Design and Analysis of Cooperative and Non Cooperative Stigmergy-based Models for Foraging

Ouarda Zedadra, Hamid Seridi, Nicolas Jouandeau, Giancarlo Fortino

Design and Analysis of Cooperative and Non Cooperative Stigmergy-based Models for Foraging

Ouarda Zedadra and Hamid Seridi
LabSTIC, 8 may 1945 University
P.O.Box 401, 24000 Guelma
Department of computer science
Badji Mokhtar-Annaba University
P.O.Box 12, 23000 Annaba, Algeria
Email: zedadra_nawel1, seridihamid@yahoo.fr

Nicolas Jouandeau
LIASD, Paris8 University
Saint Denis, France
Email: n@ai.univ-paris8.fr

Giancarlo Fortino
DIMES, Universita' della Calabria
Via P. Bucci, cubo 41c - 87036
Rende (CS) - Italy
Email:g.fortino@unical.it

***Abstract*—Swarm robotics is focused on implementation of systems which are composed of multiple simple robots rather than one single complex robot. The challenge is to develop a group of robots with simple perception and communication capabilities in order to complete a task in a collective and distributed manner without central leader. In this paper, we present cooperative and non cooperative models for foraging that exploit stigmergy in the context of the classical Army Ant Raid model. Such models use the S-MASA algorithm that produces a gradual search around the nest which provides less time for locating closest food and provides close-to-optimal paths using only the pheromone concentration. The proposed models have been evaluated under simulation with respect to models based on the reference c-marking algorithm. The obtained results show that the proposed models, specifically the cooperative one outperforms the c-marking based models both in obstacle-free and obstacle environments.**

I. Introduction

Eusocial insects (such as ants and bees) are one of the best examples of collectively intelligent systems [1]. The sophisticated and intelligent collective behavior observed in such societies is caused by simple, local and individual rules [2]. Foraging algorithms used in multi-agent and robotics, have drawn much inspiration from the studies of social insects, especially from the social insect-inspired 'swarm behaviors' methods [3]. In this work, we focus on the problem of foraging for agent swarms. Foraging algorithms enable a collection of robots to search a space for a goal (the 'food'), then return it incrementally to the nest.

Behavior-based modeling is extensively used in robotics implementation [1] [4] [5] [6] [7] . It is defined as multiple behaviors within one algorithm, in which the agent can switch between the behaviors according to surrounding events. Social insects exhibit pheromone-based interaction to coordinate their actions. The early researches on gradient-based foraging approaches utilized this indirect pheromone interaction mechanism known as stigmergy. Such as ants use pheromones to mark trails in the environment, which allow them to efficiently communicate the location of food and collectively transport them to the nest [8], agents interact by leaving and sensing artificial pheromones in the environment; creating by the way a gradient between the food location and the nest. They adopt an adaptive behavior between two algorithms, first one works good if the food is close to nest, if not they switch to sweeper to get food further away. If with the two algorithms the food still not located they choose the random walk [9]. There have been many approaches to the real implementation of pheromones such as. physical marks where robots can physically mark trails between sources and the nest in different ways (leaving alcohol [10], odor [11] or RFID tags [12], by using existing communication channels, in [13] [14] [15] robots share trails of points using wireless network, by using virtual pheromones transmitted over infrared based communication in which robots can measure the intensity of IR reception to estimate the distance to the transmitter [16] and using deployable beacons [17] where robots can deploy a movable or non-movable beacons while moving in their environment. However, it is still a burden and needs to be understood and improved by simulations.

An original approach, that allows agents to build optimal paths for foraging using simple reactive agents was proposed in [18], they define agents that do not use pheromones, but instead write a gradient (integer values) as they explore the environment. The c-marking agents model proposed in [18], uses pseudo random walk for exploration. This model provides some drawbacks such as: the large amount of time spent in search, and the large amount of time needed to build optimal paths. S-MASA algorithm had contributed to eliminate these drawbacks. It is a stigmergy-based algorithm for multi target search [19] which produces a vortex around a central location (starting point), this feature provides shorter paths. The Army Ant Raid model involves identical agents moving on a two-dimensional discrete lattice where a pheromone field is created and maintained. Agents choose one of the three frontal grid cells according to the pheromone field. It uses a search algorithm to finding best-fit solutions based on a fitness function which takes into account that the maximum number of food items must be brought in at the smallest cost for the colony [20].

