Stigmergic MASA: A Stigmergy Based Algorithm for Multi-Target Search

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

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

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

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

***Abstract*—We explore the on-line problem of coverage where multiple agents have to find a target whose position is unknown, and without a prior global information about the environment. In this paper a novel algorithm for multi-target search is described, it is inspired from water vortex dynamics and based on the principle of pheromone-based communication. According to this algorithm, called Stigmergic MASA (for "Multi Ant Search Area"), the agents search nearby their base incrementally using turns around their center and around each other, until the target is found, with only a group of simple distributed cooperative Ant like agents, which communicate indirectly via depositing/detecting markers. This work improves the search performance in comparison with pure random walks, we show the obtained results using computer simulations.**

I. Introduction

The problem of finding multiple targets whose positions are unknown without a prior information about the environment is very important in many real world applications [1]. Those applications vary from mine detecting [2] [3], search in damaged buildings [4] [5], fire fighting [6], and exploration of spaces [7] [8], where neither a map, nor a Global Positioning System (GPS) are available [9]. The random walk is the best option when there is some degree of uncertainty in the environment and a reduced perceptual capabilities [10] because it is simple, needs no memory and self-stabilizes. However, it is inefficient in a two-dimensional infinite grid, where it results in an infinite searching time, even if the target is nearby [11], it results also in energy consumption and malfunction risks. To deal with these limits, some effective ways to coordinate the multi agent system need to take place. Recently many researchers have investigated bio-inspired coordination methods [12] [13], in which agents coordinate on the basis of indirect communication principle known as stigmergy.

Approaches that treat multi-target search are of a degree of computational complexity and with idealized assumptions, such as: perfect sensors [14], stationary environments [15], unlimited direct communication [16]. These assumptions make them unrealistic in real world applications. The algorithm presented in this paper avoids such type of assumptions. It makes the following contributions: (1) it is of very low computational complexity, in which agents have a very low-range of sensors; (2) it executes a search in nearby locations first by adopting spiral turns around the starting cell and between agents; (3) agents use stigmergic communication via digital pheromone; (4) turns on known or unknown static obstacle-free environments or obstacle environments.

The rest of this paper is organized as follows. Section 2 discusses some related work. Section 3 describes the problem statement and formulation. Stigmergic MASA algorithm is described in detail in Section 4. Simulation results are shown in Section 5. A comparison with the random walk is given in Section 6 and Section 7 concludes the paper.

II. Related Work

The problem of searching a target may be considered as a partial area coverage problem that constitutes a key element of the general exploration problem [17] where coverage can be done by a single or multiple robots, with on-line or off-line algorithms. In the on-line coverage algorithms, the area and target positions are unknown, and are discovered step by step while the robot explores the environment, whereas, in the off-line algorithms, the robot has a prior information about the environment, target and obstacles positions, so it can plan the path to go through. Different approaches have been developed in the literature to solve area coverage using single or multiple robots. In this section, a brief overview of techniques that are used to solve the coverage problem using both single and multiple robots is presented. The single robot covering problem was explored by Gabriely and Rimon [18]. One of the most popular algorithms is the Spanning Tree Coverage one

