Exploring unknown environments with multi-modal locomotion swarm

Zedadra Ouarda[a], Jouandeau Nicolas[b], Seridi Hamid[a], Fortino Giancarlo[c]

[a]*LabSTIC Laboratory, 8 may 1945 University*
P.O.Box 401, 24000 Guelma, Algeria
[b]*LIASD, Paris8 University, Saint Denis, France*
[c]*DIMES, Universita' della Calabria*
Via P. Bucci, cubo 41c - 87036 - Rende (CS) - Italy

Abstract

Swarm robotics is focused on creating intelligent systems from large number of simple robots. The majority of nowadays robots are bound to operations within mono-modal locomotion (i.e. land, air or water). However, some animals have the capacity to alter their locomotion modalities to suit various terrains, operating at high levels of competence in a range of substrates. One of the most significant challenges in bio-inspired robotics is to determine how to use multi-modal locomotion to help robots perform a variety of tasks. In this paper, we investigate the use of multi-modal locomotion on a swarm of robots through a multi-target search algorithm inspired from the behavior of flying ants. Features of swarm intelligence such as distributivity, robustness and scalability are ensured by the proposed algorithm. Although the simplicity of movement policies of each agent, complex and efficient exploration is achieved at the team level.

Keywords: Swarm intelligence, Swarm robotics, Multi-target search, Random walk, Stigmergy, Multi-modal locomotion

1. Introduction

A search is defined as the action to look into or over carefully and thoroughly in an effort to find or discover something [1]. When agents lack information regarding targets, systematic searches become less effective and using random walk can enhance the chance of locating resources by increasing the chances of covering certain regions. In random strategies, the random walker (mobile robot or synthetic agent) returns to the same point many times before finally wandering away, which affects determinant parameters such as energy consumption [2], time and risks of malfunctions of agents. Stigmergy-based coordination allows very efficient distributed control and optimization. It has several other properties which are also essential to multi-robot systems, including robustness, scalability, adaptability and simplicity [3].

In [4] a cooperative and distributed coordination strategy called Inverse Ant System-Based Surveillance System (IAS-SS) 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 repelling robots rather than attracting them. A guided probabilistic exploration strategy for unknown areas is presented in [5], it is based on stigmergic communication and combines the random walk movements and the stigmergic guidance. The paper [6], provides a simple foraging algorithm that works asynchronously with identical ants, based on marking visited grid points with pheromone. It lacks robustness to faults. Authors in [7], 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-areas 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 [8], introduces the Ants Nearby Treasure Search (ANTS) problem, in which identical agents, initially placed at some central location, collectively search for a treasure in a two-dimensional plane without any communication. A survey of online algorithms for searching and exploration in the plane is given in [9].

Swarm robotics is the study of how a large number of simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interaction among agents and between agents and the environment. The mono-modal locomotion has been the principal interest of swarm robotics [10] for so long period but also using heterogeneous robots with different locomotion has been investigated [11]. However, multi-modal locomotion seems to be very interesting in order to allow agents performing a variety of tasks adaptively in different environments. Swarm robotics with mono-modal locomotion remains an active research area whose promise remains to be demonstrated in an industrial setting. Swarm robotics with multi-modal locomotion constitutes a new orientation that can benefit from the developed applications and open the issue to the development of new coordination and cooperation strategies.

Flying Ant-like Searcher Algorithm (FASA), proposed in this paper, is a multi-target search algorithm. In order to avoid returning to the same place several times in a random walk search strategy we used stigmergic communication through pheromone to mark covered regions. Through simulations we observe that agents get stuck in covered regions when their number is high and only some of them can get out of the covered regions. Therefore we propose the flying behavior whenever the neighborhood is totally covered, then we use flying behavior to return to specific locations which we call *best positions*, memorized by the agent when its current cell has at least one neighbor not covered yet. These cells are considered best positions because they allow a gradual search from the starting point and the flying behavior of agents to such cells ensures that all previous cells will be covered before going far away from them. It is an algorithm with a low computational complexity and designed for agents (and so far for robots) with very simple low-range sensors and indirect communication known as stigmergy.

The rest of the paper is organized as follows: in section 2, we present the problem formulation. In section 3, we present the behavior of ants from which the proposed algorithm is inspired and then we give the finite state machine of our agents and the pseudo code of the proposed algorithm. In section 4, we present the scenarios and performance metrics used in simulations, after that, we present the obtained results and compare them with Random Walk (RW) and Stigmergic Random Walk (SRW). We finish with a conclusion and future perspectives in section 5.