We propose two ant colony foraging models named S-ACF no-coop (for 'Stigmergic Ant Colony Foraging without cooperation') and S-ACF coop (for 'Stigmergic Ant Colony Foraging with cooperation'). These models combine the S-MASA algorithm [19] and the Army Ant Raid model [20]. The first model, do not allow cooperation to transport founded food, whereas the second, allow cooperation to transport it. In the two proposed models, we assume that agents have simple sensing capabilities (perceive the four neighboring cells) where no direct communication is used between agents, rather, agents use pheromones to communicate. The task is to search for food in a bounded 2D grid environment, and to return it back to the nest. Agents do not know the location of food a priori (scattered randomly), nor do they have GPS or odometry capabilities. We show that both the two models are able to construct shorter paths between food and nest in obstacle-free and obstacle environments than gradient-based methods [18] [21].

The remaining of this paper is organized as follows: The proposed models are described in Section 2 followed by simulation results and comparison in Section 3. Finally, Section 4 concludes the paper.

II. Definition of Proposed Models

A. Specifications of S-ACF no-coop and S-ACF coop Models

The two proposed extensions assume that agents are simple in nature and react only to stimulus that they receive. They are based on the foraging behavior of ants which deposit pheromones to mark visited cells and can sense the pheromones deposited by others. They use the principle of coloring cells (as static pheromone represented by a specific color) that allow to mark trails between founded food and the nest. Moreover, they use the move function of the S-MASA algorithm that guaranties to built shorter paths while exploring by means of pheromone concentration. An agent based modeling approach has been taken to model the behavior of ant colonies. These models implies the following key points .

- *Environment:* Is a 2D grid environment with N X N topology. Each grid can be an obstacle, food or nest, or even contain an agent (ant). Each grid is surrounded by four neighboring grids. The nest is the central place that stores collected food, it is located at the center of the environment.
- *Agent (Ant):* Simple reactive agents, with limited range sensor (can only perceive the four neighboring cells), had no memory and use the environment as their shared memory. Each agent has an initial position and a heading (0, 90, 180 or 270).
- *Pheromone:* Has a numerical meaning. It is represented by a color. Two kinds of pheromones have been used. The first can be evaporated with an evaporation rate, it is used to mark visited cells and to repulse agents from them. The second one, is used to mark trails between founded food and the nest, in order to keep track of founded food location and it had no evaporation properties, it is removed when the food is exhausted.
- *APF Values:* Integer values written by agents in the environment, to mark the short distance between any cell and the nest [18]. Note that, the APF values can be updated to get optimal ones by any agent that revisit a marked cell (visited cell with integer value). However, in our two models the pheromone concentration values can be used as APF values to build shorter paths.
- *S-MASA:* A stigmergy-based algorithm for multi-target search. In this algorithm we used the *MOVE function* that produces a vortex around the nest. This function is detailed in [19] and we refer to it as S-MASA algorithm in all the paper.
- *Food:* A certain amount of food is distributed randomly in the environment. The laden ant carry a limited amount of food (1 unit at each step) back to the nest until the food is exhausted.
- *Obstacle:* A certain amount of obstacles are also distributed in static positions in the environment.

B. Finite State Machine of S-ACF no-coop and S-ACF coop Models

The foraging behavior used is shown in Figure 1. The agents use a set of action sequences in order to carry out their task effectively. These action sequences are explained as follows:

At-Home: In this state, agents are unloading resources. They first test wether there is a trail or not. If there is no trail they'll turn into *Look-for-Food* state and if there is a trail and the amount of food is > 0, they'll *Climb* the trail back to the founded food, else they'll execute the *Remove-Trail* state.

