Proceedings of the 2014 Federated Conference on
Computer Science and Information Systems pp. 1477–1485
DOI: 10.15439/2014F395
ACSIS, Vol. 2

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

Ouarda Zedadra and Hamid Seridi
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, seridihamid@yahoo.fr

Nicolas Jouandeau
Advanced Computing laboratory
of saint-Denis Paris 8 University
Saint Denis 93526, 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*—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 S-MASA (Stigmergic 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 random walk and S-random walk (stigmergic random walk) strategies, 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 multiple agents in their searching task 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.

Complexity of multi-target search solutions depends on simplifications considered over idealized assumptions, such as: perfect sensors [14], stationary environments [15], unlimited direct communication [16]. Even if these assumptions are far from real world applications, they provide first basic solutions. 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 around agents each other;
3) agents use stigmergic communication via digital pheromone;
4) it can be executed 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. S-MASA algorithm is described in detail in Section 4. Performance evaluation is given in Section 5. A comparison with the random walk and S-random walk strategies are 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 (STC). In an STC algorithm, the robot operates in a 2D grid of large square cells. It aims to find a spanning tree for such grid, and allow the robot to circumnavigate it. This algorithm covers every cell that is accessible from the starting point, 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 move to the next disc not yet covered; an extension of this algorithm that uses heterogeneous robots is given in [17]. Instead of focusing on the on-board resources, some part of robotics literature use a single ant or a group of robots to cover an area robustly, even if they do not have any memory, do not know the terrain, cannot 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 greedy navigation strategies [28].

Whether we deal with coverage, multi-target search as foraging task, we need at the first stage to search the corresponding area. A search is defined as the action to look into the area carefully and thoroughly in an effort to find or discover something [29]. In most search strategies based on random walk, the agent tends to return to the same point many times before finally wandering away, because it has no historical information about visited regions. But when time and energy consumption are determinants, it will be efficient to guide the agent to not visited regions and repulse it from visited ones. In [30] a cooperative and distributed coordination strategy (IAS-SS) is proposed, it is applied to exploration and surveillance of unknown environments. It is a modified version of the artificial ant system, where the pheromone left has the property of repealing of robots either than attraction. A guided probabilistic exploration strategy for unknown areas is presented in [31], it is based on stigmergic communication and combines the random walk movements and the stigmergic guidance. The paper [32], provide a simple foraging algorithm that works asynchronously with identical ants, based on marking visited grid points by pheromone. It lacks robustness to faults. Authors in [33], propose a swarm intelligence based algorithm for distribute search and collective clean up. In this algorithm, the map is divided into a set of distinct sub-area and each sub-area is divided into some grid. Each robot decides individually based on its local information to which subarea it should move. A direct communication via WIFI model is used between robots and their neighbors. The paper [11], introduce the ANTS (Ants Nearby Treasure Search) problem, in which k identical agents, initially placed at some central location, collectively search for a treasure in a two-dimensional plane, without any communication between them. A survey of online algorithms for searching and exploration in the plane is given in [34]. S-MASA is a simple search algorithm that uses pheromones to guide the search process, agents are reactive and do not need any memory. It can locate nearby targets as fast as possible and at a rate that scales well with the number of agents, so it operates as some animal species that search for food around a central location, known as central place foraging theory [35]. Table I gives a comparison between our algorithm and some of the related works according to the search process used.

Even if chemical substances [36], electrical devices such as Radio Frequency Identification Devices (RFIDs) [37] [38] [39] [40] are examples of real implementation of stigmergic communication in real world experiments, it is still important to understand and improve pheromone-based algorithms in simulations. By understanding the optimal conditions required for pheromone-based coordination, the real world implementations can also be better directed [31].

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 cells. 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 1 to avoid accumulation of pheromone.