(STC). In an STC algorithm, the robot operates in a 2D grid of large square cells. It aims to find a spanning tree for the graph described above, and allow the robot to circumnavigate it. This algorithm covers every cell that is accessible from the starting point s, and it is optimal because the robot passes through each cell at least once [19]. Spiral STC is an online sensor based algorithm for covering planar areas by a square shaped tool attached to a mobile robot. The algorithm incrementally subdivides the planar work into disjoint D size cells, while following a spanning tree of the resulting grid. The spiral STC covers every subcell accessible from the starting point, and covers these subcells in O(n) time using O(n) memory [20]. In this new version of STC, the spanning tree is stored in the onboard memory, which results in a dependency of the search area on memory size. With the aim of resolving the memory problem, Gabriely and Rimon propose in [21] the ant-like STC which forms the third version of the basic STC algorithm, that uses markers on visited cells. D-STC is introduced in [21] to solve the problem of uncovered partially occupied 2D-size cells, by visiting the previously uncovered cells, which results in worst-case scenarios, a twice coverage of the environment area. A generalization of STC to multi-robots is given in [22], the MSTC, in which a spanning tree is computed, and then it is circumnavigated by each robot. Another spanning tree construction using multiple robots based on approximate cellular decomposition is proposed in [23]. Another approach developed in [1], where the environment is subdivided into n concentric discs, each disc is covered by one robot, when the entire disc is completely covered, the robot moves to the next disc not yet covered; an extension of this algorithm that uses heterogenous robots is given in [17]. Instead of concentring on the robot's on-board resources, some part of robotics literature use a single ant or a group of ants robots to cover an area robustly, even if they do not have any memory, do not know the terrain, can't maintain maps of the terrain, nor plan complete paths. They use environmental markers such as pebbles [24], [25], [26] or pheromone like traces [27] or use greedy navigation strategies [28].

III. Problem Statement and Formulation

In a collective multi-target search task, there are a lot of targets randomly distributed in an area. The agents (robots) should find as fast as possible the targets and, after that, remove them, if we deal with a cleanup task, or transport them to a nest, if we deal with a foraging task. In this paper, a new search algorithm is proposed that enables a group of agents, each with limited perception capabilities to search quickly the targets. The algorithm presented here uses the principle of pheromone-based coordination where each agent deposits pheromone on its environment to inform the others about already visited areas. The finish time of the collective search is when all targets have been found. This section defines and clarifies some key terms which will be used in this paper.

- *Environment:* we assume that agents move in an N X M grid-based environment. It is divided into N X M cell. Each cell can be an obstacle, target or the base station, and can also contain an agent.
- *Agent:* simple reactive agents, with limited range sensor (can only perceive the four neighboring cells), have no memory and use the environment as their shared memory. Each agent has an initial position and heading (0, 90, 180 or 270).
- *Pheromone:* has a numerical meaning. It is represented by a color. The intensity of the pheromone at time t is set to arbitrarily chosen value c which is a small positive constant. It evaporates with time with a coefficient p fixed to 0.075 using equation 1to avoid accumulation of pheromone.
- *Motion policy:* each agent chooses the next cell to visit using a motion policy that is function of the presence of pheromone trail and obstacles. This policy helps the agent to decide where to go next.

IV. Design of the Stigmergic MASA Algorithm

The idea behind proposing this algorithm is to reproduce the behavior observed in water vortex dynamics. The vortex is a region in which a fluid flow is mainly a rotary movement about an axis, rectilinear or curved. So each agent tries to turn around the base station and around the other agents. Doing this with agents only is difficult and needs a great number of agents, but using pheromone to repulse agents from visited cells was very helpful to reproduce the structure of a vortex.

A. Basic Stigmergic MASA

In Stigmergic MASA, each agent started from an initial given position and oriented toward a given heading. To turn around the base station and around each other, each agent checks on his right cell if it is visited or not. If it detects a pheromone (Figure 1), it indicates to the agent that it is about to enter to a visited cell and therefore the agent keeps going forward its current heading, else the agent changes its heading and moves toward a new heading.

