

Terrain Coverage Ant Algorithms: The Random Kick Effect

M. Dervisi
Computer Science Department,
IST College Athens, Greece
in collaboration with
University of Hertfordshire, UK
E-mail: maria.dervisi@gmail.com

O.B. Efremides
Computer Science Department,
IST College Athens, Greece
in collaboration with
University of Hertfordshire, UK
E-mail: obe@ist.edu.gr

D.P. Iracleous
Computer Science Department,
IST College Athens, Greece
in collaboration with
University of Hertfordshire, UK
E-mail: iracleous@ist.edu.gr

Abstract—In this work the effect of random repositioning of ant robots/agents on the performance of terrain coverage algorithms is investigated. A number of well-known terrain coverage algorithms are implemented and studied in a simulated environment. We prove that agent repositioning imposes small variations on the performance of the algorithms when random or controlled jumps occurred and evaporation and failures are allowed.

Keywords—Terrain coverage, ant agents, performance

I. INTRODUCTION

Terrain coverage algorithms using ant robots/agents [1], [2], [3] is under investigation in this work. We focus on the jump effect of agents during the execution of the algorithms. This effect can be defined as an unexpected fast move of the agent, from its current position to a new one, in a way that is not defined by the terrain coverage algorithm. This is, for example, the case that visitors of a museum accidentally kick the surveillance ant robots.

Resulting functional algorithms can have applications in everyday use, such as vacuum cleaning a large area [4], [5]. Moreover, they can also be applied in critical emergency cases, like searching the terrain of a collapsed mine, patrolling a secure area, etc.

The key point in this work is that the executing algorithm should be able to complete the assigned task, even if the terrain or the agents are abruptly affected by natural phenomena, the human factor, a sudden failure, the presence of obstacles in the area etc. In any case, this should be as efficient as possible.

Throughout this work, jump is applied on well known terrain coverage algorithms [6], [7], various terrain types and initial agent positions. In all cases the objective is to examine the performance of terrain coverage and how it is affected due to jump and its combination with the simulation settings. The most important settings include, agent battery drain [3], the probability of agent failures [8] and the evaporation of pheromone left by agents on the terrain [7], [8].

The algorithms studied in this work, have as their main purpose to visit each area or cell of the terrain, at least once [6], [7], [8]. Hence, it can be said that their goal is it to achieve the exploration of unknown areas. These include Node-counting, Recency, Alarm, Learning Real-Time A* (LRTA*),

Online Mapping Algorithms (OMA) and Wagner [9], [3], [10]. Such algorithms have found practical and inexpensive implementations in many real life problems [7]. Additionally, a brute force approach using Random walk was also studied but was excluded from the main comparisons found in the experimental results section, as this implementation is of very low performance. As each of the aforementioned algorithms follows a slightly differentiated strategy in the way terrain exploration is achieved, the effects of simulation settings are examined for each algorithm in case.

This paper is organized as follows. In Section 2 the six algorithms under investigation are presented. The setting for the experiments is given in Section 3 while the simulation results are discussed in Section 4. Conclusions are drawn in Section 5.

II. THE ALGORITHMS

Six different algorithms are under investigations herein. Node counting:

The node-counting algorithm (also referred to as “counting” later) approaches the terrain coverage in a deterministic manner since it always chooses to move to the place with the least number of visits. At each step, each agent moves from its cell to an adjacent terrain cell, which is the one with the fewest number of visits (by all agents). In the case where there are more than one cells that have the fewest visits, the ties can be broken at random [7].

LRTA*: also referred to as “learning” later, is a deterministic target location algorithm which updates the number of visits of the current cell during each step. In this method, the number of visits acts as a repellent for each agent [10]. With each movement of the ant, the value of the current cell is updated by taking the number of visits of the adjacent cell (to which the ant is about to move) and incrementing it by one. In the case where there are more than one cell that have the fewest visits, the ties can be broken at random [11].