TABLE I: A Comparison of Related Work

Reference	[32]	[30]	[31]	[11]	[33]	S-MASA
Multi-Agent	Yes	Yes	Yes	Yes	Yes	Yes
Heterogeneous	No	Yes	Yes	No	Yes	No
I.L	(0,0)	Room 1	Random	(0,0)	Random	Given
Online	Yes	Yes	Yes	Yes	Yes	Yes
Environment	I.2D.G	2D.G.R	B.2D.G	I.2D.G	B.2D.G	B.2D.G
Sensors	F.G.N	LS.R.S	F.N	L.R.S	F.G.N	F.G.N
Simulations	No	2DX	Robots	No	3DX	Agents
Redundancy	No	Yes	Yes	No	No	Yes
Robustness	No	Yes	Yes	Yes	Yes	No
Complete	Yes	Yes	Yes	Yes	Yes	Yes
Distributed	Yes	Yes	Yes	Yes	Yes	Yes
Collaboration	Yes	Yes	Yes	Yes	Yes	Yes
Communication	S.C	S.C	S.C	No	Direct	S.C
Problem	M.T.S	E.S	Exploration	ANTS	D.S.C	M.T.S and C

I.L: Initial Locations, I.2D.G: Infinite 2D grid, 2D.G.R: 2D Grid with 7 Rooms, B.2D.G: Bounded 2D Grid, F.G.N: Four Grid Neighbors, F.N: Five Neighbors, LS.R.S: LaSer Range Sensor, L.R.S: Low Range Sensor, M.T.S: Multi-Target search, E.S: Exploration and Surveillance, D.S.C: Distributed Search and Clean up, S.C: Stigmergic Communication, C: Coverage

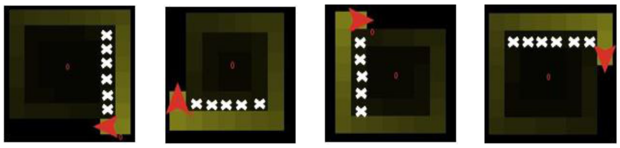
- *Motion policy:* each agent chooses the next cell to visit using a motion policy that is function of the presence of pheromone trail and obstacles. This policy helps the agent to decide where to go next.

IV. Design of the S-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 S-MASA

In S-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. S-MASA is further detailed in Algorithm 1.



(a) (b) (c) (d)

Fig. 1: S-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, where white crosses represent already visited cells

Algorithm 1 S-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 $(t+1)$ and is set to 0.075 to avoid unlimited accumulation of pheromone. S-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. S-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 [35]. 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 with coverage problem in known or unknown environments.

