A Decentralized Ant Colony Foraging Model Using Only Stigmergic Communication

Giancarlo FORTINO
Università della Calabria, DIMES
Via P. Bucci, cubo 41c - 87036 - Rende (CS) - Italy
g.fortino@unical.it

Ouarda ZEDADRA
LabSTIC Laboratory, 8 may 1945 University, Guelma, Algeria
Department of computer science, Badji Mokhtar University, BP 12, Annaba, Algeria
Zedadra_nawel1@yahoo.fr

Nicolas JOUANDEAU
Advanced Computing laboratory of saint-Denis (LIASD)
Paris 8 University, Saint Denis 93526, France
n@ai.univ-paris8.fr

Hamid SERIDI
LabSTIC Laboratory, 8 may 1945 University, BP 401 Guelma, Algeria
seridihamid@yahoo.fr

***Abstract*— This paper addresses the problem of foraging by a coordinated team of robots. This coordination is achieved by markers deposited by robots. In this paper, we present a novel decentralized behavioral model for multi robot foraging named cooperative c-marking agent model. In such model, each robot makes a decision according to the affluence of resource locations, either to spread information on a large scale in order to attract more agents or the opposite. Simulation results show that the proposed model outperforms the well-known c-marking agent model.**

***Keywords*— Collaborative foraging; reactive coordination; digital pheromone; agent behavioral model; stigmergy.**

I. Introduction

Foraging is a benchmark problem for robotics, especially for multi-robot systems [1]. It is a “two-step repetitive process in which (1) robots search a designated region of space for certain objects, and (2) once found, these objects are brought to a goal region using some form of navigation” [2]. Distributed cooperative multi-robot systems are specifically adopted to achieve foraging missions when there is no a priori information about the environment, but communication mechanisms are needed for coordination. Pheromone deposits [3] is one of the approaches inspired from the study of the stigmergy process conducted in the early 90's on insect self-organized societies [4]. The foraging behavior of ants is an example of stigmergy where ants drop pheromones as they move in the environment. Most of studies in both artificial life and robotics carried out on synthetic pheromones use a large vocabularies linked to pheromone, coming from propagation and evaporation properties [5] [6]. These properties allow a group of agents to adapt to dynamic situations.

In this paper, we consider the problem of collective foraging in an unknown outdoor environment, with a homogeneous team of reactive agents that have no prior information about the environment. The objective is to retrieve and achieve all resource locations, while minimizing the time needed to complete the whole foraging. To this purpose, agents are based on a new behavioral model, where they can choose to deposit or not deposit diffusible pheromones regarding the quantity of resources in locations. Through simulation tests, the proposed system, that is an extension of that presented in [7], is compared with such previous system in terms of the number of iterations that are required to achieve the foraging task.

The rest of the paper is organized as follows. In Section II, we discuss related work. The pheromone, agent and environment models are given in Section III. A finite state machine based agent behavior for collective foraging and the corresponding algorithm are given in Section IV. Section V describes the simulation results, and a comparison between the original c-marking agents model [7] and our new model is also provided. Finally, in Section VI, conclusions are drawn.

II. Related work

A wide range of approaches has been adopted to suggest solutions to the foraging problem in unknown environments. Most of them focus on examples of multi-robot foraging from within the field of swarm robotics. The three main strategies for cooperation in this field are: information sharing [8], physical cooperation [9] [10] [11] [12] [13], and division of labor [14] [15] [16] [17] [18] [19] [20] [21]. Pheromone based techniques inspired from ants are useful for foraging with multiple robots [22] [23]. This approach has some drawbacks such as the computation of propagation and evaporation dynamics, and agents need specific mechanisms or materials that allow them to get back home. Authors is [24] and [6] propose the use of second pheromone diffusion from the base in order to avoid this last problem. At the same time, this solution can create new local minima.

An interesting approach named c-marking agents has been proposed in [7] that allow reactive agents to build optimal paths for foraging, which have limited information about their environment. To keep track of found resource locations and to build trails between them and the base, agents drop a quantity of pheromones inside their environment. A first extension of the c-marking agents model was proposed in [25], which gives interesting results regarding the number of agents and less interesting ones regarding the environment size. In this paper, we present a second extension of the c-marking agents model based on resources affluence and designed to change the behavior of robots to enhance results. Apart from enhancements related to environment adaptation, this new extension provides a more realistic model for the foraging problem.