2. Problem 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 [2] [3]. In this work, we intend to design a search algorithm which allows a group of simple agents to locate a set of targets placed at random positions in the search space. The finish time of the collective search is when all targets have been found.

The basic concepts we use in the rest of the paper are defined as follow:

- *Environment*- A two dimensional finite grid E with NXM size. $E = E_{free} \cup E_{occupied}$, where $E_{occupied}$ denotes the subset of E containing the cell occupied by obstacles, targets or agents and $E_{free} = E_{covered} \cup E_{Ncovered}$ where $E_{covered}$ denotes the subset of E_{free} containing the covered cells (containing a pheromone) and $E_{Ncovered}$ denotes the subset of E_{free} containing the not yet covered cells. We define also E_{Best} as a subset of $E_{covered}$ containing the best positions stored by an agent. We denote a current cell C_c with coordinates $(x, y) \in E_{Best}$ if $\exists$ at least a neighbor cell C_n with coordinates $(x-1, y), (x+1, y), (x, y-1), (x, y+1) \in E_{Ncovered}$.

- *Target*- A set of static objects $T = t_1...t_n$, where n, the total number of targets is $>= 1$ and each t_i is placed at random positions in E.

- *Agent*- An Ant-like agent, which is capable of:

  1. Perceiving the four neighboring cells (detect the presence of pheromone, targets and obstacles);
  2. Depositing pheromone on current cell (to mark it as covered);
  3. Localizing itself;
  4. Moving and flying. Moving corresponds to one move from cell A to cell B with a distance of one step in one of the four directions up, down, left or right, while in flying the distance is $>= 1$;
  5. Memorizing the coordinates of best position cells;

- *Pheromone ($\mathbb{P}$)*- Chemical substance deposited by agents on visited cells to mark them as covered. It evaporates with time t.

3. Flying Ant-like Searcher Algorithm (FASA)

The FASA algorithm is a combination of random walk, stigmergic communication, and systematic search (using stored information). We provide the agents with the capacity to fly like flying ants. While workers of the Camponotus japonicus species [12] do not have wings, young female and male have them. They use them to fly away from their nest for mating and building their own colony. Figures 1(a), 1(b) and 1(c) represent the behavioral model of worker, male and female of Camponotus japonicus ants and Figure 2 represents the behavior of our Flying Ant-like Searcher agent which combines the behaviors of worker and male (or female) Camponotus japonicus ants.

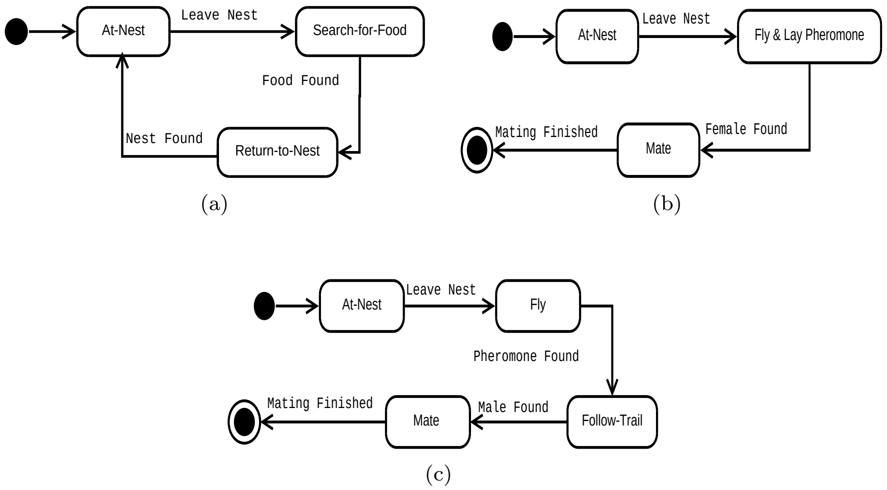


Figure 1: Behavioral model of: (a) worker, (b) male and (c) female Camponotus japonicus ants

Some particularities of FASA are: (1) agents explore gradually the sub-space in which they were initially deployed, (2) robustness to failure is provided as it works since a single agent is alive, (3) initial positions of agents, the geometry of the search space, complexity of obstacles do not influence the algorithm performances.