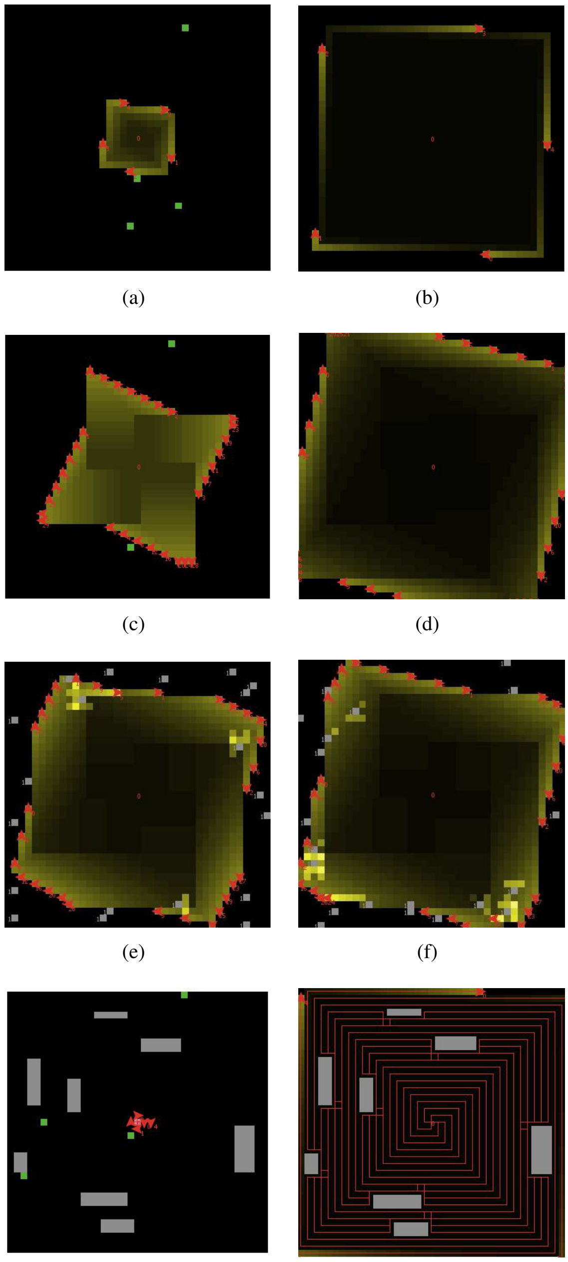
V. Performance Evaluation

We used Netlogo framework [41] to evaluate the performance of our algorithm in two scenarios. In the first scenario we evaluate the algorithm by varying the number of agents from 5 to 30 agents 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 cells to 100 X 100 cells. 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 S-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 from initial given positions and each agent has a heading, we vary the number of agents from 5 to 30. The environment consists of a square of size 40 X 40 cells shown in Figure 2, free or with obstacles, with four targets distributed randomly. An example of execution of S-MASA on a group of 5 agents, 30 agents on obstacle-free environment, a group of 30 agents on obstacle environment (obstacles are uniformly distributed) and a group of 5 agents on obstacle cluster environment (where the distribution of obstacles in the environment gives a cluster or line shapes either than the uniform distribution) are illustrated in Figure 2.

Table II 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 only 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.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 2: The evolution of search achieved by S-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. (g), (h) Initial and final position of 5 agents group in an obstacle cluster environment.

TABLE II: Effect of agent 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

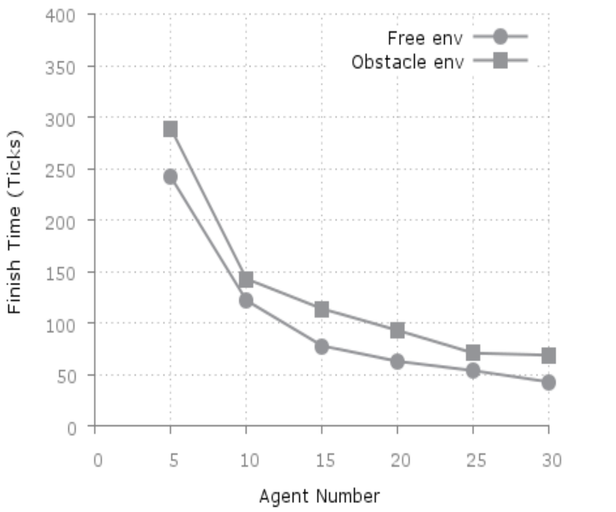


Fig. 3: Effect of agent number on performance in obstacle-free and obstacle environment

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 III 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, S-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 III: 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: S-MASA for Coverage Problem

Simulations presented in this section show that by changing the finish condition of the algorithm, the agents can achieve

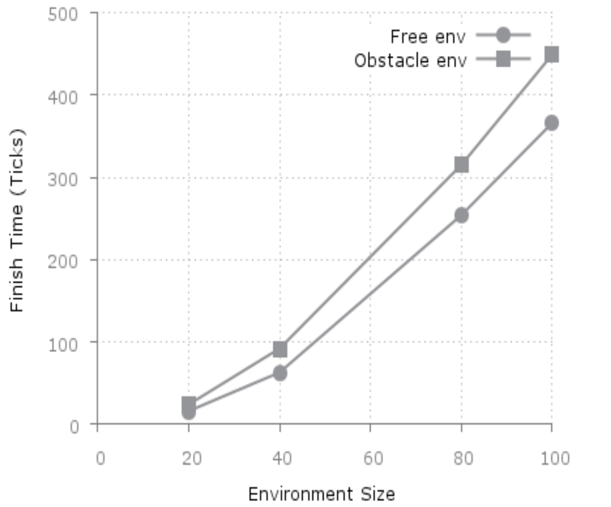


Fig. 4: Effect of environment size on performance in obstacle-free and obstacle environment