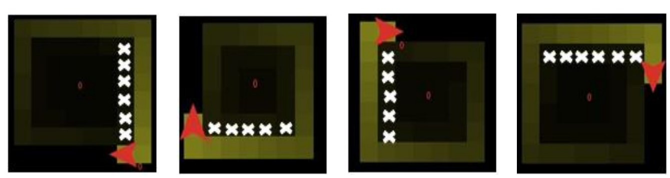


Fig. 1. Stigmergic MASA coordination principle: (a) Changing heading from 180 to 270 (b) Changing heading from 270 to 0 (c) Changing heading from 0 to 90 (d) Changing heading from 90 to 180

Stigmergic MASA is further detailed in Algorithm 1

Algorithm 1 Stigmergic MASA

Input: position and heading for each agent,
Output: iteration number,

while number of targets and boundaries are not reached **do**
    Move
    Lay pheromone
    Update Pheromone
end while

Move function is the motion policy. Each agent has initially a given heading (0, 90, 180 or 270) that allows it to move up, right, down or left in the four neighboring cells. The agent checks always its right cell which is the up cell if the heading is 270, the down cell if the heading is 90, if no pheromone is there it can change his heading to the new one using the move function and goes forward in that new heading. The move function is detailed in Algorithm 2

Algorithm 2 Function Move

if (pheromone is detected in right cell) **then**
    go forward
else if (heading = 270) **then**
    set heading to 0
else
    set heading to heading + 90
end if

Update pheromone function is used for pheromone evaporation, using the equation

$$\Gamma_i(t+1) = \Gamma_i(t) - p * \Gamma_i(t) \quad (1)$$

Where: p is a coefficient which represents the evaporation of trail between time t and (t+1) is set to 0.075 to avoid unlimited accumulation of pheromone. Stigmergic MASA can be applied to environment with or without obstacles, the agent executes the function avoid obstacle to avoid obstacles, where the agent follows in this case the obstacle boundary until a not visited cell is encountered, which means that agents are going around the obstacle in the direction of visited cells to guarantee the completeness of the algorithm.

B. Stigmergic MASA Extensions

The proposed algorithm allow to cover gradually the environment starting from the base station and reproducing by the way principle of central place foraging theory [29]. Although, this algorithm generates very efficient search results based on relatively simple motion rules, it can be extended to deal with dynamically changing environments, and to deal with coverage problem in known or unknown environments.

V. Performance Evaluation

We used Netlogo framework [30] to evaluate the performance of our algorithm in two scenarios. In the first scenario

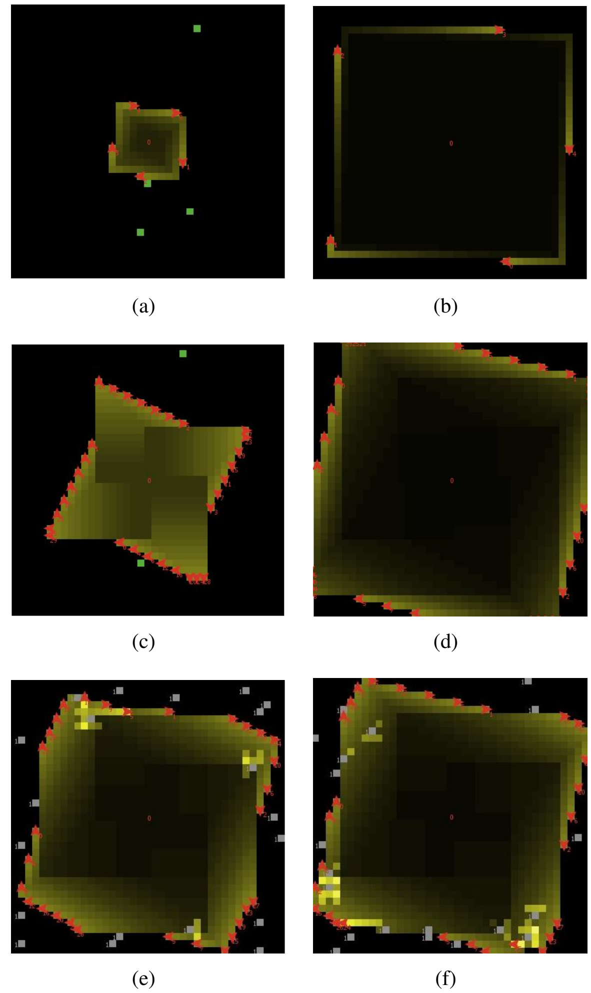


Fig. 2. The evolution of search achieved by Stigmergic MASA: (a), (b) Initial and final position of the 5-agents group in an obstacle-free environment. (c), (d) Initial and final position of the 30-agents group in an obstacle-free environment. (e), (f) Initial and final position of the 30-agents group in an obstacle environment