FASA consists of three steps:

1. Sets a temporization t to a random (value);
2. The agent repeats the following steps until $t = 0$:
   - Stores the coordinates of the current cell C_c in E_{Best}, if it has at least one neighbor not yet covered;
   - Deposits $\mathbb{P}$ on current cell C_c;
   - Moves to one of the four neighbors not yet covered;

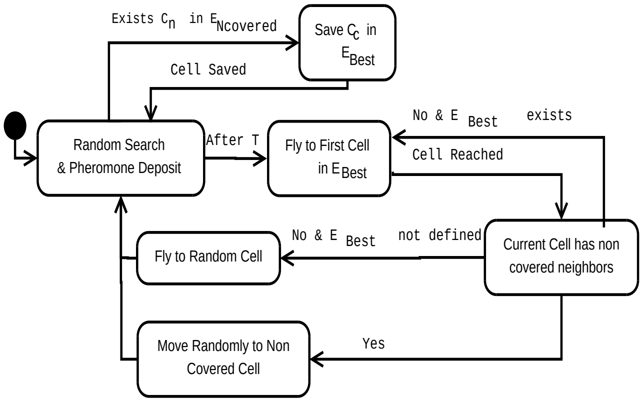

Exists Cn in ENcovered
Save Cc in EBest
Cell Saved
Random Search & Pheromone Deposit
After T
Fly to First Cell in EBest
No & E Best exists
Cell Reached
Current Cell has non covered neighbors
No & E Best not defined
Fly to Random Cell
Yes
Move Randomly to Non Covered Cell

Figure 2: State transition diagram of FASA Agents

Algorithm 1: FASA

```
while Targets not reached do
    $t \leftarrow$ random (value)
    $E_{Best} \leftarrow \emptyset$
    while $t \neq 0$ do
        if $\exists$ $C_n$ (a neighbor of $C_c$) $\in E_{Ncovered}$ then
            push_front $C_c$ in $E_{Best}$
        Deposit $\mathbb{P}$ on $C_c$
        Move randomly to $C_j \in E_{Ncovered}$
        $t \leftarrow t - time\_step$
    if $E_{Best} \neq \emptyset$ then
        Fly to first element in $E_{Best}$
        while ($E_{Best} \neq \emptyset$) and ($\nexists$ a neighbor of $C_{Best} \in E_{Ncovered}$) do
            Remove first element of $E_{Best}$
            Fly to next $C_{Best} \in E_{Best}$
    while ($C_c \in E_{covered}$) or (all neighbors of $C_c \in E_{covered}$) do
        fly randomly to new cell
```

3. Flies to the first best position in E_{Best}. If E_{Best} is void, keeps flying to random cells until finding a non covered cell or a cell having non covered neighbors and go to 1.

4. Performance Evaluation

4.1. Simulation Scenarios and Performance Metrics

Simulations are implemented and run on a Java-based simulation platform (Netlogo [13]). In all our experiments, targets were considered non-mobile. Ini-

Table 1: Experimental initialization of random value

	4	**9**	**14**	**21**	**43**	**50**	**100**
Time	*1096*	*643*	*625*	*597*	*278*	*561*	*569*
Targets Found	*169*	*230*	*188*	*163*	*285*	*230*	*228*

tial conditions such as: World Size (WS), Agents Number (AN), Obstacles Density (OD) – the total amount of obstacles is calculated by: $AmountofObstacles = OD \times (\frac{worldsize}{4})$ – and Number of Targets (NT), were varied from a scenario to another. We used obstacles with complex shapes for simulations of obstacle environments.

We used two metrics to evaluate the performance of our algorithm. We compare it to random walk (RW) [14] and to Stigmergic Random Walk (SRW) where robots use stigmergic communication to avoid already visited cells when they lay pheromone trails [15]:

- Search Time: is the time in seconds needed to discover all the targets in the environment.
- Search Efficiency: is defined by 1:

  $$Search_{eff} = \frac{Targets_{found}}{Targets_{total}} \times 100 \tag{1}$$

  where: $Search_{eff}$ denotes the percentage of found targets over the total number of targets, $Targets_{found}$ denotes the number of found targets during an elapsed time t and $Targets_{total}$ denotes the total number of targets placed at the environment.

We defined two scenarios where targets are placed randomly and agents start all from the center of the environment. The time t spent in random walk is fixed after experiments to *random (x) with* $x = 43$ (see Table 1). We varied value from 4 to 100 and then we recorded in the first scenario the average time required to find all the targets and in the second scenario the number of targets found. With Random (43), the time is the lowest and the number of targets found is the highest among the other values. Each simulation is performed 10 times, then the mean and standard deviation for each metric are computed:

- *Scenario 1:* to test the scalability of the algorithm, when increasing agents number. We fix the WS to 500 × 500 cells (Netlogo units), the NT to 40, the OD to 70% and we vary AN from 100 to 4000. We fix the time for each simulation to 240 sec and we report at the end the mean value of $Search_{eff}$.
- *Scenario 2:* to test the efficiency of search in larger environments. We fix AN to 300, NT to 40, OD to 70% and we vary WS from 200 × 200 cells to 1000 × 1000 cells.

