2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics
Suntec Convention and Exhibition Center
Singapore, July 14-17, 2009

# Probabilistic Ants (PAnts) in Multi-Agent Patrolling

James Guo Ming Fu and Marcelo H. Ang Jr

***Abstract*— We propose a *Probabilistic Ants* (PAnts) Algorithm for solving the *Multi-Agent Patrolling Problem* in an online and robust manner, based purely on local information. As only local information is required, this strategy can be deployed distributively. As our proposed strategy does not require a pre-processing of the map, it can be used for a map with a dynamic topology as well as dynamically changing number of agents. Our proposed strategy makes use of virtual pheromone traces which will act as potential fields, guiding each agent towards areas which have not been visited for a long time. Each agent only needs to make its decision on where to go next based on its local pheromone information. It does not need to keep a topology of the map in memory. Decision making is done probabilistically based on local pheromone information. This method is also non-intrusive to the environment and all traces are kept in virtual memory. In our experimental evaluation, we compare our method with the traditional Ant Algorithm as well as a variant of it. All three methods are benchmarked against the theoretical ideal for clarity.**

## I. INTRODUCTION

Patrolling is the task of traversing an environment repeatedly for the continual purpose of information updating. Applications for this ranges from surveillance [1], inspection [2], security [3][4], intrusion detection and cleaning [5].

A good patrolling strategy is one which minimizes the time delay between two successive visits for every important location throughout a map. It may be just certain key features on a map or every traversable part of the map. If a security guard on patrol simply has to ensure the security of five rooms in a building, then his patrolling strategy would be such that he needs only to check the doors to the five rooms rather than checking the whole building. Likewise if he were to ensure the security of all items in a museum, then his patrolling strategy would be one that covers the whole building. Our study focuses on the case where the patrolling has to be done on every traversable part of the map, like in the case where cleaning robots having to continually cover every segment of an office space as dust accumulates.

The *Multi-Agent Patrolling Problem* [6] is an interesting area of research which has surprisingly not received its due attention from the robotics community. The closest problem which can be used for analysis is the *Watchman Route Problem* [7]. A good comparison of several strategies were done in [8]. However, one of the main assumptions that most researchers who are working on the *Patrolling Problem* and its similarly related counterparts is that the number of agents as well as the environment in question remains unchanged. There are some strategies which are adaptive to the group size, the environment size and type [9][10].

Most employed strategies require complete knowledge of the topology of a given map. These strategies are unable to adapt to an unknown or a dynamically changing map as well as to a dynamically changing number of active agents. And the few strategies which are adaptive require real-time information on how the map and the agent number is varying. Many of these strategies are complicated and computationally expensive and the solution has to be constantly recalculated in the case of a varying map.

We propose a *Probabilistic Ants* (PAnts) algorithm, which utilizes the laying of virtual pheromone trails as an indication of how an unknown map has been traversed by a group of agents [11][12][13][14]. Each agent plans its next move in a probabilistic manner based purely on its local virtual pheromone information. The behavior of the conventional Ants algorithm produces a rather inefficient coverage strategy where there is a high overlapping of tasks of the agents. We introduce a variant of the Ants algorithm (with biased motion) for comparison as well. This variant exhibits a more 'pack-like' motion of the agents. Our proposed *Probabilistic Ants* (PAnts) algorithm is a balance between both. We compare our proposed strategy with the conventional Ants algorithm and our introduced variant of it. All three methods are also benchmarked against a theoretical ideal for better clarity of the results.

## II. PROBLEM FORMULATION

In this section, we describe the nature of the problem and any assumptions made. The task being considered is the continual patrolling of an unknown map. Each cell of a map is considered patrolled when an agent traverses it, rather than just being in the line of sight. All agents start at the same corner of the map. Each agent is assumed to have perfect holonomic control. Each agent is able to accurately localize itself with respect to its starting point. Each agent is only able to sense its neighboring cells and their respective pheromone levels. A central memory is used to keep track of the pheromone levels in the environment rather than depositing actual chemical pheromone traces. Another assumption is that the agents have free access to this central memory.