Look-for-Food: Agents first, test wether they are at food. If they are, they turn into *Pick-Food* state. If there is no food, they move using rules of the *Choose-Next-Patch* state.

Choose-Next-Patch: This state allows the agent to decide where to move next, while creating a vortex around the nest and building at the same time the shorter paths. Agents can avoid obstacles by executing the avoid obstacle function. When the next step is taken, the agent turns automatically to the *Look-for-Food* state.

Pick-Food: Agent in this state picks a limited amount of it and looks for a trail, if there is one it executes a *Return-to-Nest* state; else it turns into *Return-and-Color* state.

Remove-Trail: If the food is exhausted, the agent moves to a colored cell with the greatest pheromone concentration and reset its color (yellow color) to the default one (black color) until there is no colored cell, it turns then into the *Look-for-Food* state.

Climb: Agents in this state move to a colored cell (part of trail) with pheromone concentration greater than theirs and check if there is a food (turn automatically into *Look-for-Food* state).

Return-to-Nest: If there exists a trail, the agent moves to one of colored neighboring cells (part of trail) with the lowest pheromone concentration. If nest is reached, it turns into *At-Home* state.

Return-and-Color: The agent moves to one of the four neighboring cells with the lowest pheromone concentration and changes its color from default one (scale of green color, which represents the evaporation of pheromone) to a trail color (yellow) until it reaches the nest; it turns then to the *At-Home* state.

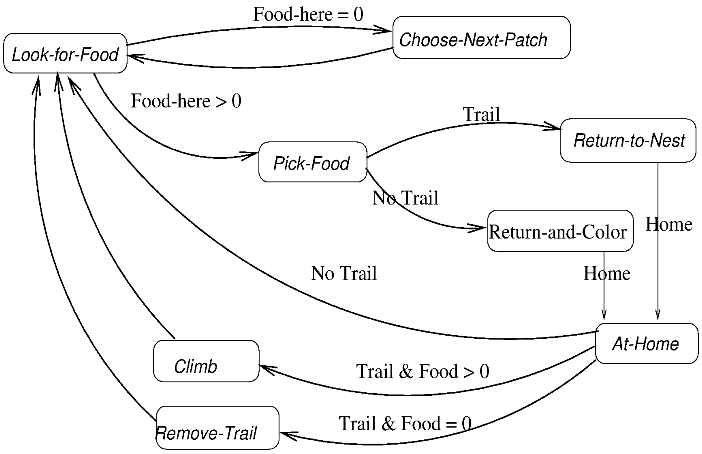


Fig. 1: Finite state machine of an autonomous foraging agent (S-ACF no-coop and S-ACF coop)

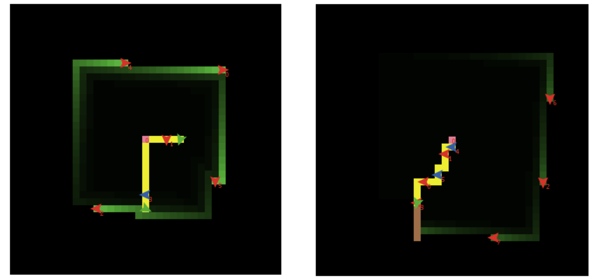
C. Differences between S-ACF no-coop and S-ACF coop

Both of the models use the finite state machine depicted by Figure 1. However, the set of action sequences in states: *Pick-Food*, *Choose-Next-Patch* and *Remove-Trail* are different from each other. S-ACF no-coop do not allow cooperation between agents and each founded food is exploited by its finder; whereas, the other agents proceed to their search process. The description of different states of the FSM in Figure 1(Section II-B), represent exactly a S-ACF no-coop (Figure 2(a)). However in S-ACF coop (Figure 2(b)), cooperation is allowed between agents and each founded food is exploited (transported) by all the agents informed by its location (stigmergic information via pheromones represented by brown color). The S-ACF coop model use the same FSM (Figure 1) and the same description of states in Section II-B unless for the states: *Pick-Food*, *Choose-Next-Patch* and *Remove-Trail*. They are defined as follows:

Pick-Food: When an agent finds a food, it picks a limited amount of it and diffuses the information to the neighboring cells by depositing diffusible pheromones (with brown color). It looks after that for a trail, if there is one it executes a *Return-to-Nest* state; else it turns into *Return-and-Color state*.