4.2. Results and Discussion

Through the obtained results, FASA outperforms the two other protocols in both scenario 1 and 2. It is more efficient in locating targets and faster than the two others in searching the total number of targets in larger environments.

In scenario 1, the $Search_{eff}$ in FASA increases when increasing AN, it is about 48% with 100 agents and 100% over 300 agent. Also the search time decreases when increasing AN, from 225 sec (300 agents) to 95 sec (2000 agents) but over 4000 agents the search time starts at increasing (105 sec). FASA gives better results than SRW and RW algorithms. SRW's $Search_{eff}$ reaches 90% with 300 agents and then decreases till 28% with 4000 agents. RW gives the worst $Search_{eff}$ since there is no guidance in search, the $Search_{eff}$ increases slowly from 12% to 24% (with 100 to 800 agents respectively) over 800 agents it decreases till 7% with 4000 agents. Using large number of agents in SRW or RW causes agents to get stuck in already covered regions and agents keep turning in a closed covered region, while in FASA the flying behavior helps the agents to get out of the closed covered region and give them the chance to cover more regions and to find more targets (see Table 2 and Figure 3(a) for detailed results).

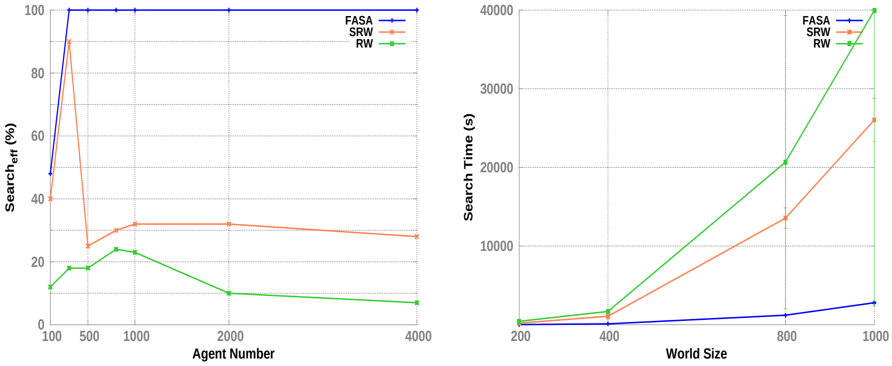
Table 2: $Search_{eff}$ of FASA, SRW and RW when increasing AN

	100	**300**	**500**	**800**	**1000**	**2000**	**4000**
FASA $Search_{eff}$	*48%*	*100%*	*100%*	*100%*	*100%*	*100%*	*100%*
FASA *Search Time*	240	225	164	113	106	95	105
SRW	40%	90%	25%	30%	32%	32%	28%
RW	12%	18%	18%	24%	23%	10%	7%

Table 3: Search Time of FASA, SRW and RW when increasing WS

	200x200	**400x400**	**800x800**	**1000x1000**
FASA	*19*	*109*	*1213*	*2793*
STD	*1*	*10*	*83*	*111*
SRW	201	1077	13554	26031
STD	46	354	1309	2761
RW	433	1686	20669	39941
STD	65	216	18661	37538

In scenario 2, search time increases in the three algorithms when increasing WS. FASA gives better results than SRW and RW. The search in SRW and RW becomes inefficient when world size is over 400 × 400 cells. In FASA the return to best positions by flying behavior results in gradual search over the whole



(a)

(b)

Figure 3: Simulation results of FASA, SRW and RW : (a) $Search_{eff}$ when increasing AN, (b) Search Time when increasing WS

search space even if there is some redundancy while searching (see Table 3 and Figure 3(b) for detailed results).

5. Conclusion

We presented in this paper a multi-target search algorithm called FASA. It tends to introduce guidance in search through pheromone, getting out from covered regions through the flying behavior and enabling a gradual search that ensures the completeness of the algorithm through the flying to best positions (stored while searching). Results obtained in comparison to random walk and pheromone guided random walk are promising. Parameters such as the random time of search and the distance of fly needs to be improved through simulations.