coverage mission as well as search one. The S-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 IV 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, which are similar to Scenario 1 results. A possible reason is the random distribution of targets, so if there is one target close to boundaries, the search will be very close to coverage task and in the two tasks the number of iterations will be very close.

TABLE IV: 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 V 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.

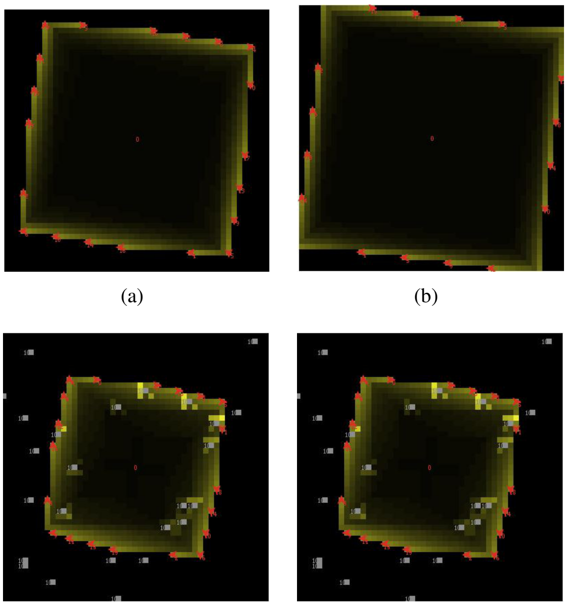


Fig. 5: The evolution of coverage achieved by S-MASA: (a), (b) 20-agents group in an obstacle-free environment in iterations 78 and 101. (c), (d) 20-agent group in an obstacle environment in iterations 44 and 108.

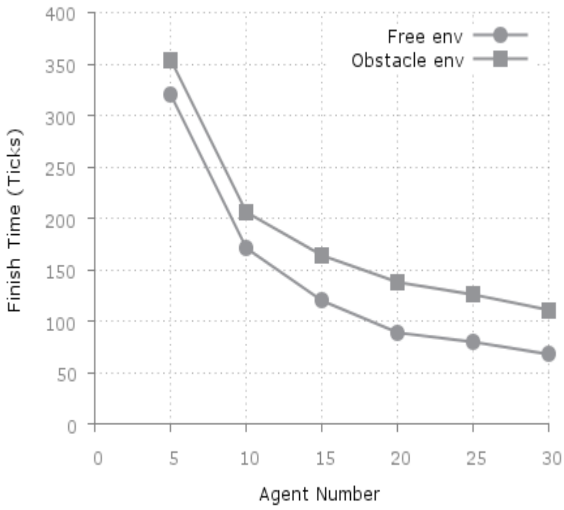


Fig. 6: Finish time of coverage in free-obstacle and obstacle environment when varying the number of agents

TABLE V: 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

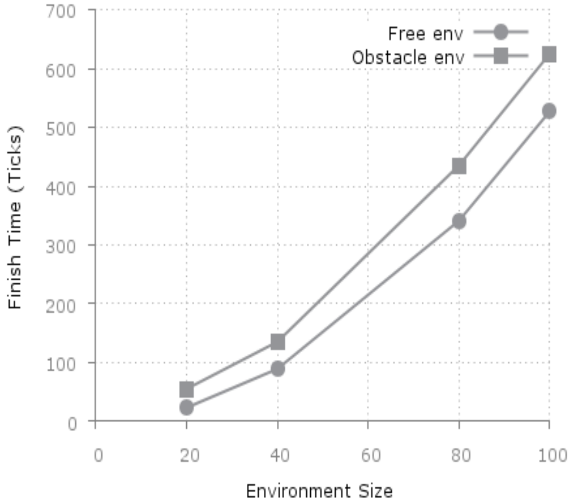


Fig. 7: Finish time of coverage in free-obstacle and obstacle environment when varying the environment size