III. Modeling System Components

The different components of our reactive multi-agent system are: Environment, Pheromone and Agent (or Robot) models.

A. Environment Model

The environment is modeled as a squared grid with variable size that has resources in multiple locations. These locations are scattered randomly and are unknown by the agents. Each location has a given quantity of resources. Cells in the environment can:

- Be an obstacle (grey color);
- Contain a resource (green color) of a limited quantity;
- Be the base station (red color), always positioned in the environment center, forming the starting point of all agents;
- Contain an agent (blue color).

B. Pheromone Model

The pheromone is modeled as a piece that can be spread to the four neighboring cells, if the quantity of resources in a location is more or equal to a maximum reference quantity QR_{max}; or it is modeled as a static piece that takes effect just in the current cell, if the quantity of resources in a location is less than a minimum reference quantity QR_{min}. Pheromones are directly managed by agents.

C. Agent Model

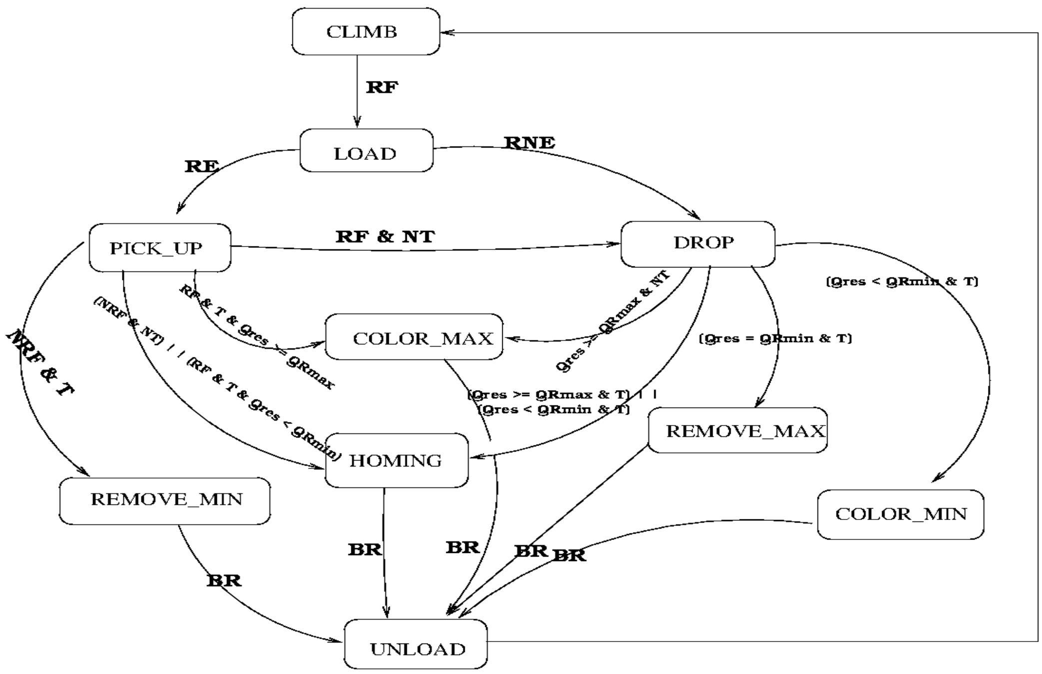
Agents have limited information about their environment. Due to the pheromone model, agents directly manipulate real pieces and are then close to real robots. At each time step (or iteration), each agent can:

- Move from a cell to another, which is not an obstacle in the four cardinal directions, like real robots.
- Perceive and read the values of the four neighboring cells. So agents can detect and load resources according to a maximum capacity Q_{max}.

Agents can read or write integer values that represent the Artificial Potential Field (APF) values [7], which represent the minimum distance between any cell and the base station cell. They are distributed to all agents, and can be modified to get the optimal values.