The first thing which needs to be identified in a patrolling task is which areas of a map are needed to be visited. It could be every traversable part of the map, or it could just be selected portions of the map. The whole search space can always be replaced by a graph representing the possible routes. Voronoi diagrams, visibility graphs and C-cells can

James Guo Ming Fu (james.fu@nus.edu.sg) and Marcelo H. Ang Jr (mpeangh@nus.edu.sg) are with the Department of Mechanical Engineering, National University of Singapore, 9 Engineering Drive 1, Singapore 117576

be used to generate such a graph [15]. Once a given map is represented by such a graph with nodes and edges, any algorithm developed to solve the *Patrolling Problem* can be used on similar problems which can be represented by such a graph with nodes and edges.

Once a graphical representation of the problem has been established, the patrolling task can now refer to the continuous visiting of all nodes on the graph such that the time delay between two visits of each node is minimized. In such a representation, the nodes can refer to important points which have to be visited and the edges can correspond to the cost (in our case would be the distance) of traveling from one node to the other. As we are interested in covering every traversable part of the map, every cell of the map can also be considered as a node. We assume that any map can be divided to square cells and the cell is considered traversed when an agent moves into it.

The goal of the problem is to have all the agents (as a whole) to continually traverse the map such that the idle time of all cells within the map is kept to a minimum. We adopt from [15] a common benchmark and evaluation criteria for strategies to the *Patrolling Problem*.

A single cycle is the time an agent takes to travel between two nodes, given that it is of unitary edged length between these two nodes, or the distance between the connected nodes is one. Then, the *instantaneous node idleness* would be the number of cycles that a node has remained unvisited. The *instantaneous node idleness* would be measured at each cycle. Thus, the longer it takes for an agent to revisit the same node, the *instantaneous node idleness* of that particular node would simply accumulate over time until the agent eventually revisits that node, in which case the *instantaneous node idleness* would be zero.

The *instantaneous graph idleness* is the average *instantaneous idleness* of all the nodes throughout the graph at a given cycle. Intuitively, the aim of the *Patrolling Problem* would be to ensure that, after a certain amount of time, the *instantaneous graph idleness* does not escalate without bounds and instead reach a steady state. The more effective algorithms would be able to keep the *instantaneous graph idleness* at steady state to the minimum.

The *average graph idleness* is the average *instantaneous graph idleness* over an $n$-cycle simulation. This is the most general criterion in measuring the effectiveness of a particular multi-agent patrolling architecture.

The *worst idleness* is the largest value of *instantaneous node idleness* which occurred throughout the whole $n$-cycle simulation. This is a necessary criterion to be looked at as there may be some scenarios of the *Patrolling Problem* where some key locations cannot be left unattended for too long even though the overall *graph idleness* can be considerably low. An example would be in the case where an agent has to ensure that the temperature in a number of boiler rooms cannot exceed a threshold temperature.

The goal of the *Patrolling Problem* is thus to minimize both the *average graph idleness* and the *worst idleness* of any given map.

## III. Limitations of Currently Used Strategies

Few algorithms in literature directly addresses this problem. A similar problem which can be analyzed is the *Watchman Route Problem* (WRP). The WRP is only concerned for a single tour of the graph whereas the *Patrolling Problem* involves an infinite tour of the graph. The WRP is essentially an optimization problem in computational geometry where the objective is to compute the shortest route that a watchman would have to take in a given map with obstacles to ensure that he covers the whole area in a single tour. Intuitively, if one can solve the WRP, one only needs to repeat the same tour infinite times to solve for the *Patrolling Problem*. However, this will only hold for the case where the topology of the map is known and time-invariant and the number of active agents also remains time-invariant. Most literature tackling the WRP usually break it down to two further problems [16][17]: (1) The *Art Gallery Problem* (AGP) or *Museum Problem* - to find a set of locations such that all points on the map can be viewed from these locations. Followed by (2) the *Traveling Salesman Problem* (TSP) - to find the optimum path which connects all the locations as found from solving the AGP.