we evaluate the algorithm by varying the number of agents from 5 agents to 30 agent in two environment configurations: obstacle-free environment and obstacle environment. In the second scenario, we evaluate the algorithm by varying the size of the environment from 20 X 20 cell to 100 X 100 cell, in two environment configurations: obstacle-free environment and obstacle environment. Obstacles in the two scenarios were defined in two ways: (i) given a desired percentage, cells were randomly designated as obstacles (ii) obstacles were specifically designed by hand. Then, one possible extension on Stigmergic MASA is discussed and related simulation results are illustrated. To evaluate average performance, each simulation is repeated 20 times, where time is defined as the number of iterations required by the agents to discover all the targets.

A. Scenario 1: Influence of Number of Agents on Performance

Agents start all from the base station which is situated at the center of the environment and each agent has a heading,

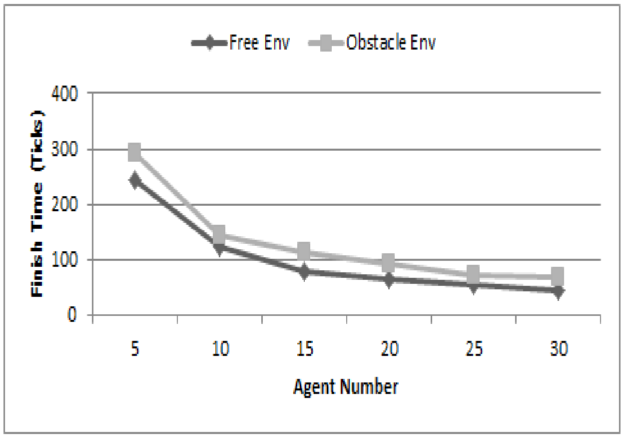


Fig. 3. Effect of number of agent on performance in obstacle-free and obstacle environment

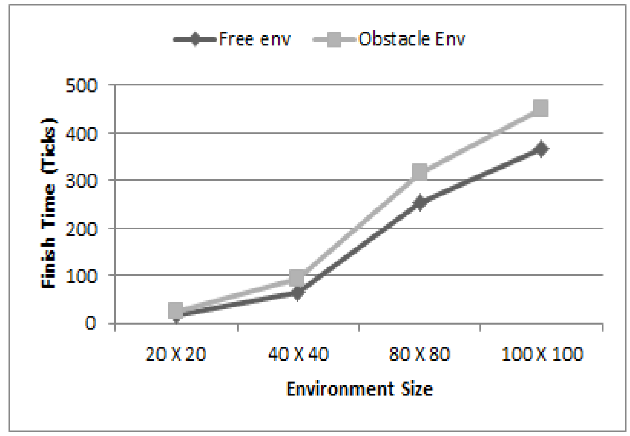


Fig. 4. Effect of environment size on performance in obstacle-free and obstacle environment

we vary the number of agents from 5 to 30. The environment consists of a square of size 40 X 40 cell shown in Figure 2, free or with obstacles, with four targets distributed randomly. An example of execution of Stigmergic MASA on a group of 5 agents, 30 agent on obstacle-free environment and a group of 30 agent on obstacle environment are illustrated in Figure 2.

Table I shows the performance of the algorithm in scenario 1 while the number of agents is varying from 5 to 30. It is represented graphically in Figure 3. The search time becomes dramatically faster with an increase in the number of agents. Note that there is no direct communication between agents, the one communication tool is the pheromone deposited in the environment. The standard deviation of the number of iterations reflects the impact of the random distribution of the targets between simulations. There is a linear decrease in the iterations number.