IV. Finite State Machine-based Agent Behavior for Collective Foraging

Figure 1 shows the finite state machine (FSM) diagram representing the behavior of an autonomous foraging robot (or agent). Such agent in its lifecycle goes through the following main states: 1) *CLIMB*; 2) *LOAD*; 3) *DROP*; 4) *PICK_UP*; 5) UNLOAD; and the following additional states: *COLOR_MAX*, *COLOR_MIN*, *REMOVE_MAX*, *HOMING*, and *REMOVE_MIN*. In all cases when the base station cell is reached, the agent executes the state *UNLOAD* and changes automatically to the *CLIMB* state when finished. The state details of the FSM, representing the proposed cooperative c-marking agents V2 model, are given below along with Algorithm 1 that provides further details.



Transitions: RF: Resource Found; RNE: Resource Not Exhausted; RE: Resource Exhausted; NT: No Trail exists; T: Trail exists; NRF : NoResource Found; Q_{res}: Quantity of resources; QR_{max}: Maximum amount of resources; QR_{min}: Minimum amount of resources; BR: Base Reached.

Fig. 1. Finite State Machine of the agent-based model of autonomous foraging robot

1) *CLIMB*: it is the initial state for all agents, in which the highest priority task for an agent is to exploit a resource when it is detected or to climb a trail, by choosing a colored cell with max value of APF or finally to execute an exploration & APF construction [7].

2) *LOAD*: the agent in this state picks up a Q_{max} of resource. If the resource is exhausted, the agent goes to *PICK_UP*; otherwise it goes to *DROP*.

3) *DROP*: it is a transitory state towards one of the following four states (transitions are labeled by guards detailed in Figure 1):

- *COLOR_MAX*: when the amount of resources is more than QR_{max}, agents drop diffusible pheromones; by such means, they create a max trail, within which colored cells with min values are not chosen in order to avoid common trails problem;
- *COLOR_MIN*: When the amount of resources is less than QR_{min}, agents drop non diffusible pheromones, so creating min trails. Colored cells with min values are not chosen in order to avoid common trails problem;
- *REMOVE_MAX*: if the amount of resources is equal to QR_{min} and there exists a max trail, agents remove such an amount;
- *HOMING*: if no trail exists and the resource is exhausted, agents just follow min values until the base is reached.

4) *PICK_UP*: it is a transitory state towards one of the following four states (transitions are labeled by guards detailed in Figure 1): *DROP*, *COLOR_MAX*, *HOMING*, and

- *REMOVE_MIN*: it consists in removing the min trail in order to avoid attraction of agents to an exhausted resource.

5) *UNLOAD*: when the agent reaches the base station cell, it drops all resources and changes immediately its state to *CLIMB*.

Algorithm 1: *Cooperative c-marking agents V2*

CLIMB
IF (Resource Found) **goto** *LOAD*
ELSEIF (Trail Exists) Move to cell with highest value
ELSE do *exploration*

LOAD
Pick up Q_{max}
IF (Resource Not Exhausted) **goto** *DROP*
ELSE goto *PICK_UP*

DROP
IF (Q_{res}> = QR_{max} & No Trail exists) **goto** *COLOR_MAX*
ELSIF (Q_{res}= QR_{min} & Trail exists) **goto** *REMOVE_MAX*
ELSIF (Q_{res}< QR_{min} & No Trail exists) **goto** *COLOR_MIN*
ELSE goto *HOMING*

PICK_UP
IF (Resource Found & No Trail exists) **goto** *DROP*
ELSIF (No Resource Found & Trail exists)
goto *REMOVE_MIN*
ELSE goto *HOMING*

COLOR_MAX
IF (Base Reached) **goto** *UNLOAD*
ELSE
- Move to a new neighboring, not colored cell with the least value;
- Color the current cell with dark gray color and the 4 neighboring cells with light gray color.