Most of the solutions currently being used to treat the *Patrolling Problem* or similar problems (i.e. TSP) are not only complicated but also require high computation time. Many solutions also require some form of learning and thus the map needs to undergo some form of pre-processing to obtain an optimal solution [10][20]. Furthermore, an optimal solution for the WRP may not be the optimal solution for the *Patrolling Problem*. For cases where the map is known and not dynamic, having a solution which require a high computation time is acceptable as the computation is done only once and the agents on patrol simply execute the same strategy repeatedly. However, if the nature of the map is dynamically changing and the number of available agents is also dynamic, a strategy which is simple and not so computationally expensive would thus be preferred. This strategy must also be robust enough to be able to deal with the case where the full information of the environment is unknown. This is one of the main motivations for our proposed strategy.

## IV. Existing Algorithms

Two algorithms have been selected for comparison with our proposed *Probabilistic Ants* (PAnts) algorithm, namely the conventional *Ants* algorithm and the *Biased Ants* algorithm (a modified variant of Ants which we have used). These algorithms share the same assumptions. The algorithms' decision making are done distributively based only on local information. Thus, there is no need of any pre-processing of the map. As the decision making is based only on local information, it is also assumed that these algorithms will be able to cater to a dynamically changing map with changing number of agents.

### A. Ants

The Ants algorithm involve ants leaving pheromone traces as they move around an unknown map. These pheromone traces are an indication of how often that cell has been traversed. The ants are constantly drawn towards cells with lower pheromone levels. If more than one cell have the same amount of pheromone level, the agent simply chooses the cell at random. It has been shown in [21] that the agents will eventually cover the whole map as long as the free space is continuous. This would also imply that the agents can continuously perform the patrolling task.

Details of the pseudocode are provided in Algorithm 1.

**Algorithm 1** Ants

```
1: increase current cell pheromone level by 1.
2: Move to adjacent cell with the lowest pheromone level
3: if there is more than one cell sharing the lowest
   pheromone level then
4:     randomly pick one of the cell to move to.
5: end if
6: go to 1
```

As only local information is used, each agent's complexity is kept to a minimal, resulting in robustness and adaptability. Since a priori knowledge of the map is not required, this strategy solves any given continuous map. However, due to the presence of randomness in the algorithm, decision-making is sub-optimal. This algorithm does not work well on maps with bottle necks or with many rooms as agents will take a while to reach the more secluded parts of the map.

### B. Biased Ants

This is a variant of the Ants algorithm. The only difference is the absence of the randomness involved. Random motion in patrolling often creates non-uniformity in the pheromone distribution, creating pockets of local minima. Removing the randomness offers more predictability in the algorithm. The disadvantage is its inherent inefficiency in certain maps.

The only difference from the conventional Ants algorithm is in the case where there are more than one adjacent cell with the lowest pheromone level. Instead of moving to a random cell with the lowest pheromone level (step 4 of Algorithm 1), the agent would simply move to a cell in an ordered preference list, i.e. North, East, South, West.

## V. Our Probabilistic Ants (PAnts) Algorithm

Our proposed strategy utilizes the concept of pheromone traces with pheromone decay. Decision making is done in a weighted probabilistic manner, which makes the algorithm adaptable to various maps, minimizing the effects of any local minima formed in the pheromone distribution. No knowledge of the map is required. However, a central virtual memory is required to track the global pheromone levels. As each agent's decision making is based purely on local pheromone levels, this strategy can cater to a changing map with changing number of agents. Our simulations have shown improvement over the compared algorithms.

### A. Pheromone Deposit and Decay

The conventional Ants algorithm increments a cell's pheromone level by one when an agent visits it. The pheromone level does not accurately reflect the cell's idleness, which is the main objective function that a patrolling strategy should minimize. We thus introduce pheromone decay in every cell. This would indicate how often and how recent a cell has been visited. As such, pheromone deposit during an agent's visit to a cell has to be much larger for the cell's pheromone level to have any significance after a substantial period of decay. The pheromone deposit during an agent's visit to a cell should depend on the map size and the agent number. For simplicity, we shall take the pheromone deposit, $Ph_d$, to be $\frac{m}{n}$, where $m$ is the number of open cells within the map and $n$ is agent number. When an agent visits a cell, the pheromone level would be updated as

$$Ph_{x,y}(t) = Ph_{x,y}(t-1) + Ph_d \tag{1}$$

where $Ph_{x,y}(t)$ is the pheromone level of a cell at coordinate $(x, y)$ at time $t$.