TABLE I
EFFECT OF AGENT'S NUMBER ON PERFORMANCE

	5	10	15	20	25	30
Iterations in free env	242,85	122,2	78,85	63,5	54,8	43,9
STD Deviation	46,62	24,84	17,87	14,15	11,35	10,15
Iterations in obstacle env	289,85	143,35	114,15	93,55	71,8	69,55
STD Deviation	56,76	36,93	18,29	18,58	22,97	22,24

B. Scenario 2: Influence of Environment Size on Performance

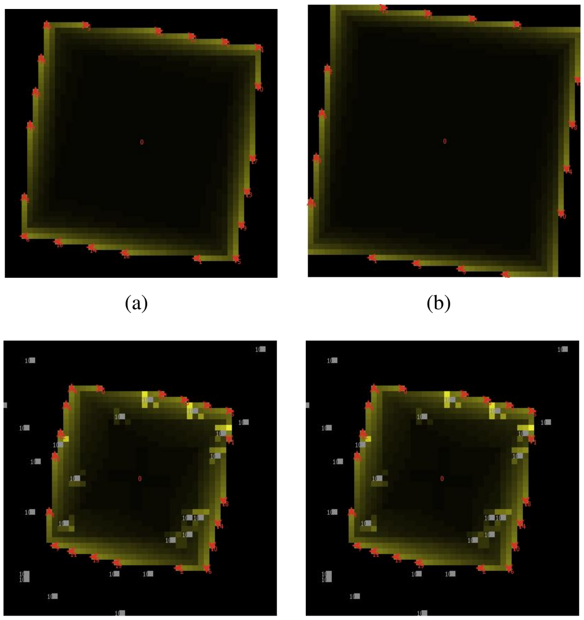
We now show how the size of the environment affects the performance of the algorithm when the number of agents is set to 20. Also here we used an obstacle-free environment and an obstacle environment, just varying the size of the environment from 20 X 20 cells to 100 X 100 cells. Table II shows the performance of the algorithm in scenario 2. It is represented graphically in Figure 4. The search time increases by increasing the size of the environment which is evident because the number of cells increases. The results show a difference in iterations number, Stigmergic MASA is robust to obstacles but this increase in number of iterations is due principally to the avoidance of obstacles that takes at least four iterations more, to go around a simple obstacle.

TABLE II
EFFECT OF ENVIRONMENT SIZE ON PERFORMANCE

	20X20	40X40	80X80	100X100
Iterations on Free env	16,2	63,4	254,8	366,15
STD Deviation	3,28	13,48	50,47	101,03
Iterations on Obstacle env	24,5	92,2	315,3	449
STD Deviation	8,06	23,37	71,49	131,06

C. Extension 1: Stigmergic MASA for Coverage Problem

Simulations presented in this section show that by changing the finish condition of the algorithm, the agents can achieve coverage mission as well as search one. The Stigmergic MASA algorithm can be applied for instance to known or unknown static environments, free or obstacle environments. Each simulation is repeated for 20 times in obstacle environments, because the obstacles are disseminated randomly in the environment and according to their position the agent take more or less iterations to go around the obstacle. Figure 5 represents the two simulations in obstacle-free and obstacle environment. As in scenario 1 and scenario 2, we test the performance of the algorithm on coverage problem by varying the number of agents and by varying the size of the environment in the two types of environments. Table III and Figure 6 show the obtained results when varying the number of agents. There is a linear decrease in number of iterations when increasing the number of agents, and there is a difference between iterations in obstacle-free environment and obstacle environment. If we compare these results to those of scenario 1, we can say they are close. A possible reason is the random distribution of targets, so if there is one target close to boundaries, the search



(a) (b)

(c) (d)

Fig. 5. The evolution of coverage achieved by Stigmergic MASA: (a), (b) 20-agents group in an obstacle-free environment in iterations 78 and 101. (c), (d) 20-agent group in an obstacle environment in iterations 44 and 108.

will be very close to coverage task and in the two tasks the number of iterations will be very close. Table IV and Figure 7 show the obtained results when varying the environment size, because here there is no random distribution of targets or there are no targets, the coverage time in obstacle environment is greater than the coverage time in obstacle-free environment, but there is always an increase in the number of iterations in the two cases of simulations.