VI. Comparison results

We compare S-MASA algorithm with two search strategies: the random walk, in which the agent chooses randomly one of the four neighbors even if it is already visited, that causes a increase in the global finish time; and the S-random walk, in which the agent chooses one of its four neighbors that are not visited yet, at each step the agent deposit a pheromone to mark already visited cells. Tables II and VI show the obtained results with S-MASA algorithm, random walk and S-random walk respectively when varying the number of agents from 5 to 30 in free-obstacle and obstacle environment where obstacles are uniformly distributed (2 *(e), (f)*). Figure 8 represents a comparison between these strategies according to Tables II and VI. Our algorithm performs much better than the random walk and the S-random walk, when the number of agents is less than 15. The results of the three strategies are close when the number of agents is more than 15, but our algorithm gives the best results.

TABLE VI: Effect of agent number on performance in random walk and S-random walk

	5	10	15	20	25	30
random walk free	2536,3	1365,65	932,6	567,05	508,7	487,3
STD Deviation	2021,98	1014,65	811,29	271,43	242,56	340,66
random walk obs	673,5	313,6	218,35	164,7	111,45	105,35
STD Deviation	853,37	223,99	207,19	83,20	43,07	34,95
S-random walk free	320	171	120	89	80	68
STD Deviation	354,25	206,7	164,3	138,6	126,25	111,55
S-random walk obs	798,7	543,1	306,7	205,7	182	157,2
STD Deviation	653,18	343,38	87,74	129,71	99,00	66,57

Tables III and VII show the obtained results with S-MASA algorithm, random walk and S-random walk respectively when varying the size of the environment from 20 X 20 cells to

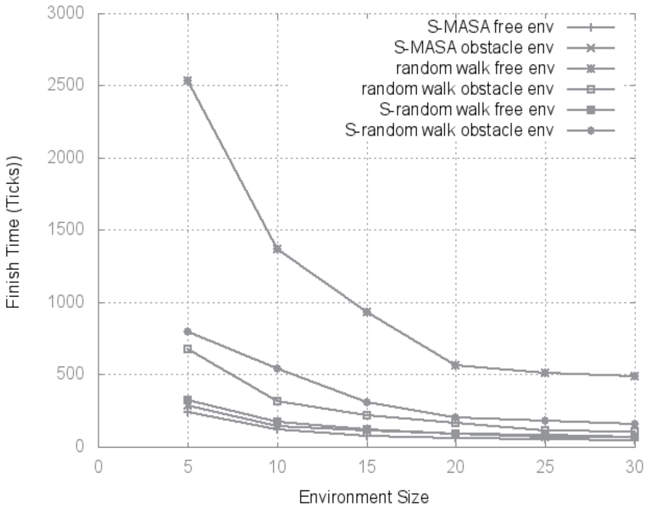


Fig. 8: Comparison of S-MASA with random walk and S-random walk when varying the number of agents (uniform distribution of obstacles)

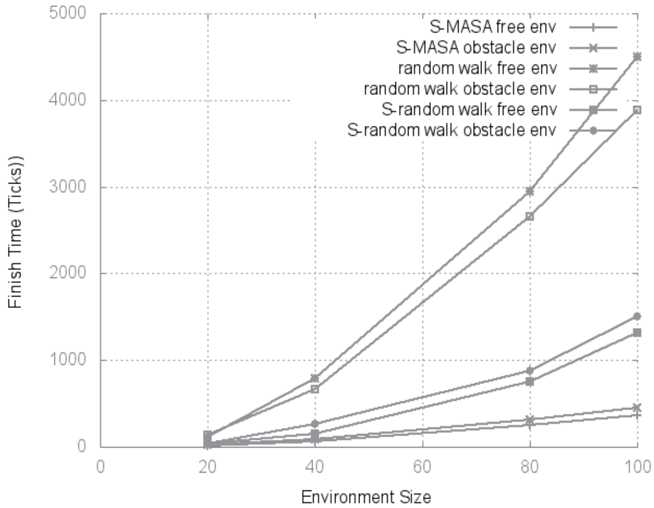


Fig. 9: Comparison of S-MASA with random walk and S-random walk when varying the environment size (uniform distribution of obstacles)