Pheromone decay occurs at each simulation cycle and each cell in the map would have its pheromone level updated by

$$Ph_{x,y}(t) = \lambda \times Ph_{x,y}(t-1) \tag{2}$$

where $\lambda$ is the pheromone decay rate.

The combination of pheromone deposits by the agents as well as pheromone decay over time will eventually build a continuous potential field [18] across the map where agents are pulled towards areas of lower pheromone levels or areas of higher idleness.

### B. Probabilistic Decision Making

When the whole map eventually gets traversed, it will be covered with pheromone demarcating where the agents have last been. So at a certain time $t$, an agent has four choices (north, south, east or west) in deciding what its next move should be. It would also make its decision based on the pheromone levels which act as weights in its probabilistic decision making. Cells with lower pheromone levels would have higher weights and thus have a higher probability of being selected as the next cell to move to. To place an even higher weight on cells with lower pheromone levels, an exponential distribution is used. The weight of a cell would thus be updated by

$$W_{x,y}(t) = e^{-Ph_{x,y}(t)} \tag{3}$$

where $W_{x,y}(t)$ is the weight of the cell.

### C. The Algorithm

When the agents first traverse the map, it adopts the strategy of the Biased Ants, i.e. cells are chosen from an ordered list [North, East, South, West] when there are neighboring cells with zero pheromone levels. This is a quick way for the agents to traverse the map and deposit the first round of pheromone. After which, it will commence with the probabilistic-decision making when an agent is no longer presented with neighboring cells with zero pheromone levels.

The PAnts algorithm is broken down into two parts:

**Agent Navigation**: Each agent deposits pheromone traces by $Ph_d$ as it moves around the map. This is communicated to the central memory. Each agent also has access to the pheromone levels of the entire map via the central memory. The agent then probabilistically decides which cell to move to based on the weights of the neighboring cells. Details of the pseudocode are provided in Algorithm 2.

**Algorithm 2** PAnts - Agent Navigation

```
increase current cell pheromone level by Ph_d
if there is at least one adjacent cell with zero pheromone level then
    move to the cell with zero pheromone level based on an ordered list, i.e. [North, East, South, West]
else
    for all adjacent unoccupied cell do
        W_{x,y}(t) = e^{-Ph_{x,y}(t)}
    end for
    randomly move to a cell based on its weights
end if
go to 1
```

**Map Update**: Pheromone levels of all cells decay in the central memory based on equation 2 at each simulation cycle.

### D. Selection of Parameters

The presence of pheromone decay in every cell over a long period may result in an inaccurate decision making process due to the nature of the exponential distribution of weights. If the pheromone decay causes the cells' pheromone levels to drop too low (near zero), the weights, $W_{x,y}(t) = e^{-Ph_{x,y}(t)}$, would in turn tend to 1. If a few cells have the resulting probabilistic weights of 1, they would then have equal chances of being selected, which implies that they have equal idleness. This is not always true and should be avoided.

A cell's pheromone level is affected by (1) the pheromone decay rate, $\lambda$, (2) the amount of pheromone which an agent deposits, $Ph_d$, and (3) the duration of the cell left idle. The duration of a cell being left idle depends on the size of the map and the number of agents. Thus, we approximate the cell's idleness to be $(\frac{m}{n} - 1)$, where $m$ is the size of the map and $n$ is the number of agents used for patrolling. The reason for the subtraction of 1 is that the agent is now next to the cell, rather than returning to that cell.

A cell's pheromone level after an idle time of $(\frac{m}{n} - 1)$ would be $Ph_d \times \lambda^{(\frac{m}{n}-1)}$. If constrain the cell's pheromone level to be more than a certain threshold, T, we now have

$$Ph_d \times \lambda^{(\frac{m}{n}-1)} > T \tag{4}$$

$$\lambda > (\frac{T}{Ph_d})^{\frac{1}{\frac{m}{n}-1}} \tag{5}$$

The essential parameters must thus be selected in order for the probabilistic decision process to be effective.