TABLE III
EFFECT OF NUMBER OF AGENT ON PERFORMANCE

	5	**10**	**15**	**20**	**25**	**30**
Iterations in free env	320	171	120	89	80	68
Iterations in obstacle env	354,25	206,7	164,3	138,6	126,25	111,55

TABLE IV
EFFECT OF ENVIRONMENT SIZE ON PERFORMANCE

	20X20	**40X40**	**80X80**	**100X100**
Iterations on Free env	23	89	341	527
Iterations on Obstacle env	55,4	135,7	435,9	625,1

D. Comparison results

In Figure 8, we compare our algorithm with the random walk one when varying the number of agents from 5 to 30. This last method lets agents revisit visited cells which causes too much repeated search so global finish time increases. There is a difference in iteration number between results given by our algorithm and those given by random walk one. Our algorithm performs much better in obstacle-free environments than the random walk one that takes a huge number of iterations at least 1000 iterations when number of agents is less than 15. In obstacle environment our algorithm performs better than the random one too, when number of agents is less than 15; if the number of agents is equal or greater than 15 the random walk gives a very close results to our algorithm and the reason for that is the random walk of agents. Figure 9 presents a comparison when varying the environment size. Results are very different. Our algorithm gives the best results and random walk operates in a very slow manner when the environment size increases, even if targets are very close to the base station.

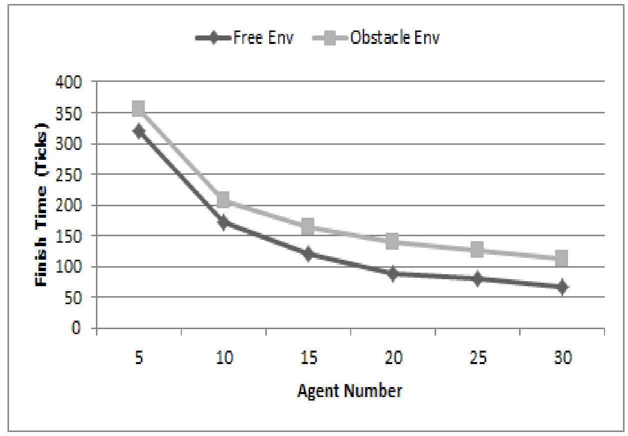


Fig. 6. Finish time of coverage in free-obstacle and obstacle environment with varying the number of agents

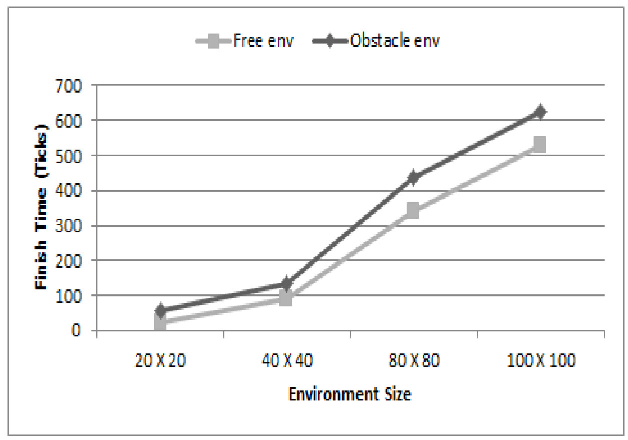


Fig. 7. Finish time of coverage in free-obstacle and obstacle environment varying the environment size