COLOR_MIN
IF (Base Reached) **goto** *UNLOAD*
ELSE
- Move to a new neighboring, not colored cell with the least value;
- Color the current cell with dark gray color

REMOVE_MAX
IF (Base Reached) **goto** *UNLOAD*
ELSE
- Move to a new neighboring colored cell with the least value;
- Reset the color of the 4 neighboring cells to the default color (white color);

REMOVE_MIN
IF (Base Reached) **goto** *UNLOAD*
ELSE IF (one colored cell exists in neighboring)
- Move to min colored cell in neighboring
- Reset the color to the default color (white color).

HOMING
IF (Base Reached) **goto** *UNLOAD*

ELSEIF (Trail Exists)
- Move to min colored cell in neighboring
- **do** *update-value*

ELSE
- Move to min valued cell in neighboring
- **do** *update-value*

UNLOAD
Depose resources
goto *CLIMB*

exploration:
IF (There exists a neighboring cell without value)
- Move randomly to such cell
- **do** *update-value*

ELSE
- Move randomly to a free cell
- **do** *update-value*

update-value:
Write val = min (val, 1+ min (4 neighbor values)) in current cell

It is worth noting that the states *CLIMB*, *HOMING* and *UNLOAD* are the same as in [7]. Finally, the ELSE clause in the states *COLOR_MAX*, *COLOR_MIN*, *REMOVE_MAX* and *REMOVE_MIN* implies the return into the same state.

V. Simulation Results and Comparison

Two simulation scenarios have been defined by using the JADE framework [26] to evaluate the proposed model. In the first one, we test the influence of the agents' number on the system performance by varying the number from 5 to 160; whereas in the second one, we test the influence of the environment size on the system performance by changing the size from 12X12 to 100X100. The foraging time is defined as the number of iterations required for discovering and exhausting all the resources in the environment.

Scenario 1: The environment is composed of 40X40 cells with 30% obstacles; 20 cells are resources locations; each resource contains 1000 units of resources and each robot can load a maximum of 100 units. The number of robots is varying between 5-160 agents.

TABLE I. EFFECT OF AGENT'S NUMBER ON PERFORMANCE

Iterations	5	10	20	40	80
C-marking agents	19200	8697	4114	2263	1070
Cooperative c-marking agents V1	10476	6917	3403	1125	609
Cooperative c-marking agents V2	10255	6500	3200	950	510

Table I shows the simulation results of scenario 1; where the increase in the number of agents provides a decrease in the foraging time. Even if cooperative c-marking agents V1 gave interesting results than c-marking agents one [25], cooperative c-marking agents V2 gives interesting results than the two previous models, where the number of iterations is reduced but with a less degree regarding cooperative c-marking agents V1. Foraging time still considerable when number of agents is 5 to 20; it is fast when number of agents is 40 to 80. Avoiding the creation of common trails has contributed to the reduction of the foraging time.

Scenario 2: The environment contains 5% obstacles; 20 cells are resource locations; each resource contains 2000 units of resources and the number of robots is 50. Each robot can carry a maximum of 100 units. The environment size varied from 12X12 to 100X100.

Table II show the simulation results of scenario 2; where the foraging time increases less by increasing the size of the environment, until 100X100, the foraging time increases dramatically.

TABLE II. EFFECT OF ENVIRONMENT SIZE ON PERFORMANCE

Environment size / Iterations	**12X12**	**25X25**	**50X50**	**100X100**
Cooperative c-marking agents V1	192	652	1395	10777
C-marking agents	155.5	345	805	2290
Cooperative c-marking agents V2	150	315	630	1250

We compared the proposed behavioral model (cooperative c-marking agent model V2) to the original c-marking agent model As one can see in Figure 2, increasing the level of cooperation between agents by the spread of a diffusible pheromones, allows agents to spend more time in exploitation rather than exploration, which means that resources will be exhausted rapidly and agents can spread out to exploration. When the quantity is less important, agents spread a non-diffusible pheromone that means that they did not need cooperation.