Wagner’s: Wagner proposed a deterministic terrain coverage method that updates the current cell value only if it is less than the value of the adjacent cell to which the ant is about to move. No update will occur to the current cell if the value of the current cell is higher than the cell to which the ant is moving to. This method is similar to both the LRTA* and

the Thrun's algorithms. The only difference is that the current value will not always be updated [7].

Thrun's - OMA: The algorithm proposed by Thrun [12] is also a deterministic terrain coverage model that is similar to LRTA*, the only difference is that it guarantees that the current cell value will be incremented, for each visited cell, during each step. Whenever the adjacent cells have lower cell values, the value of the current cell will still be increased as opposed to the LRTA* algorithm. With LRTA*, it is possible that if the adjacent cells have lower values, the current cell can have its value lowered. This does not happen with Thrun's algorithm [11].

Recency: Agens next move is decided based on the elapsed time since its adjacent cell has been visited. The only difference between this and node counting algorithm is that the recency counts in time units rather than the number of visits [8].

Alarm: In this algorithm the agent selects the next cell to move probabilistically based on the amount of pheromones in the adjacent cells. Moreover, the algorithm uses pheromone level to repel agents instead of attracting them. In each step, the agents move to cells with less pheromones [8].

III. EXPERIMENTAL SETTING

The Netlogo simulation environment [13] is used for implementing and testing our algorithms. In order to efficiently study the effect of jump for the above mentioned algorithms while having full control over a) the agents that might be affected and b) the frequency at which jump may occur, two different approaches were implemented:

- The human controlled jump setting and
- The random jump setting

Both settings have the same effect; agents are moved from their original position to a new one without visiting the intermediate cells and without leaving any trace element on the cells they skip. The difference is that the first, as its name implies, allows the control of how many agents from the total available will be affected, the maximum length of the jump and the frequency at which it can occur. Conversely, in the case of random jump all events occur in random, resulting in a more natural behavior, which however is less controlled and more difficult to extract useful conclusions about the role jump can play on terrain exploration algorithms.

Terrain types scale from the lowest complexity (None) to the most complex terrain type with special features (e.g., a number of tight borders/corridors) as it is shown in Figure 1.

IV. EXPERIMENTAL RESULTS

A. Initial Experiment - No Jump

The first experiment includes all algorithms, on all terrain types, without any specific environmental settings, in order to build our base for performance comparisons. As expected, brute force Random Walk algorithm proved much slower than all the others and it was removed from all subsequent tests. In this configuration, the OMA and the Wagner algorithms

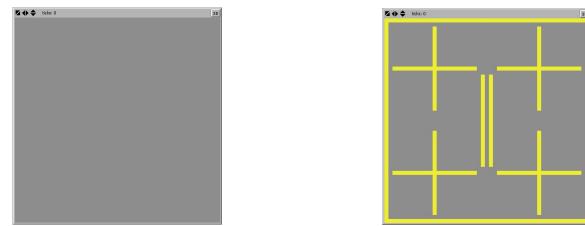


Fig. 1. Simplest to complex terrain types

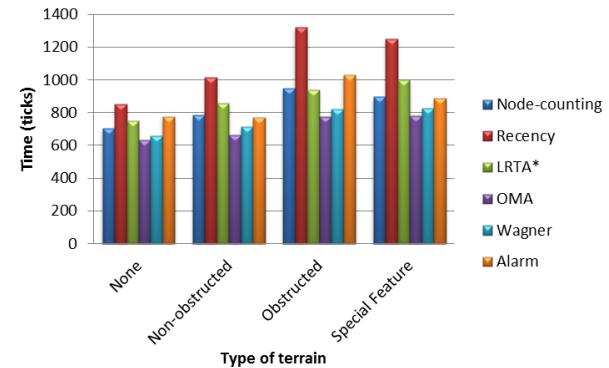


Fig. 2. Results excluding Random Walk (lower is better)

achieve the highest performance in covering the entire terrain. This is true even for the complex terrain settings, although the time required to complete the task was higher. Figure 2 shows the time results for all combinations of environments and algorithms.