A cell with a calculated probabilistic weight of $e^{-0.01} = 0.990$ is acceptable. We thus set the threshold of the pheromone level at 0.01 for the purpose of our simulations. The largest map used for testing is of size 2500. The minimal agents used is 5. In our simulations, we have set the pheromone deposit to be $Ph_d = \frac{m}{n}$. The constraint of the pheromone decay is thus calculated to be

$$0.9786 < \lambda < 1 \tag{6}$$

### E. Robustness and Adaptability

Our PAnts algorithm is an online method which only relies on local information during decision making. As its decision making is independent on the decisions made by other agents as well as the topology of the map, PAnts is able to dynamically adapt to any given map or if the topology of the map changes in real-time. This also holds true if the number of agents dynamically changes.

## VI. EXPERIMENTAL RESULTS

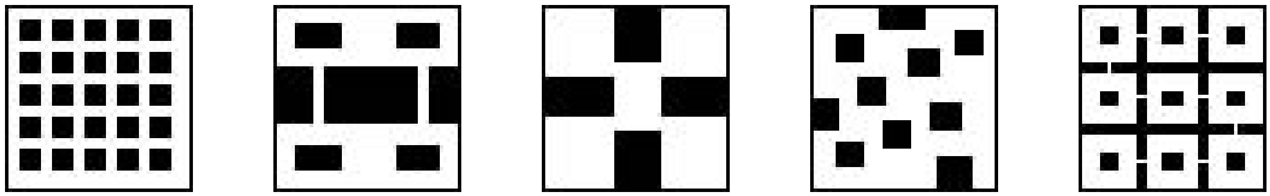
A GUI in *MATLAB* was created to run the algorithms and to provide a platform to vary the agent number (5, 10, 15 and 20) and the map type. Six maps (Fig. 1) were created to encapsulate the possible scenarios for patrolling. Agents start from a clustered position at the bottom left of each map. A shared memory of pheromone placement in the map among the agents was used. There is no communication between agents. For each simulation run, the agents patrolled for 5000 cycles. Both the *average graph idleness*(AGI) and the *worst idleness*(WI) for each run were recorded. These two values were only monitored after the agents have collectively covered the map once. As we are interested in the performance of the strategies in patrolling rather than exploring, data relating to the explorative phase was not considered. 10 simulation runs were done for each possible permutation and the overall AGI and the overall average WI were recorded. Fig. 2 and Fig. 3 show the AGI and the average WI respectively when using the different algorithms for the six different maps.

The lowest theoretical AGI and WI are appended to the graphs. For a map with $m$ open cells, the lowest theoretical WI (assuming agents still take 1 step to teleport from one cell to another) for $n$ patrolling agents would simply be $(\frac{m}{n} - 1)$ steps. The lowest theoretical AGI would be $[0 + 1 + 2 + ... + (\frac{m}{n} - 2) + (\frac{m}{n} - 1)]/\frac{m}{n} = \frac{1}{2}(\frac{m}{n} - 1)$.

Three different pheromone decay values were used for our *Probabilistic Ants* (PAnts) algorithm. Since $0.9786 < \lambda < 1$ (from equation 6), we used the values of 0.98 and 0.995. We also used the value of 0.97 to see how the algorithm would fair if the $\lambda$ value was chosen just below the constraint. The three variants of the PAnts algorithm are labeled as PAnts-97, Pants-98 and PAnts-995.

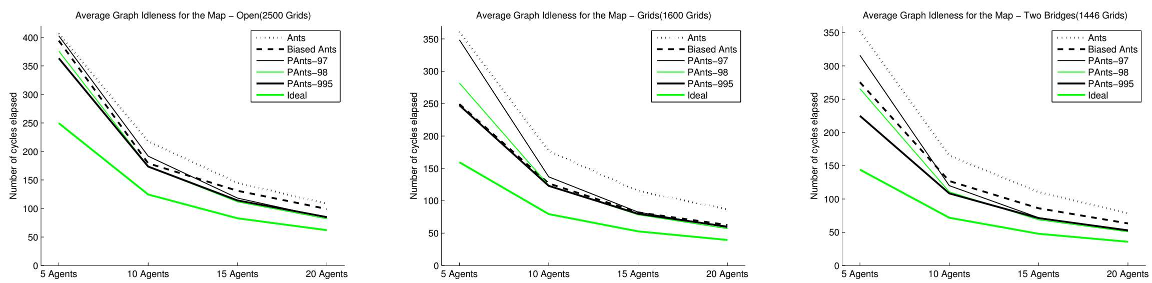
The worst idleness, although being lower than that of Ants and Biased Ants, appears to be rather high. This is due to the probabilistic nature of our algorithm and that only the worst idleness of each simulation run is recorded.