Stigmergic communication via pheromone has shown to efficiently coordinate a team of robots and to allow them to quickly explore a given area [16]. Simulations can support the analysis and improvement of pheromone-based algorithms before their real robotic implementation [5]. However, the implementation of the pheromone itself in real world constitutes a challenging technical issue. Several works proposed mechanisms to the real implementation of pheromone as: (i) physical marks using: *virtual marks* [16] or *RFID tags* [17], (ii) a model to be transmitted using wireless network [18], (iii) virtual pheromone transmitted using infrared communication [19], (iv) beacons where robots are used as pheromones [20]. Despite the proposed approaches, the implementation of pheromone is still in its early development stages and most of the works are available in research laboratories.

In order to test the applicability of the proposed algorithm, we intend to implement it in a robotic platform (ARGoS [21], Gazebo [22]) by also exploiting specific MAS-based methodologies [23] [24].

References

[1] V. Méndez, D. Campos, F. Bartumeus, Random search strategies, in: Stochastic foundations in movement ecology, Springer, 2014, pp. 177–205.

[2] O. Zedadra, H. Seridi, N. Jouandeau, G. Fortino, An energy-aware algorithm for large scale foraging systems, Scalable Computing: Practice and Experience 16 (4) (2016) 449–466.

[3] O. Zedadra, H. Seridi, N. Jouandeau, G. Fortino, A cooperative switching algorithm for multi-agent foraging, Engineering Applications of Artificial Intelligence 50 (2016) 302–319.

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

[5] I. T. T. Kuyucu, K. Shimohara, Evolutionary optimization of pheromone-based stigmergic communication, in: Applications of Evolutionary Computation, Springer, 2012, pp. 63–72.

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

[7] A. L. D. Liu, X. Zhou, H. Guan, A swarm intelligence based algorithm for distribute search and collective cleanup, in: IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), Vol. 2, IEEE, 2010, pp. 161–165.

[8] O. Feinerman, A. Korman, Z. Lotker, J. S. Sereni, Collaborative search on the plane without communication, in: Proceedings of the 2012 ACM symposium on Principles of distributed computing, ACM, 2012, pp. 77–86.

[9] S. K. Ghosh, R. Klein, Online algorithms for searching and exploration in the plane, Computer Science Review 4 (4) (2010) 189–201.

[10] B. Ranjbar-Sahraei, S. Alers, K. Tuyls, et al., Stico in action, in: Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems, International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 1403–1404.

[11] P. Pace, G. Aloi, G. Caliciuri, G. Fortino, A mission-oriented coordination framework for teams of mobile aerial and terrestrial smart objects, Mobile Networks and Applications (2016) 1–18.doi:10.1007/s11036-016-0726-4.

[12] T. Abe, On the behavior of the ant camponotus japonicus at the nuptial flight, Kontyu 41 (1973) 333–341.

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

[14] W. J. Bell, Searching behaviour: the behavioural ecology of finding resources, Springer Science & Business Media, 2012.

[15] L. Pitonakova, R. Crowder, S. Bullock, Understanding the role of recruitment in collective robot foraging, in: Proceedings of The Fourteenth International Conference on the Synthesis and Simulation of Living Systems (ALIFE), MIT Press, 2014, pp. 1477–1485.

[16] J. Svennebring, S. Koenig, Building terrain-covering ant robots: A feasibility study, Autonomous Robots 16 (3) (2004) 313–332.

[17] M. Mamei, F. Zambonelli, Spreading pheromones in everyday environments through rfid technology, in: 2nd IEEE Symposium on Swarm Intelligence, 2005, pp. 281–288.

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

[19] D. W. Payton, M. J. Daily, B. Hoff, M. D. Howard, C. L. Lee, Pheromone robotics, in: Intelligent Systems and Smart Manufacturing, International Society for Optics and Photonics, 2001, pp. 67–75.

[20] E. J. Barth, A dynamic programming approach to robotic swarm navigation using relay markers, in: Proceedings of the American Control Conference, Vol. 6, IEEE, 2003, pp. 5264–5269.

[21] C. Pinciroli, V. Trianni, R. OGrady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, et al., Argos: a modular, parallel, multi-engine simulator for multi-robot systems, Swarm intelligence 6 (4) (2012) 271–295.

[22] N. Koenig, J. Hsu, The many faces of simulation: Use cases for a general purpose simulator, in: Proc. of the ICRA, Vol. 13, 2013, pp. 10–11.

[23] G. Fortino, A. Garro, S. Mascillaro, W. Russo, Using event-driven lightweight dsc-based agents for mas modelling, International Journal of Agent-Oriented Software Engineering 4 (2) (2010) 113–140.

[24] G. Fortino, A. Garro, W. Russo, An integrated approach for the development and validation of multi-agent systems, International Journal of Computer Systems Science & Engineering 20 (4) (2005) 259–271.