Choose-Next-Patch: The agent check if there is a pheromone in current cell, if yes it concludes that there is a food in neighboring cells. It diffuses also the information to its neighbors and follow pheromones to the food location; else it chooses to move according to the other rules of *Choose-Next-Patch* state and it turns automatically to *Look-for-Food* state.

Remove-Trail: When the food is exhausted, the agent moves to a colored cell with the greatest pheromone concentration and reset the color of visited cells to the default color (black) until no colored cell is founded (brown cells, those of diffusion), it turns then to the *Look-for-Food* state.



(a) (b)

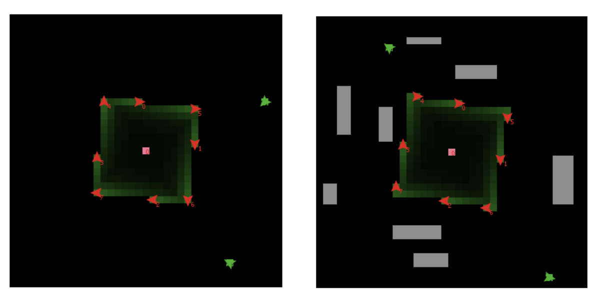
Fig. 2: The evolution of foraging achieved by (a) S-ACF no-coop (b) S-ACF coop. where Blue arrows represent laden agents and red ones represent searcher agents

III. Performance Evaluation

A. Simulation Parameters and Metrics

In this section, we discuss the performance and the comparison of four models (S-ACF no-ccop, S-ACF coop, C-marking no-coop and C-marking coop [18]) in obstacle-free and obstacle environments. The two world setups including positions of nest, food and agents, are given in Figure 3. There are several-related parameters which must be chosen, such as world size, food density, food concentration, agent's capacity and agent's number where: *Food Density* is the number of food locations (sites), each site contains a limited amount of food. These locations are distributed randomly in the environment. *Food concentration* Indicate the amount of food that every site contains (we refer to it as unit in the paper). *Agent's capacity* Is the amount of food (units) that an agent can transport at each time. The parameters of the three different simulations (scenario 1, scenario 2 and scenario 3) used, are given in Table I. To measure the performance of S-ACF no-coop and S-ACF coop, two metrics were used:

- *Total food returned* – The total amount of food that has been returned to the nest by all the agents after a given elapsed time steps.



(a) (b)

Fig. 3: World setups used in simulations (a) Obstacle-free environment (b) Obstacle environment

TABLE I: Parameters of scenario 1, scenario 2 and scenario 3

	Parameter	Value
Scenario 1		
	World size	40 X 40 cells
	Number of agents	1–50
	Food density	2 sites
	Food concentration	30 units
	Agent's capacity	1 unit
Scenario 2		
	World size	40 X 40 – 100 X 100
	Number of agents	20
	Food density	2 sites
	Food concentration	30 units
	Agent's capacity	1 unit
Scenario 3		
	Number of ticks	200–4000
	World size	40 X 40 cells
	Number of agents	20
	Food density	1 site
	Food concentration	60 units
	Agent's capacity	1 unit

- *Run time* – The finish time of the foraging mission. It is when all the food sites are discovered and exhausted. Run time is measured in steps or ticks.