PAnts-995 outperforms the other algorithms and also proves to be the best variation of the PAnts algorithm. PAnts-995 exhibits the lowest AGI and WI on all six maps. The difference in the AGI and WI values between PAnts-995

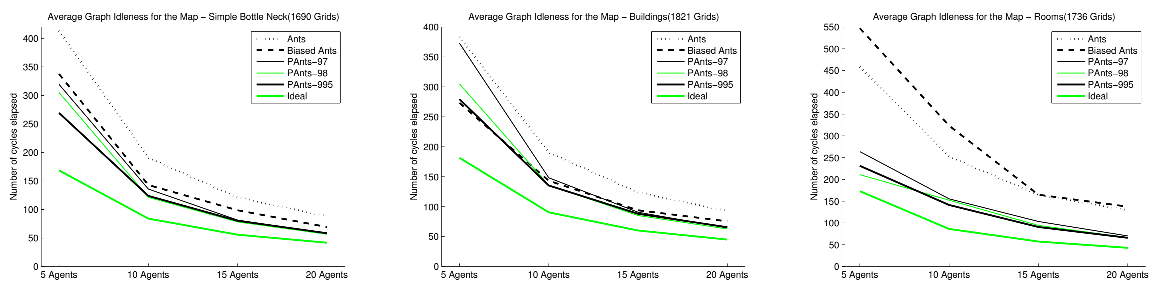


(a) Grided Map(1600) (b) Two Bridges(1446) (c) Simp. Bottle Neck(1690) (d) Buildings(1821) (e) Rooms Map(2500)

Fig. 1. Maps used for simulations. The bracketed numbers represent the total number of open cells in that particular map. Each map is 50 by 50 cells. A 50 by 50 open map was also used but not reflected here. (a) This map has a total of 25 obstacles (or holes) placed in an ordered manner. This arrangement allows each part of the map to have an equal likelihood of being reached. (b) This map is split into two portions (top and bottom) with two corridors linking the map portions. (c) This is an alteration of the Open Map. There are no obstacles (or holes) within the map. (d) This is an alteration of the Grided Map where the layout of the obstacles (or holes) are not in any ordered manner. (e) This map features 9 inter-connecting rooms where the only way to move from the room in the bottom-left corner to the room in the top-right corner is through the other 7 rooms.




(a) Open Map - 2500 Open Grids (b) Grided Map - 1600 Open Grids (c) Two Bridges - 1446 Open Grids




(d) Simple Bottle Neck - 1690 Open Grids (e) Buildings - 1821 Open Grids (f) Rooms - 1736 Open Grids

Fig. 2. Average Graph Idleness for the 6 different maps.

and the theoretical ideal diminishes with increasing agent number, reinforcing the innate cooperative ability which the algorithm is able to exhibit with increasing agent number.

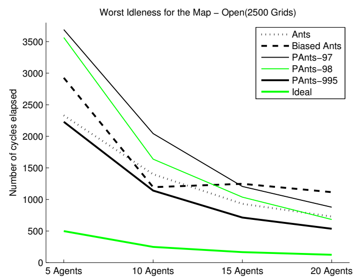
It is also interesting to note that even though the $\lambda$ value of PAnts-97 does not fulfil the constraint in equation 6, the algorithm still performs fairly well.

