i,j)$ with stigmergy density $\tau_{ij}^k(t)$. The parameter $\lambda$ controls the degree of randomness with which each robot follows the gradient of stigmergy. $\delta^k$ denotes the sensor capability of robot $k$ to sense stigmergy decreases at high concentrations.

To emulate the stigmergy creation, enhancement and elimination procedure in natural world, the stigmergy density $\tau_{ij}^k(t)$ can be updated by the following equation:

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

where $\tau_{ij}^k(t)$ represents the stigmergy density at time $t$, which is an indication of the number of robots who will potentially process the corresponding target at location $(i,j)$. When we say "potentially", we mean all the robots who have received the same stigmergy information may end up moving to the same target. However, they may also go to other targets with stronger stigmergy intensity based on their local decisions. $0<\rho<1$ is the enhancement factor of stigmergy density. $T_{ij}^k$ is the stigmergy interaction intensity received from the neighboring robots for a target at $(i, j)$, which is defined as

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

where $0<\beta<\alpha<1$. If a robot discovers a target by itself, it is defined as the source robot. The source robot then propagates the source stigmergy, to its neighbors. A propagation robot is a non-source robot, and simply propagates stigmergy it received to its neighbors. Basically, $T_{ij}^k$ is used for stigmergy enhancement. $m$ represents the elimination factor. In the ants system, the stigmergy 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 stigmergy trail is totally eliminated, it means that no target is needed to be processed through this stigmergy. To slow down the elimination relative to enhancement, we set $m<1$.

To define the probability that robot $k$ moves toward the target at $(i, j)$ with stigmergy weighting function $w_{ij}^k(t)$, the target utility function is defined as following:

$$\mu_{ij}^k(t) = w_{ij}^k(t)e^{-\gamma}$$

where $\gamma$ represents local target redundancy, which is defined as the number of the local neighbors who have sent the stigmergys referring to the same target at $(i, j)$ to robot $k$. 

1372
Generally speaking, the higher the target utility is, the more attractive the corresponding target is to the robot. Therefore, the benefit of moving to this target would be higher in terms of the global optimization. If the local target redundancy is high, it means that there will be more potential robots (globally) moving to this target, which may lead to the less available material (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 robots within a local neighbor move to the same target instead of exploring new targets elsewhere.

Initially, the robots 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 robot adjusts its behavior based only on the target utility, it may lead the robot to be very greedy in terms of the robots’ behaviors, since the robots would rather move to the target with higher utility than explore new areas. This greedy behavior of the robots may easily lead to local optima.

To prevent the local optima scenarios in this utility-based approach, we have to take target visibility into consideration, which is defined as:

$$\eta^k_{ij}(t) = \max\left(\frac{r^k}{d^2_{ij}(t)}, 1\right)$$

(5)

where \(\eta^k_{ij}(t)\) denotes the target visibility for robot \(k\) in terms of target at location \((i, j)\), \(r^k\) represents the local detection range of robot \(k\), and the \(d^2_{ij}(t)\) represents the distance between the robot \(k\) and the target at location \((i, j)\). When the target visibility is higher, it means the distance between the target and the robot is smaller, it would be of more benefit to move to this target due to its less cost compared to moving to a more distinct target under the same environmental condition.

### C. Coordination of Robot Behaviors

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 robot. To tackle this issue, we turned our attention to 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).

Basically, the PSO algorithm can be represented as in (6), which is derived from the classical PSO algorithm [14] with minor redefinitions of formula variables as follows:

$$v_j = \text{explorative} + \text{cognitive} + \text{social}$$

(6)

where \(v_j\) is the velocity of a robot. To determine which behavior is adopted by robot \(k\) of the swarm, the velocity, \(v_{ij}^k(t)\) has to be decided first. If the received stigmergy intensity is high, the robot 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 robot, then the velocity of the robot would prefer the cognitive factor to the social factor. Furthermore, at any given time, the velocity of the robot 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 robot respective views), and social factor (global swarm wide views) are considered simultaneously and try to merge these three factors into consistent behaviors for each robot.