Simulations have been carried out using agent-based modeling within Netlogo [22]. Netlogo is a multi-agent programmable modeling environment which allows to prototype quickly systems of situated agents evolving in a two dimensions world. The chose of the world could be carried out by users: the 2D environment can be either simulated as a grid or simulated as a continuous metric space. In this paper simulations are performed with a grid environment where cells can be either empty or occupied by food, robot, obstacle or the nest. We have considered two environment settings, first one, is obstacle-free and the second one is obstacle environment. The position of obstacles is fixed for all simulations in order to exclude its impact on the multi-agent systems performance. Agents have the same size as a cell they communicate by depositing pheromone in each visited cell and they start all from initial given positions and they have specific initial heading. Three type of scenarios have been used to test the performance of the models Table II. In scenario 1, we varied the number of agents from 1 to 50. In scenario 2, the size of the environment (world) was varied from 40 x 40 cells to 100 x 100 cells. In scenario 3, the number of agents is fixed to 20, and the environment size is fixed to 40 X 40 cells. At each time step during each run, we measured the total food that had been returned to nest. Each simulation was repeated 20 times in the three scenarios. The average reading was then calculated from the 20 trails for the three scenarios. The performance of the two proposed models as measured by these metrics, will be compared with each other and with C-marking coop [18]and with C-marking no-coop which is a non cooperative version of C-marking coop. In this last model, agents use pseudo random walk to search in their environment, they are able to build gradient while exploring. The APF values written by agents are not optimal and need a huge amount of time to its convergence to optimal values.

B. Performance of S-ACF no-coop and S-ACF coop over time and comparison with C-marking no-coop and C-marking coop

Here we show the results of S-ACF no-coop and S-ACF coop over time, operating in obstacle-free and obstacle environments and compare them with C-marking no-coop and C-marking coop. A key aim is to understand how efficient the two proposed models.

1) Results in Scenario 1: Results of simulations in scenario 1 demonstrate that the performance of S-ACF coop outperforms the three other models in obstacle-free and obstacle environments(Figure 4 and Table II). It costs less time to find the food, no time to find the close-to-shortest paths (pheromone concentration) and less time to transport food (because of the cooperation). S-ACF coop provides an important improvement in performances according to C-marking coop one, where 5 agents in S-ACF coop can achieve the foraging task in approximatively the same time needed by 30 agents in C-marking coop to achieve the same task in obstacle-free and obstacle environment. At each increase in the number of agents, the number of iterations required for foraging is decreased and results seems to be close to each other in the four models.

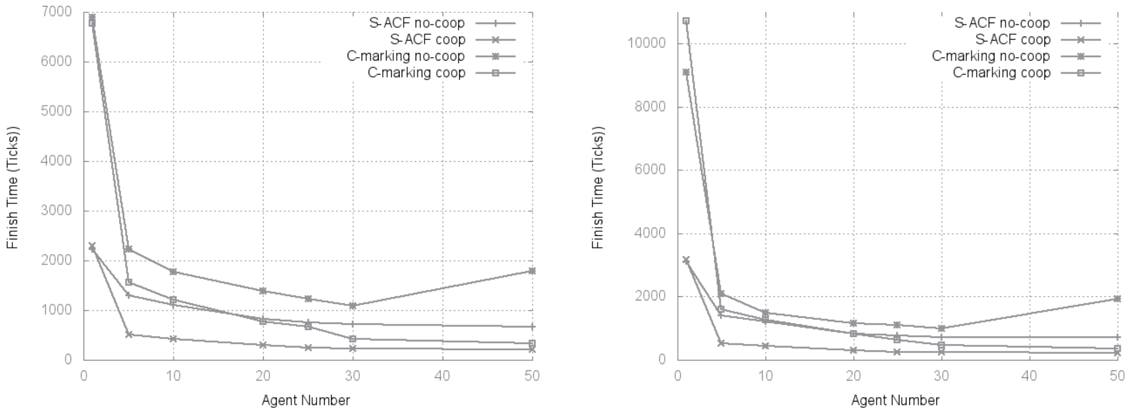
TABLE II: Effect of agent number in scenario 1