100 X 100 cells in free-obstacle and obstacle environment where obstacles are uniformly distributed (2 *(e), (f)*). Figure 9 represents a comparison between these strategies according to Tables III and VII. When the size of the environment is equal to 20 X 20 cells , the three strategies provide very close results. By increasing the environment size from 40 X 40 cells to 100 X 100 cells, our algorithm outperforms the two others.

TABLE VII: Effect of environment size on performance in random walk and S-random walk

	20X20	40X40	80X80	100X100
random walk free	108,2	789,65	2946,8	4501,85
STD Deviation	81,08	625,99	1398,71	2169,28
random walk obs	137,85	665,45	2651,5	3889,9
STD Deviation	68,93	376,37	1616,13	2307,50
S-random walk free	37,05	149,8	754	1312,25
STD Deviation	10,21	38,39	378,31	546,87
S-random walk obs	39,65	261,5	882,4	1502,75
STD Deviation	26,12	141,18	529,19	830,73

A comparison between the three strategies when varying the number of agents from 5 to 30 in an obstacle cluster environments (Figure 2 *(g), (h)*) is given in Figure 10, where our algorithm proves its performance among pure random walk and S-random walk. Table VIII shows the obtained results and the standard deviation in each simulation for the three strategies.

TABLE VIII: Effect of number of agent on performance in S-MASA, random walk and S-random walk

	5	10	15	20	25	30
S-MASA	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
random walk	3472,95	1393,05	930,85	639,7	474,05	370,45
STD Deviation	2440,36	435,20	542,51	301,39	237,19	144,05
S-random walk	886,95	428,95	319,9	188,1	149,65	106,95
STD Deviation	717,16	230,36	398,51	96,82	79,94	39,23

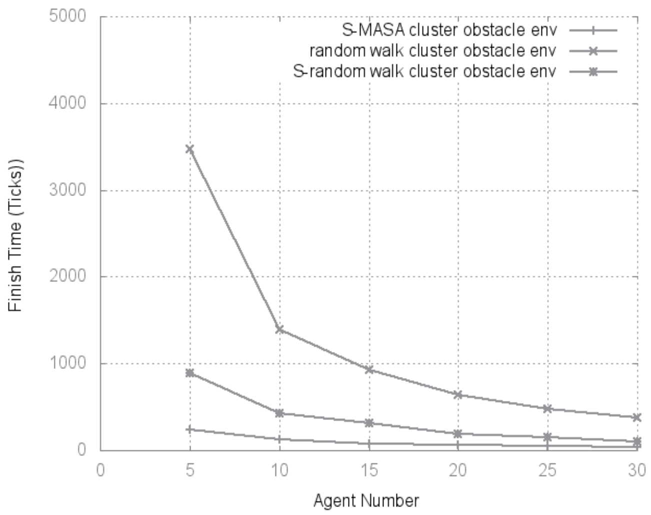


Fig. 10: Comparison of S-MASA with random walk and S-random walk when varying the number of agents (cluster obstacle environment)

VII. Conclusion

A multi-target search algorithm called S-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 algorithm in comparison with the S-random walk which is guided by pheromones to repulse agents from visited areas and with random walk strategy. We believe that future work improvements should reduce searching time, consider more complex, dynamic and unknown environments in the context of foraging problem.