B. Jump Enabled Experiment

Tests using either human controlled jump or random jump options both indicated a slightly positive influence (see Figure 3). On simple terrains where jump can freely take place (since there are no obstacles applying extra restrictions), it increases performance to an average of 4%. In the beginning of the coverage process where the terrain is mostly unvisited, through jump, agents are often given the chance to restart from a new area instead of revisiting the nearby cells. As the terrain is further covered agents again have the chance to go to a less visited area and hence decrease area revisiting while achieving faster area coverage.

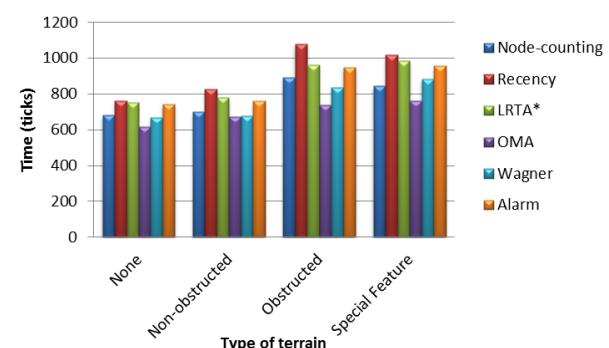


Fig. 3. Human controlled jump enabled

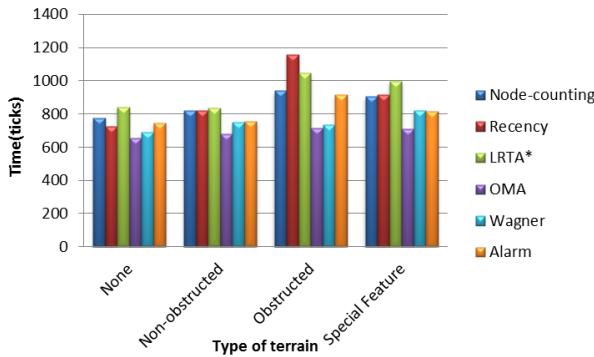


Fig. 4. Evaporation with human controlled jump

Further experiments, indicated that the above benefit is lost in highly obstructed terrains where the agent's jump gets frequently interrupted. In such setups there was actually a decrease in performance since during the jump effect, agents do not mark the cells they pass over and time is lost while their goal is to cover the terrain. It must be noted that when random jump was enabled the performance differences are not easily noticeable as the entire process occurs in random. This means that since there is no control over the distance, frequency and number of agents that are affected by jump, no safe conclusions can be made. Hence, in all cases the human controlled jump provided results that can be more easily predicted and interpreted, concerning the actual effect of jump on the terrain coverage process, for the purposes of this work.

C. Pheromone Evaporation and Jump

In the next series of experiments, pheromone evaporation was added to the environmental settings. OMA and Wagner again proved to be the fastest algorithms, while the performance of Alarm [9] was much closer to the rest. In addition, total coverage times were higher, as evaporation led to an increase in the revisits counter, compared to the previous experiment.

The addition of both types of jump resulted in a similar to the previous behavior. Specifically, human controlled jump clearly indicated an improved behavior for non-obstructed terrains, with its advantages being lost in the complex terrain setup. Again, this was less noticeable for the random occurring jump setting. The order of the algorithms based on their performance was not influenced by jump, with OMA and Wagner being the fastest ones (see Figure 4).

D. Failure Probability Experiments

The final series of experiments take under consideration the probability of one or more agents failing during the simulation. Actually, three different categories were simulated, with the results in all cases not being comparable in performance terms with the previous tests, as the number of working agents decreased during execution.