	1	**5**	**10**	**20**	**25**	**30**	**50**
Ticks in obstacle-free environment							
S-ACF no-coop	2236	1294	1111	822	752	714	658
S-ACF coop	2295	509	414	293	251	234	218
C-marking no-coop	6900	2236	1777	1393	1228	1096	1786
C-marking coop	6777	1566	1210	779	665	427	340
Ticks in obstacle environment							
S-ACF no-coop	3125	1403	1208	833	771	720	718
S-ACF coop	3179	527	429	307	259	237	231
C-marking no-coop	9093	2100	1490	1165	1107	985	1918
C-marking coop	10735	1603	1266	814	641	455	372

2) Results in Scenario 2: In scenario 2, we vary the size of the environment from 40 X 40 cells to 100 X 100 cells. S-ACF coop proves its superiority to the other three models in free-obstacle and obstacle environments (Table III and Figure 5). While S-ACF no-coop gives similar results to C-marking coop when the size of the environment is under 60 X 60 cells, it gives better results than it when the size is over 60 X 60 cells. However, C-marking no-coop gives the worst results in the two world setups.

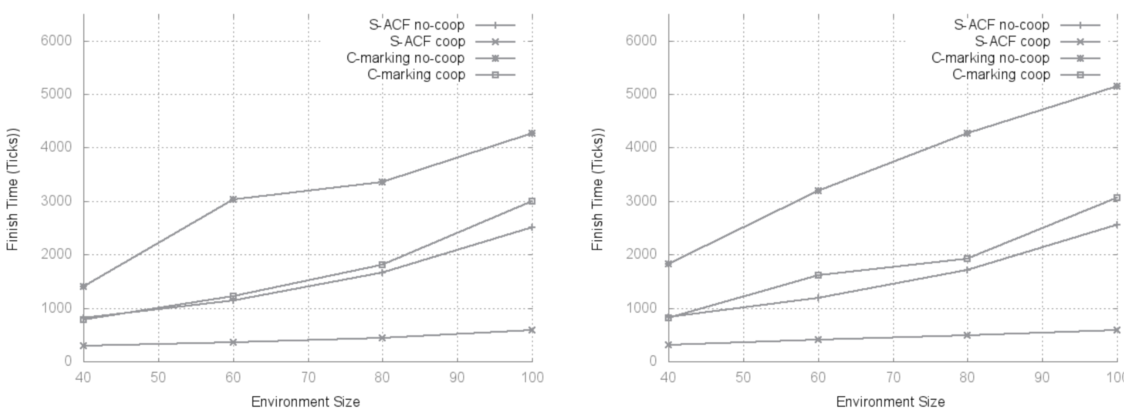
TABLE III: Effect of environment size in scenario 2

	40X40	**60X60**	**80X80**	**100X100**
Ticks in obstacle-free environment				
S-ACF no-coop	822	1144	1666	2515
S-ACF coop	293	360	444	594
C-marking no-coop	1393	3033	3355	4270
C-marking coop	779	1227	1813	3004
Ticks in obstacle environment				
S-ACF no-coop	833	1194	1716	2565
S-ACF coop	307	410	494	594
C-marking no-coop	1824	3198	4263	5155
C-marking coop	814	1613	1921	3057



(a) (b)

Fig. 4: Comparison results of S-ACF no-coop, S-ACF coop, C-marking no coop and C-marking coop , when varying the agent number (a) Obstacle-free environment (b) Obstacle environment



(a) (b)

Fig. 5: Comparison results of S-ACF no-coop, S-ACF coop, C-marking no-coop and C-marking coop, when varying the environment size (a) Obstacle-free environment (b) Obstacle environment