## VII. CONCLUSION

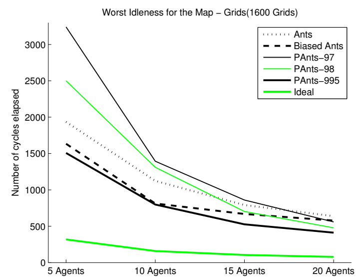
In this paper, we introduced a new algorithm, *Probabilistic Ants* (PAnts) for the *Multi-Agent Patrolling Problem*. We have shown its effectiveness by comparing the *average graph idleness* and the *worst idleness* with two other algorithms (Ants and Biased Ants) across six different maps. As the algorithm is robust and only relies on local information, it can work for a dynamically changing map as well as changing agent number. The algorithm is simplistic in nature and is not computationally demanding. The probabilistic nature of the algorithm has proven that it works well for different maps. The conventional Ants algorithm does not work very effectively in maps with choke points and bottle necks. PAnts, however, is able to overcome this.

Using virtual pheromones is non-intrusive to the environment. This is very practical as compared to the many multi-agent strategies which manipulates the environment, i.e. leaving markers in the environment [19].

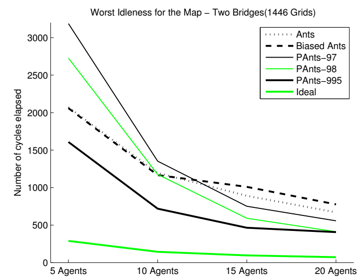
In the future, we hope to explore how to select the pheromone decay value, $\lambda$, to achieve the most efficient patrolling behavior. Another challenge would be for each agent to be able to probabilistically predict the pheromone deposit of other agents and hence do away with the need to centrally maintain the global pheromone levels.



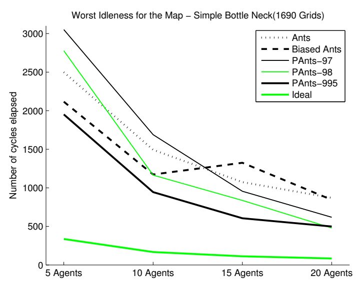

(a) Open Map - 2500 Open Grids



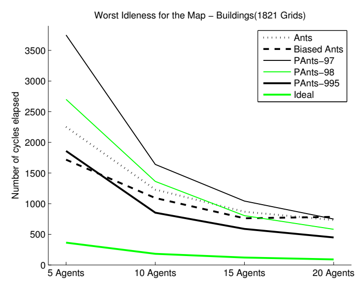

(b) Grided Map - 1600 Open Grids



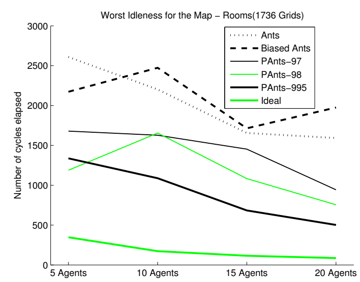

(c) Two Bridges - 1446 Open Grids




(d) Simple Bottle Neck - 1690 Open Grids




(e) Buildings - 1821 Open Grids




(f) Rooms - 1736 Open Grids

Fig. 3. Average Worst Idleness for the 6 different maps.

## REFERENCES


[1] Kaveh Albekord, Adam Watkins, Gloria Wiens, Norman Fitz-Coy and Kuo-Chi Lin, "Multiple-Agent Surveillance Mission with Non-Stationary Obstacles", in Proceedings of 2004 Florida Conference on Recent Advances in Robotics, University of Central Florida Orlando, Florida, 2004.

[2] Kjerstin Williams and Joel Burdick, "Multi-robot Boundary Coverage with Plan Revision", in *Proceedings of 2006 IEEE International Conference on Robotics and Automation (ICRA'06)*, Orlando, Florida, pp 1050-4729, 2006.

[3] Danilo Reis, Adriano Melo, Andr6 L. V. Coelho, and Vasco Furtado, "Towards optimal police patrol routes with genetic algorithms", in *Proceedings of the 4th IEEE Intelligence and Security Informatics Conference (ISI)*, San Diego, California, pp 485 - 491, 2006.

[4] Sudarshan S. Chawathe, "Organizing Hot-Spot Police Patrol Routes", in *Proceedings of IEEE International Conference on Intelligence and Security Informatics (ISI)*, New Brunswick, New Jersey, USA, pp 79 - 86, 2007.