VI. CONCLUSION

A multi-target search algorithm called Stigmergic MASA is presented in this paper. This algorithm reduces overall finish time without any direct communication between agents. Simulation results demonstrate the higher performance of our

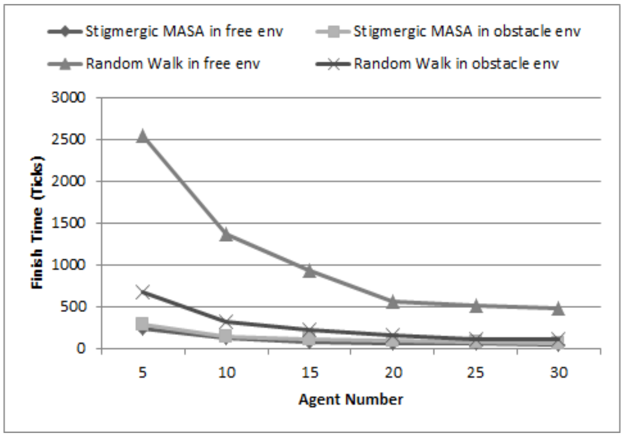


Fig. 8. Comparison of Stigmergic MASA with Random Walk when varying the number of agents

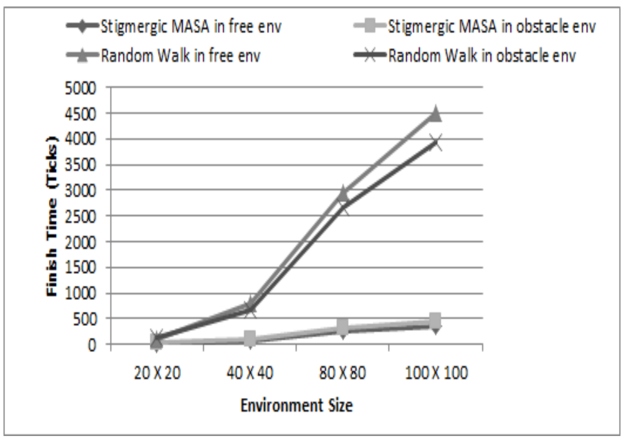


Fig. 9. Comparison of Stigmergic MASA with Random Walk when varying the environment size

algorithm in comparison to random walk strategy. Future work include improvements to accelerate searching time, applying the algorithm to dynamically changing environments, unknown ones and foraging problem [31].

REFERENCES

[1] S. Sarid, A. Shapiro, and Y. Gabriely, "Mrsam: A quadratically competitive multi-robot online navigation algorithm," in *ICRA'06*, 2006, pp. 2699–2704.

[2] E. Acar, H. Choset, Y. Zhang, and M. Schervish, "Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods," *International Journal of Robotics Research 22*, p. 441466, 2003.

[3] D. Gage, "Many-robot mcm search systems," in *Autonomous Vehicles in Mine Countermeasures Symposium, vol. 9*, 1995, p. 5664.

[4] G. Kantor, S. Singh, R. Peterson, D. Rus, A. Das, V. Kumar, and G. Pereira, "Distributed search and rescue with robot and sensor teams," in *Field and Service Robotics*, 2006, p. 529538.

[5] J. Jennings, G. Whelan, and W. Evans, "Cooperative search and rescue with a team of mobile robots," in *8th International Conference on Advanced Robotics, ICAR*, 1997, p. 193200.

[6] A. Marjovi, J. unes, L. Marques, and A. de Almeida, "Multi-robot exploration and fire searching," in *International Conference on Intelligent Robots and Systems, IROS*, 2009, p. 19291934.

[7] G. Landis, "Robots and humans: Synergy in planetary exploration," in *AIP Conference Proceedings, vol. 654*, 2003, p. 853.

[8] K. Schilling and C. Jungius, "Mobile robots for planetary exploration," *Control Engineering Practice 4*, 1996.