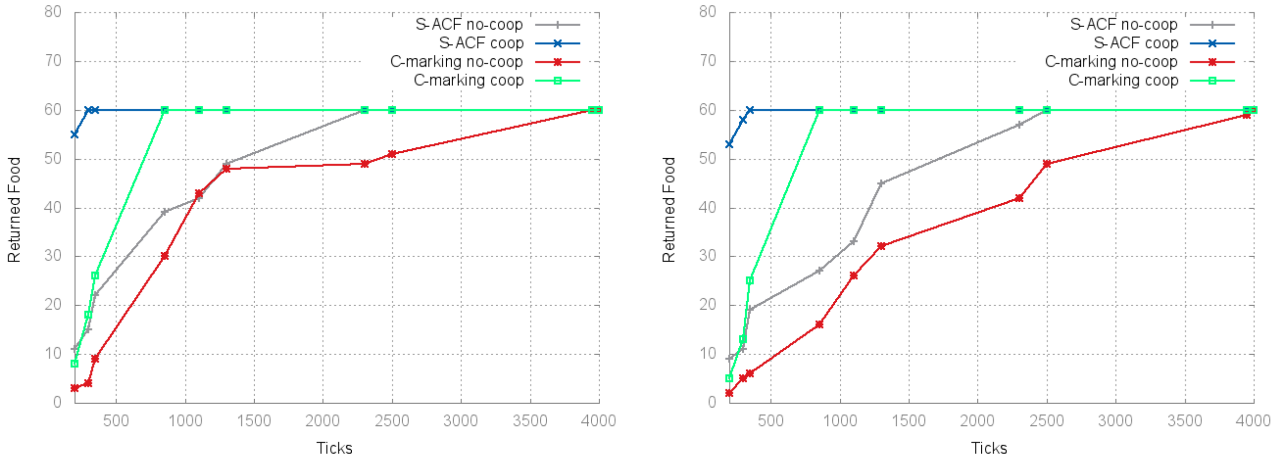
3) Results in Scenario 3: S-ACF no-coop and S-ACF coop give interesting results than C-marking no-coop and C-marking coop respectively, in terms of amount of food returns over 4000 tick (Table IV and Figure 6). S-ACF coop reaches and returns the total amount of food in only 300 ticks, which is less than half of the time needed by C-marking coop in obstacle-free and obstacle environment. However, S-ACF no-coop takes more time to exhaust the founded food (2300, 2500 ticks in obstacle-free and obstacle environment respectively), but it still more less than C-marking no-coop (3950, 4000 ticks obstacle-free and obstacle environment respectively) in the two environment settings. With the rapidity of Search and the close-to-shortest paths provided by S-MASA algorithm and the cooperation allowed in transporting food, S-ACF coop is superior to the other three models.

TABLE IV: Returned food over 4000 ticks (scenario 3)

	200	**300**	**350**	**850**	**1100**	**1300**	**2300**	**2500**	**3950**	**4000**
Obstacle-free environment										
S-ACF no-coop	11	15	22	39	42	49	60	60	60	60
S-ACF coop	55	60	60	60	60	60	60	60	60	60
C-marking no-coop	3	4	9	30	43	48	49	51	60	60
C-marking coop	8	18	26	60	60	60	60	60	60	60
Obstacle environment										
S-ACF no-coop	9	11	19	27	33	45	57	60	60	60
S-ACF coop	53	58	60	60	60	60	60	60	60	60
C-marking no-coop	2	5	6	16	26	32	42	49	59	60
C-marking coop	5	13	25	60	60	60	60	60	60	60

IV. Conclusion

In this paper, we presented two models (S-ACF no-coop and S-ACF coop) that extend the Army Raid Ant model by using S-MASA algorithm. Both of the models perform better than the C-marking no-coop and C-marking coop and S-ACF coop perform much better than the three other models. Using S-MASA algorithm provides both quick search and shorter paths, avoiding by the way the drawbacks of the pseudo



(a) (b)

Fig. 6: Comparison results of S-ACF no-coop, S-ACF coop, C-marking no-coop and C-marking coop, when varying the ticks number (a) Obstacle-free environment (b) Obstacle environment

random walk used by c-marking agents. The two proposed models give interesting results in obstacle-free and obstacle environments. In the future, we intend to study the parameters that can affect the performance of the proposed models other than agent number, environment size and amount of returned food.

References

[1] S. Momen, "Ant-inspired decentralized task allocation strategy in groups of mobile agents," *Elsevier Procedia Computer Science*, vol. 20, pp. 169–176, 2013.

[2] M. Waldrop and J. Gleick, "Complexity: the emerging science at the edge of order and chaos [m]," *info London: Viking*, 1992.