The preliminary results of scenario 2 in our previous work (cooperative c-marking agents V1 model) [25] are less important than the c-marking model [7] because of the common trails problem. When agents return home and color min or max trails, there is a possibility that they meet existing trails and they use them as part of their trail. As a result, they got a common part for the two trails to different resources. If one of the two resources is exhausted, agents proceed to the REMOVE_MIN state that will remove the common part, even if the second resource is not exhausted yet. When agents included in the second trail execute the HOMING state, they will look for the rest of the trail, which is removed, and they will get stuck in that common part. When agents execute COLOR_MAX or COLOR_MIN states, they must avoid the colored cells, which mean that they avoid creating common parts with existing trails. Such two states enhanced in this paper, have contributed to improve results of scenario 2 (shown by table II). Figure 3 shows a comparison with the original c-marking agents model and the previous cooperative c-marking agents model, regarding scenario 2.

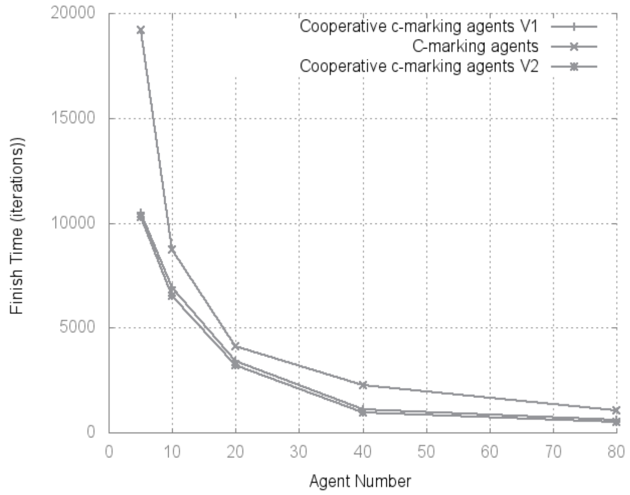


Fig. 2. Results comparison with c-marking agent model and cooperative c-marking model V1, regarding scenario 1

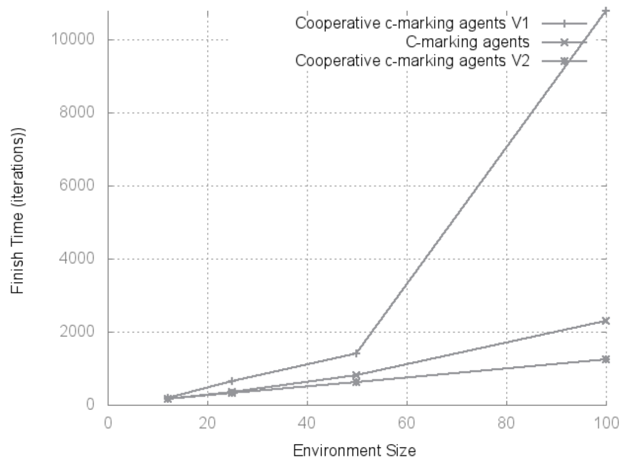


Fig. 3. Results comparison with c-marking agent model and cooperative c-marking model V1, regarding scenario 2

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new behavioral model for the foraging problem that aims to decrease the foraging time regarding the quantity of resources in locations. The new behavioral model based on resource affluence gives interesting results with respect to the original model (c-marking agent model). Agents in our system can perceive the environment, pick up resources, transport them to a storage point and manage the pheromone as a real piece, thus they are close to real robots. In perspective, we think that robot's behavior can be enhanced by introducing both new exploration approaches and solutions to problems such as the fast convergence of the Artificial Potential Field.

REFERENCES

[1] A. Winfield, “Foraging robots”. Encyclopedia of Complexity and Systems Science. Springer, pages 3682-3700, 2009.

[2] E. H. Ostergaard, G. S. Sukhatme, and M. J. Mataric. “Emergent bucket brigading: A simple mechanism for improving performance in multi robot constrained-space foraging tasks”. In Int. Conf. On Autonomous Agents, 2001.

[3] B. Hrolenok, S. Luke, K. Sullivan, and C. Vo. “Collaborative foraging using beacons”. In AAMAS'10. IFAAMAS, pages 1197-1204, 2010.

[4] E. Bounabeau and G. Theraulaz, “Intelligence Collective”. Hermes edition, 1994.

[5] M. Nakamura and K. Kurumatani. “Formation mechanism of pheromone pattern and control of foraging behavior in an ant colony model”, 1997.