[5] Israel A. Wagner and Alfred M. Bruckstein, "Cooperative cleaners: a study in ant robotics," Tech. Rep. 9512, CIS Report, Technion, IIT, Haifa, June 1995.

[6] Yann Chevaleyre, "Theoretical Analysis of the Multi-agent Patrolling Problem", in *Proceedings of the Intelligent Agent Technology, IEEE/WIC/ACM International Conference on (IAT'04)*, pp 302 - 308, 2004.

[7] Wei-pang Chin and Simeon Ntafos, "Optimum watchman routes", *Proceedings of the second annual symposium on Computational geometry*, Yorktown Heights, New York, United States, pp 24 - 33, 1986.

[8] Alessandro Almeida, Geber Ramalho, Hugo Santana, Patrícia Tedesco, Talita Menezes, Vincent Corruble and Yann Chevaleyre, "Recent Advances on Multi-Agent Patrolling", in *17th Brazilian Symposium on Artificial Intelligence*, So Lus, Brazil, pp 474 - 483, 2004.

[9] Francois Sempé and Alexis Drogoul, "Adaptive Patrol for a Group of Robots", in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03)*, Volume 3, pp 2865 - 2869, 2003

[10] Hugo Santana, Geber Ramalho, Vincent Corruble and Bohdana Ratitch, "Multi-Agent Patrolling With Reinforcement Learning", in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, Volume 3, pp 1122 - 1129, 2004.

[11] Israel A. Wagner, Michael Lindenbaum and, Alfred M. Bruckstein, "Cooperative Covering by Ant-Robots using Evaporating Traces", *IEEE Transactions on Robotics and Automation*, Volume 15, Issue 5, pp 918 - 933, 1999.

[12] Tucker Balch and Ronald Arkin, "Avoiding the Past: A Simple but Effective Strategy for Reactive Navigation", in *Proceedings of IEEE International Conference on Robotics and Automation*, pp 678 - 685, 1998.

[13] D. Payton, M. Daily, B. Hoff, M. Howard and C. Lee, "Autonomy-Oriented Computation in Pheromone Robotics", in *Proceedings of Workshop on Autonomy-Oriented Computation (AOC)*, Fifth International Conference on Autonomous Agents, pp 69 - 77, 2001.

[14] Richard T. Vaughan, Kasper Sty, Gaurav S. Sukhatme and Maja J. Mataric, "Whistling in the Dark: Cooperative Trail Following in Uncertain Localization Space", in *Proceedings of the Fourth International Conference on Autonomous Agents*, pp 187 - 194, 2000.

[15] Aydano Machado, Geber Ramalho, Jean-Daniel Zucker and Alexis Drogoul, "Multi-agent Patrolling: An Empirical Analysis of Alternative Architectures", *Multi-Agent-Based Simulation II: Third International Workshop*, MABS 2002, pp 155 - 170, 2002.

[16] Jan Faigl and Miroslav Kulich, "Sensing Locations Positioning for Multi-robot Inspection Planning", *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)*, pp 79 - 84, 2006.

[17] Miroslav Kulich and Jan Faigl, Libor Preucil, "Cooperative planning for heterogeneous teams in rescue operations", *IEEE International, Safety, Security and Rescue Robotics, Workshop*, pp 230 - 235, 2005.

[18] H. Van Dyke Parunak, Sven Brueckner, John Sauter and Jeff Posdamer, "Mechanisms and Military Applications for Synthetic Pheromones", in *Proceedings of Workshop on Autonomy Oriented Computation*, 2001.

[19] Jonas Svennebring and Sven Koenig, "Trail-Laying Robots for Robust Terrain Coverage", in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'03)*, pp 75 - 82, 2003.

[20] Fabrice Lauri and Franois Charpillet, "Ant Colony Optimization applied to the Multi-Agent Patrolling Problem", in *Proceedings of Swarm Intelligence Symposium, 12/05/2006, Indianapolis, Indiana, USA*, IEEE (editor), 2006.

[21] Jonas Svennebring and Sven Koenig, "Building terrain-covering ant robots: A feasibility study." *Auton. Robots*, Volume 16, No. 3, pp 313 332, 2004.