References

[1] S. Sarid, A. Shapiro, and Y. Gabriely, "Mrsam: A quadratically competitive multi-robot online navigation algorithm," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. DOI: 10.1109/ROBOT.2006.1642109, IEEE, 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(7-8)*, pp. 441–466, 2003.

[3] D. Gage, "Many-robot mcm search systems," in *Autonomous Vehicles in Mine Countermeasures Symposium, vol. 9*. DOI: 10.1.1.38.771, 1995, pp. 56–64.

[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*. DOI: 10.1007/10991459-51, Springer Berlin Heidelberg, 2006, pp. 529–538.

[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*. DOI: 10.1109/ICAR.1997.620182, IEEE.

[6] A. Marjovi, J. unes, L. Marques, and A. de Almeida, "Multi-robot exploration and fire searching," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. DOI: 10.1109/IROS.2009.5354598, IEEE, 2009, pp. 1929–1934.

[7] Landis and A. Geoffrey, "Robots and humans: Synergy in planetary exploration," in *SPACE TECHNOLOGY AND APPLICATIONS INT. FORUM-STAIF 2003: Conf. on Thermophysics in Microgravity; Commercial/Civil Next Generation Space Transportation; Human Space Exploration; Symps. on Space Nuclear Power and Propulsion (20th); Space Colonization (1st)*, vol. 654, no. 1. DOI: org/10.1063/1.1541377, AIP Publishing, 2003, pp. 853–860.

[8] K. Schilling and C. Jungius, "Mobile robots for planetary exploration," *Control Engineering Practice*, vol. 4, no. 4, pp. 513–524, 1996.

[9] M. A. Batalin and G. S. Sukhatme, "Spreading out: A local approach to multi-robot coverage," in *Distributed autonomous robotic systems 5*. DOI: 10.1007/978-4-431-65941-9-37, Springer, 2002, pp. 373–382.

[10] C. A. Pina-Garcia, G. Dongbing, and H. U. Huosheng, "A composite random walk for facing environmental uncertainty and reduced perceptual capabilities," in *Intelligent Robotics and Applications*. DOI: 10.1007/978-3-642-25486-4-62, Springer, 2011, pp. 620–629.

[11] O. Feinerman, A. Korman, Z. Lotker, and J. S. Sereni, "Collaborative search on the plane without communication," in *Proceedings of the 2012 ACM symposium on Principles of distributed computing*. DOI: 10.1145/2332432.2332444, ACM, 2012, pp. 77–86.

[12] R. L. Stewart and R. A. Russell, "A distributed feedback mechanism to regulate wall construction by a robotic swarm," *Adaptive Behavior*, vol. 14, no. 1, pp. 21–51, 2006.

[13] M. J. B. Krieger, J. B. Billeter, and L. Keller, "Ant-like task allocation and recruitment in cooperative robots," *Nature*, vol. 406, no. 6799, pp. 992–995, 2000.

[14] I. Roman-Ballesteros and C. PfeifferF, "A framework for cooperative multi-robot surveillance tasks," in *Electronics, Robotics and Automotive Mechanics Conference, 2006*, vol. 2. DOI: 10.1109/CERMA.2006.3, IEEE, 2006, pp. 163–170.

[15] M. Schwager, D. Rus, and J. J. Slotine, "Decentralized, adaptive coverage control for networked robots," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.

[16] J. C. dn S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," in *IEEE International Conference on Robotics and Automation, ICRA'02*, vol. 2. DOI: 10.1109/ROBOT.2002.1014727, IEEE, 2002, pp. 1327–1332.

[17] S. Sarid, "Heterogeneous multi-robot search algorithms," Ph.D. dissertation, Ben-Gurion University of the Negev, 2011.

[18] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 77–98, 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 *Proceedings IEEE International Conference on Robotics and Automation, ICRA'02.*, vol. 1. DOI: 10.1109/ROBOT.2002.1013479, IEEE, 2002, pp. 954–960.