The first set examines fixed age failure probability. It can be deducted that the results seen from the experiment were affected by the too short life time of agents and that with more available time the coverage of simple terrains is performed

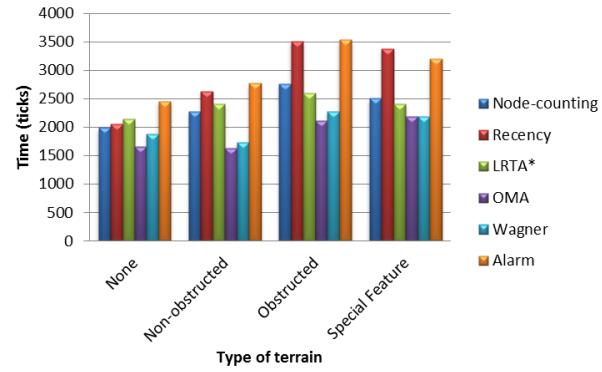


Fig. 5. Constant Failure Probability set to 80%

more efficiently. The addition of jump did not affect the results, which were almost similar in simple terrains and slightly slower in complex ones, when compared to not using jump.

The second sub-experiment series examine the case of the constant failure probability. OMA and Wagner had the most distinguishing performance. In both experiments with jump enabled, the performance was slightly improved for simple terrains. In contrast, it was slower for the obstructed terrains.

The final failure experiment included the increasing failure probability case. This was actually a more aggressive failure setting, compared to constant, as with time the chance of an agent to fail increases. Subsequently, the execution times are slower than the previous experiments that did not include the failure factor, as agents were decreased during the simulation.

The addition of human controlled and random jump appeared to play no significant role and in general its effect cannot be conclusive as the factor of an increasing failure probability obscures any definitive conclusions, as these were noticed and elaborated for the first experiments in this section.

V. CONCLUSIONS

This work attempts to explore the variations on the terrain coverage algorithm performance when the ant robots/agent can randomly or controlled jump to another position of the terrain instead of following the “next move” imposed by the applied each time algorithm heuristic rule. Based on the conducted experiments, we conclude that the jump effect has a small positive impact on the performance when referring to non-complex terrains. However, as the terrain becomes more complex, with more and more obstacles, a gradual degradation of the performance results is observed. Further, investigation can potentially link jumps with other terrain coverage metrics.

REFERENCES

- [1] M. Dorigo, V. Maniezzo, and A. Coloni, The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 26(1):29–41, 1996.
- [2] M. Dorigo, M. Birattari and T. Stutzle, Ant colony optimization, *Computational Intelligence Magazine, IEEE*, 2006
- [3] Marco Dorigo and Thomas Stutzle, *The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances*, International Series in Operations Research & Management Science, 2003, Volume 57, 250-285, DOI: 10.1007/0-306-48056-5_9

- [4] J. R. VanderHeide and N. S. V. Rao. Terrain coverage of an unknown room by an autonomous mobile robot. Technical report, Oak Ridge National Laboratory, Tennessee, United States, 1995.
- [5] F. Adler and D. Gordon, Information collection and spread by networks of patrolling ants, *The American Naturalist*, 140(3):373–400, 1992
- [6] Sven Knoenig and Yixin Liu, Terrain Coverage with Ant Robots: A Simulation Study, College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280, 2001
- [7] Jonathan Mason and Ronaldo Menezes, Autonomous Algorithms for Terrain Coverage Metrics, Classification and Evaluation, IEEE Congress on Evolutionary Computation, 2008
- [8] Robert Treverton and Ronaldo Menezes, Evaluating Failure in Terrain Coverage by Autonomous Agents, IEEE, 978-1-4244-2753-6, 2009
- [9] C. Lloyd. The alarm pheromones of social insects: A review. Technical report, Colorado State University, 2003
- [10] Eric Bonabeau , Marco Dorigo and Guy Theraulaz, Swarm intelligence: from natural to artificial systems, Oxford University Press, Inc., New York, NY, 1999
- [11] S. Thrun, Efficient exploration in reinforcement learning, Technical Report CMU-CS-92-102, School of Computer Science, Carnegie Mellon University, 1992
- [12] Howie Choset, Coverage for robotics – A survey of recent results, *Annals of Mathematics and Artificial Intelligence* Volume 31, Numbers 1-4, 113-126, DOI: 10.1023/A:1016639210559
- [13] U. Wilensky, NetLogo. Center for Connected Learning and Computer-Based Modeling, Evanston, 1999