[6] M. Resnick, “Turtles, termites, and traffic jams: Explorations in massively parallel microworlds”. Cambridge, MIT Press, 1994.

[7] O. Simonin, F. Charpillet, and E. Thierry, “Collective construction of numerical potential fields for the foraging problem”. Technical report, INRIA, France, 2007.

[8] M. J. Mataric and M. J. Marjanovic. “Synthesizing complex behaviors by composing simple primitives”. In Self-Organization and Life, From Simple Rules to Global Complexity, European Conference on Artificial Life (ECAL-93)}, pages 698-707, 1993.

[9] A. J. Ijspeert, A. Martinoli, A. Billard, and L. M. Gambardella. “Collaboration through the exploration of local interactions in autonomous collective robotics: The stick pulling experiment. Autonoumous robots”, 2001.

[10] K. Easton A. Martinoli and W. Agassounon. “Modeling swarm robotic systems: A case study in collaborative distributed manipulation”. Journal of Robotics research, Special Issue on Experimental Robotics, 2004.

[11] L. E. Paker, “Alliance: an architecture for fault tolerant, cooperative control of heterogeneous mobile robots”, In IEEE/ RSJ International Conference on Intelligent Robots and Systems, 1994.

[12] R. Grob, E. Tuci, M. Dorigo, M. Bonani, and F. Mondada, “Object transport by modular robots that self-assemble”. In IEEE International Conference on Robotics and Automation, 2006.

[13] M. Dorigo, E.Tuci, R.Grob, V.Trianni, T.H. Labella, S.Nouyan, and C. Ampatzis, “The swarm-bot project”. In Swarm Robotics Workshop: State-of-the-art survey, number 3342 in Lecture Notes in Computer Science, Berlin Heidelberg, Springer Verlag, 2005.

[14] T. Balch and R. C. Arkin, “Communication in reactive multiagent robotic systems”. Autonomous robots, 1994.

[15] D. Goldberg and M. J. Mataric, “Interference as a tool for designing and evaluating multi robot controllers”. In 14th National Conference on Artificial intelligence (AAAI-97). MIT Press, 1997.

[16] K. Lerman, “Mathematical model of foraging in a group of robots inspired by ants foraging behavior”. In ACM Transactions on Autonomous and Adaptive Systems, 2006.

[17] E. Bonabeau, C. Theraulaz, and J. L. Deneubourg, “Quantitative study of the fixed threshold model for the regulation of division of labor in insect societies”. In the RoyalSociety of London, Series B Biological Sciences, 1996.

[18] E. Bonabeau, G. Theraulaz, and J.L. Deneubourg, “Fixed response thresholds and the regulation of division of labour in insect societies”. In Bulletin of Mathematical Biology, 1998.

[19] M. Krieger and J.B. Billeter, “The call of duty: Self-organised task allocation in a population of up to twelve mobile robots”. In Jour of Robotics and Autonomous Systems, 2000.

[20] J. L. Deneubourg, S. Goss, J. M. Pasteels, D. Fresneau, and J. P. Lachaud, “Self-organization mechanisms in ant societies (ii): learning in foraging and division of labour”. Experientia Suppl, 1987.

[21] T. H. Labella, M. Dorigo, and J. L. Deneubourg, “Division of labour in a group of robots inspired by ants foraging behaviour”. ACM Transactions on Autonomous and Adaptive Systems, 2006.

[22] J. L. Deneubourg, S. Aron, S. Goss, J. M. Pasteels, and G. Duerrinck, “Random behavior, amplification processes and number of participants: How they contribute to the foraging properties of ants”. Physica, 1986.

[23] B. Holldobler and E.Wilson, “The Ants”. 1990.

[24] L. Panait and S. Luke, “A pheromone-based utility model for collaborative foraging”. In AAMAS'04. ACM, pages 36-43, 2004.

[25] O. Zedadra, N. Jouandeau, H.Seridi, “Cooperative c-marking agents for the foraging problem”. In the Fourth International Conference on Advances in System Simulation, IARIA, 2012.

[26] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa, "JADE: A software framework for developing multi-agent applications. Lessons learned," Information & Software Technology 50(1-2): 10-21, 2008.