[21] Y. Gabriely and E. Rimonn, "Competitive on-line coverage of grid environments by a mobile robot," *Computational Geometry*, vol. 24, no. 3, pp. 197–224, 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, "Constructing spanning trees for efficient multi-robot coverage," in *IEEE International Conference on Robotics and Automation, ICRA 2006*. DOI: 10.1109/ROBOT.2006.1641951, IEEE, 2006, pp. 1698–1703.

[24] X. Deng and A. Mirzaian, "Competitive robot mapping with homogeneous markers," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 532–542, 1996.

[25] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Robotic exploration as graph construction," *Robotics and Automation, IEEE transactions on*, vol. 7, no. 6, 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," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. DOI: 10.1145/276698.276759, ACM, 1998, pp. 269–278.

[27] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein, "Distributed covering by ant-robots using evaporating traces," *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 5, pp. 918–933, 1999.

[28] I. A. Wagner and A. M. Bruckstein, "From ants to a (ge) nts: A special issue on ant-robotics," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1, pp. 1–5, 2001.

[29] D. C. V. Méndez and F. Bartumeus, "Random search strategies," *Stochastic Foundations in Movement Ecology, Springer-Verlag Berlin Heidelberg*, pp. 177–205, 2014.

[30] M. F. R. Calvo, J. R. de Oliveira and R. A. F. Romero, "Bio-inspired coordination of multiple robots systems and stigmergy mechanims to cooperative exploration and surveillance tasks," in *Cybernetics and Intelligent Systems (CIS), 2011 IEEE 5th International Conference on*. DOI: 10.1109/ICCIS.2011.6070332, IEEE, 2011, pp. 223–228.

[31] I. T. T. Kuyucu and K. Shimohara, "Evolutionary optimization of pheromone-based stigmergic communication," in *Applications of Evolutionary Computation*. DOI: 10.1007/978-3-642-29178-4-7, Springer, 2012, pp. 63–72.

[32] C. Lenzen and T. Radeva, "The power of pheromones in ant foraging," in *1st Workshop on Biological Distributed Algorithms (BDA)*, 2013.

[33] A. L. D. Liu, X. Zhou and H. Guan, "A swarm intelligence based algorithm for distribute search and collective cleanup," in *Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on*, vol. 2. DOI: 10.1109/ICICISYS.2010.5658776, IEEE, 2010, pp. 161–165.

[34] S. K. Ghosh and R. Klein, "Online algorithms for searching and exploration in the plane," *Computer Science Review*, vol. 4, no. 4, pp. 189–201, 2010.

[35] G. H. Orians and N. E. Pearson, "On the theory of central place foraging," *Analysis of ecological systems. Ohio State University Press, Columbus*, pp. 155–177, 1979.

[36] T. H. R. Fujisawa, H. Imamura and F. Matsuno, "Communication using pheromone field for multiple robots," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. DOI: 10.1109/IROS.2008.4650971, IEEE, 2008, pp. 1391–1396.

[37] R. Johansson and A. Saffiotti, "Navigating by stigmergy: A realization on an rfid floor for minimalistic robots," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. DOI. 10.1109/ROBOT.2009.5152737, IEEE, 2009, pp. 245–252.

[38] T. Sakakibara and D. Kurabayashi, "Artificial pheromone system using rfid for navigation of autonomous robots," *Journal of Bionic Engineering*, vol. 4, no. 4, pp. 245–253, 2007.

[39] L. M. A. F. V. A. Ziparo, A. Kleiner and D. Nardi, "Cooperative exploration for usar robots with indirect communication," in *Proc. of 6th IFAC Symposium on Intelligent Autonomous Vehicles, IAV, DOI: 10.3182/20070903-3-FR-2921.00094*, 2007.

[40] G. W. B. Ranjbar-Sahraei and A. Nakisaee, "A multi-robot coverage approach based on stigmergic communication," in *Multiagent System Technologies*. DOI: 10.1007/978-3-642-33690-4-13, Springer, 2012, pp. 126–138.

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