The exploration factor can be easily emulated by random movement. A direct PSO adoption to swarm robots would be difficult, because swarm robots 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 robots is proposed here. This new meta-heuristics draws on the strengths of the autocatalytic mechanism of stigmergy-based systems through environment and PSO's cognitive capabilities through interplay among robots.

The challenging part is how to define local best (cognitive factor) and global best (social factor). One straightforward 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.

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 robot would control its movement behaviors by following this equation:

$$v^k_{ij}(t + 1) = \psi^e \cdot \text{rand}_e() \cdot v^k_{ij}(t) + \psi^c \cdot \text{rand}_c() \cdot (p_c - x^k_{ij}(t)) + \psi^s \cdot \text{rand}_s() \cdot (p_s - x^k_{ij}(t))$$

(7)

where, \(\psi^e, \psi^c, \text{and} \psi^s\) represent the propensity constraint factors for explosive, cognitive, and social behaviors, respectively, \(0 \leq \text{rand}(x) < 1\) where \(\Theta = e, c, \text{or} s\), and \(x^k_{ij}(t)\) represents the position of robot \(k\) at time \(t\). \(p_s = \max(\mu^k_{ij}(t))\) represents the global best from the neighbors, and \(p_c = \max(\eta^k_{ij}(t))\) represents the local cognitive best. The position of each robot \(k\) at time \(t + l\) can be updated by

$$x^k_{ij}(t + 1) = x^k_{ij}(t) + v^k_{ij}(t + 1).$$

(8)
IV. INTER-ROBOT COMMUNICATION

In the LIVS algorithm, inter-robot communication would become extensive as the scale of robot team increases. Among those inter-robot communications, some of the messages are redundant due to the random movement of robots. Therefore, elimination of the redundant communication is necessary.

For multi-robot communication, if we can trace the stigmergy’s passing route, we can easily eliminate the duplicative transferring of stigmergy and estimate the local target redundancy \( \gamma \), which is defined as the number of the local neighbors who have sent the stigmergys referring to the same target at \((i, j)\) to robot \(k\).

Here, an IP tracing method is proposed to eliminate the duplicate stigmergy communication. Basically, the tracing routing path of IP datagram head structure is added into stigmergy data structure, where the storage area can save up to \(N\) nodes assuming there are \(N\) robots in total. Inside each stigmergy data structure, there are \(N\) node bits defined, and each node bit can be marked by 1 if the current stigmergy has been passed through the node, otherwise, the node bit is marked as 0. When robots propagate the stigmergy to its neighbors, all neighborhood robots would first check who are within their own neighbors. Then, they would check whether this stigmergy has been transferred through those neighborhood robots by checking the corresponding node bits.

Other than this communication funneling feature for stigmergy communication, the node bits are also used to calculate the target redundancy by summing all of non-zero node bits together, which can be defined as

\[
\gamma = \sum_{i=1}^{N} node(i)
\]

(9)

V. SIMULATION AND EXPERIMENTAL RESULTS

A. Simulation Results using a Self-Developed Java Simulator

To evaluate the performance of the LIVS algorithm in a distributed swarm robot system, first we use our self-developed JAVA-based virtual simulator.

The searching environment is a 2D area with 640 x 480 pixels. The local communication radius of each robot is 30 pixels, and the target visibility range is 10 pixels. The robots are represented by the black dots, where the aqua links connecting the dots indicate that the robots are within the local communication range, and they can exchange the stigmergy information with their neighbors. The targets are represented by the different size of red circles, and the static obstacles are represented by grey rectangles.

Fig. 1 shows a set of sequential snapshots of the simulation using the LIVS method. Initially, the robots are randomly searching for targets. Once the targets have been detected, the robots who have detected the targets send virtual stigmergy to their neighbors. Each robot makes its own decision based on the LIVS algorithm. Once a robot arrives at a target, it starts cleaning up the target, which leads to the target size become smaller, as shown in Fig. 1(a).

After a target is cleaned up by robots, the target becomes empty and the associated robots around this target would be dispersed and search for new unfinished targets, as shown in Fig. 1(b) and 1(c) 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.