[9] M. A. Batalin and G. S. Sukhatme, "Spreading out: A local approach to multi-robot coverage," in *International Symposium on Distributed Autonomous Robotics Systems*, 2002, pp. 25–27.

[10] C. A. Pina-Garcia, G. Dongbing, and H. Huosheng, "Collaborative search on the plane without communication," in *ICIRA, Springer Verlag Berlin Heidelberg*, 2011, pp. 620–629.

[11] O. Feinerman, A. Korman, and Z. Lotker, "A composite random walk for facing environmental uncertainty and reduced perceptual capabilities," in *31st Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, PODC'12*, 2012, pp. 620–629.

[12] R. L. Stewart and R. A. Russell, "A distributed feedback macanism to regulate wall construction by a robotic swarm," *Adaptive Bahavior*, 2006.

[13] M. J. B. Krieger, J. B. Billeter, and L. Keller, "Ant-like task allocation and recruitment in cooperative robots," *Nature, Vol 406*, pp. 992–995, 2000.

[14] I. Roman-Ballesteros and C. Pfeiffer, "A framework for cooperative multi-robot surveillance tasks," in *Electronics, Robotics and Automotive Mechanics Conference*, 2006, pp. 163–170.

[15] M. Schwager, D. Rus, and J. J. Slotine, "Decentralized, adaptive coverage control for networked robots," *International Journal of Robotics Research, 28(3)*, pp. 357–375, 2009.

[16] J. C. dn S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *Robotics and Automation, IEEE Transactions on, 20(2)*, pp. 243–255, 2004.

[17] S. Sarid, "Heteregenous multi-robot search algorithms," Doctorat in Philosophy, Negev, Beer-Sheva, 2011.

[18] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continous areas by a mobile robot," *Mathematics and Artificial Intelligence*, 2001.

[19] R. Meir, Y. Peleg, and N. Pochter, "lecture 2," in *Mathematical Foundation of AI*, 2008.

[20] Y. Gabriely and E. Rimon, "Spiral stc: An on-line coverage algorithm of grid environments by a mobile robot," in *International Conference on Robotics and Automation*, 2002.

[21] Y. Gabriely and E. Rimonn, "Competitive on-line coverage of grid environments by a mobile robot," *Computational Geometry Theory and Applications*, 2003.

[22] N. Hazon and G. A. Kaminka, "On redendency, efficiency and robustness in coverage for multi-robot," *Autonoumous System 56*, 2008.

[23] N. Agmon, N. Hazon, and G. A. Kaminka, "Constructiong spanning trees for efficient multi-robot coverage," in *International Conference on Robotics and Automation*, 2006.

[24] X. Deng and A. Mirzaian, "On redendency, efficiency and robustness in coverage for multi-robot," *IEEE Transactions on Robotics and Automation*, pp. 532–542, 1996.

[25] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Robotic exploration as graph construction," *IEEE Transactions on Robotics and Automation*, pp. 859–865, 1991.

[26] M. Bender, A. Fernandez, A. S. D. Ron, and S. Vadhan, "The power of a pebble: Exploring and mapping directed graphs," *Information and computation*, 2002.

[27] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein, "Distributed covering by ant-robots using evaporating traces," *IEEE Transactions on Robotics and Automation*, pp. 918–933, 1999.

[28] I. A. Wagner and A. M. Bruckstein, "Special issue on ant robotics," *Annals of Mathematics and Artificial Intelligenece*, 2001.

[29] G. F. Orians and N. E. Pearson, "On the theory of central place forgaing," *Anal. Ecol. Syst*, pp. 155–177, 1979.

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

[31] O. Zedadra, N. Jouandeau, H. Seridi, A. Guerrieri, and G. Fortino, "A decentralized behavioral model for collective foraging problem," in *International Conference on Collaboration Technologies and Systems (CTS 2014), Minneapolis, Minnesota, USA*, 2014.