[3] M. Zhigang, Z. Beiji, and Z. Yu, "Considering direct interaction of artificial ant colony foraging simulation and animation," *Taylor & Francis Journal of Experimental & Theoretical Artificial Intelligence*, vol. 24, no. 1, pp. 95–107, 2012.

[4] R. C. Arkin, *Behavior-based robotics*. MIT press, 1998.

[5] C. Jones and M. Mataric, "Behavior-based coordination in multi-robot systems. autonomous mobile robots: Sensing, control," *Decision-Making Applications*, 2005.

[6] S. Nouyan, R. Groß, M. Bonani, F. Mondada, and M. Dorigo, "Teamwork in self-organized robot colonies," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 695–711.

[7] C. Parker and H. Zhang, "Collective unary decision-making by decentralized multiple-robot systems applied to the task-sequencing problem," *Springer Swarm Intelligence*, vol. 4, no. 3, pp. 199–220, 2010.

[8] B. Hölldobler, *The ants*. Harvard University Press, 1990.

[9] N. Hoff, R. Wood, and R. Nagpal, "Distributed colony-level algorithm switching for robot swarm foraging," in *Springer Distributed Autonomous Robotic Systems*, 2013, pp. 417–430.

[10] T. Sharpe and B. Webb, "Simulated and situated models of chemical trail following in ants," in *5th International Conference Simulation of Adaptive Behavior*, 1998, pp. 195–204.

[11] R. A. Russell, *Odour detection by mobile robots*. World Scientific, 1999, vol. 22.

[12] M. Mamei and F. Zambonelli, "Physical deployment of digital pheromones through rfid technology," in *Proceedings IEEE Swarm Intelligence Symposium, SIS 2005*, 2005, pp. 281–288.

[13] R. Vaughan, K. Støy, G. Sukhatme, and M. J. Matarić, "Blazing a trail: insect-inspired resource transportation by a robot team," in *Springer Distributed autonomous robotic systems 4*, 2000, pp. 111–120.

[14] R. T. Vaughan, K. Støy, G. S. Sukhatme, and M. J. Matarić, "Whistling in the dark: cooperative trail following in uncertain localization space," in *Proceedings of the fourth international conference on Autonomous agents*. ACM, 2000, pp. 187–194.

[15] R. T. Vaughan, K. Støy, G. Sukhatme, and M. J. Mataric, "Lost: Localization-space trails for robot teams," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 796–812, 2002.

[16] D. W. Payton, M. J. Daily, B. Hoff, M. D. Howard, and C. Lee, "Pheromone robotics," in *Intelligent Systems and Smart Manufacturing*, 2001, pp. 67–75.

[17] E. Barth, "A dynamic programming approach to robotic swarm navigation using relay markers," in *Proceedings of the 2003 American Control Conference*, vol. 6, 2003.

[18] O. Simonin, F. Charpillet, and E. Thierry, "Revisiting wavefront construction with collective agents: an approach to foraging," *Swarm Intelligence*, pp. 113–138, 2014.

[19] O. Zedadra, N. Jouandeau, H. Seridi, and G. Fortino, "S-MASA: A stigmergy based algorithm for multi-target search," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. P. M. Ganzha, L. Maciaszek, Ed., vol. 2. IEEE, 2014, pp. pages 1477–1485.

[20] R. V. Solé, E. Bonabeau, J. Delgado, P. Fernández, and J. Marín, "Pattern formation and optimization in army ant raids," *MIT Press Artificial life*, vol. 6, no. 3, pp. 219–226, 2000.

[21] G. Fortino, O. Zedadra, N. Jouandeau, and H. Seridi, "A decentralized ant colony foraging model using only stigmergic communication," in *Proceedings of XV Workshop Dagli Oggetti agli Agenti (WOA 2014)*, vol. 1260. CEUR, 2014.

[22] U. Wilensky, "Netlogo. http://ccl.northwestern.edu/netlogo/,," in *Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL*, 1999.