Whenever a robot detects a static obstacle within a predefined distance, it would turn right at 45 degrees and move forward until the obstacle is beyond the predefined distance, then the robot move toward its original destination. If another mobile robot is detected within a predefined distance, the robot stops until other robots move away beyond a predefined distance, then it continues its movement.

To evaluate the performance of the LIVS algorithm, two other methods are carried out for comparison. One is random movement (Random), where all robots search for targets in a random fashion. Second is the utility-based (UB), where the robots make their movement decision based on utility value defined in Equation (4) only. To obtain the statistic performance, 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 three 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 (i.e., time steps). One iteration represents the time that all of the robots need to make their movement decisions once sequentially. The experimental results are shown in Fig. 2.

From Fig. 2, it is obvious that LIVS outperforms the other two methods, where random is the worst. With the increase of the swarm size, the LIVS becomes more efficient than the UB method in a significant way, especially in the case of 100 robots. The reason behind this observation is because the robots using UB are extremely greedy and would always try to achieve the best utility. Therefore, they would rather move to detected targets with highest utilities than explore new areas. On the other hand, the LIVS 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 robots using the LIVS method to be more dispersed for different targets, which may lead to efficient searching results.

![Mean Values](image1)

![Variance Values](image2)

Fig. 2. The mean and variance of cleanup times with different swarm sizes from 50 to 100.

One interesting scenario we observed from the simulation is that when the robot receives the stigmergy information of multiple targets, sometimes the movement of the robot becomes absurd since it is trying to pick which target to move toward. Furthermore, it can be seen from Fig. 2 that the LIVS method is more stable and consistent than the other two.

B. Simulation Results using Embodied Robot Simulator Player/Stage

To take the real world robot constraints into considerations, Player/Stage is selected as our embodied robot simulator. As shown in Fig. 3, the environment is an open space with the size of 41.8m x 45.1m, where several targets are distributed randomly with different colors. 20 homogeneous Pioneer 3DX robots are used, where each one is equipped with a camera system, a laser range finder, a sonar sensor, and a wireless communication card. 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 in a circle instead of an arc.

It is not allowed to dynamically change the target configuration during run time in Player/Stage, therefore, the main focus here is to evaluate if the LIVS algorithm is feasible and robust with the real-world robot constraints under a dynamic environment. It is assumed that the initial positions of all robots relative to a global coordinate frame of searching area are given, odometry-based robot localization is employed. A color blob detection using on-board camera system is carried out for target detection.

An open space cleanup experiment is conducted, as shown in Fig. 3. Initially, the robots are randomly searching for targets, as shown in Fig. 3(a) at t = 17. Once a robot detects a target, it would propagate the stigmergy of this target to its neighbors, as shown in Fig. 3(b), where a small rectangle beside a robot indicates that the on-board vision system has detected the targets. After receiving a stigmergy message, robots make their own movement decisions based on LIVS algorithm and eventually converge to the targets, as shown in Fig. 3(c).

To evaluate the robustness of the LIVS algorithm under a dynamic environment, another set of experiments are conducted in an open space. Since it is not allowed to dynamically change the target configuration in Player/Stage, the target relocations are conducted manually. As shown in Fig. 4, initially, 20 robots search for targets to cleanup. After some robots have converged to a target, the target is manually relocated to somewhere else. All of the robots around this target would disperse to explore new areas for new targets until all of the robots converge to the targets where the target relocations are conducted manually. It can be seen that the LIVS algorithm is very robust to adapt to the unexpected events, such as target relocation.

VI. CONCLUSION AND FUTURE WORK

The proposed LIVS algorithm has the following characteristics: (1) Robots act independently, asynchronously, and in parallel, without maintaining a global model; (2) Robots use a simple control algorithm regardless of the changes under a dynamic environment; and (3) Robots can only communicate with their neighbors to share information. These characteristics of the LIVS system provide the multi-robot systems the ability to adapt to any unexpected changes in a dynamic environment. In the future work, we will focus on the several issues of the LIVS method, such as energy efficiency, communication efficiency, and more accurate self-location.
REFERENCES


Fig. 3. 20 robots search for randomly distributed targets in an open space on a player/stage simulator at t = 17, 50, and 277 time steps.

Fig. 4. Simulation in an open space with dynamic target relocations.