Proceedings of the IEEE SMC UK-RI Chapter Conference 2006
on Advances in Cybernetic Systems, September 7-8, Sheffield, UK

Optimal Path Planning Applied to Ant Foraging

Anand Veeraswamy, Balasundram P. Amavasai and Stuart Meikle
Microsystems and Machine Vision Laboratory
Materials and Engineering Research Institute
Sheffield Hallam University,
Pond Street, Sheffield S1 1WB, United Kingdom
anandvincent78@hotmail.com, b.p.amavasai@shu.ac.uk, stu@stumeikle.org
http://www.shu.ac.uk/meri/mmvl

Abstract – *Of recent, ant foraging has been of great interest to the robot swarming community. In this paper we implement and study a double pheromone approach to ant foraging. In addition, we introduce the A* path planning algorithm to improve the efficiency of foraging. The Netlogo simulation tool has been used to compare the performance of ant foraging with and without the invocation of an extra path planning ant/agent. With the invocation of path planning, an improvement in foraging is observed, especially with high rates of pheromone evaporation and low rates of diffusion. This could be particularly useful for real robots in a dynamic environment where the rate of evaporation of pheromones could be very high due to various reasons. The introduction of path planning helps the ants/agents to converge to an optimal path in lesser time. Also the number of effective ants required for foraging is fewer. This paper also differs from most previous approaches in that the food source and nest location changes over time which makes the introduction of path planning more effective, since without it the success of ants foraging for food is merely stochastic.*

Keywords: Robotics, artificial intelligence, path planning, ant foraging, swarm intelligence.

1 Introduction

Swarm/collective intelligence is categorised in the “united we stand divided we fall” paradigm. A single ant/agent is hardly of any use by itself. Instead, a swarm of ants are capable of solving complex tasks. This principle is not new to computing. Grid computing makes use of multiple networked computers to perform complex tasks which cannot be performed by a single computer within a specified time. This functional similarity makes swarm engineering very popular for various applications.

The simplicity of the technique used in swarm intelligence makes it suitable for future applications in a variety of domains like bioinformatics, homeland security, web searching and surveillance. For example, NASA has plans to use swarms in space exploration [1].

Other applications include the design of a gripper using the pheromone technique [2]. Here, the Ant Colony Optimization (ACO) method is used to estimate the optimum force with which objects should be held. Ants/agents move along the contact points following the pheromone gradient until an optimum solution is reached. The number of ants on a contact point determines the force exerted by a finger on that point.

1.1 Ant Foraging

Ants are generally very efficient in finding food and building nests by using simple rules. They make use of pheromones, a chemical substance to mark a trail. They roam randomly until they sense the effect of pheromones, at which point of time they will decide to either follow the trail or continue to roam randomly. Using these techniques, ants are capable of finding the shortest possible path to the food source quickly and in a finite amount of time. Social insects like ants communicate very efficiently with their colony members to accomplish complex tasks like finding a path between food source and nest spanning over 100 meters. They achieve this self organisation based entirely on local information [3]. Self-organisation relies mainly on positive feedback, negative feedback, amplification of fluctuations, and multiple interactions [4].

In this paper we propose combining the pheromone approach with path planning techniques to increase the performance of simple mobile robots. By introducing path-planning, the time required to locate the food sources can be reduced due to the removal of the random exploration stage of the standard ant foraging technique. This new hybrid technique can be used in a number of environments, *e.g.* in the case of assembling systems with parts located in multiple locations or in the case of equipment provision to workers in a dynamic industrial environment. In the latter situation, the workers will not have a fixed position, and tools will have to be allocated to these workers as efficiently as possible. In this paper the food source and nest locations changes at fixed intervals to simulate a dynamic environment.

2 Previous Work

This section first discusses virtual pheromone techniques used in real robots and then looks at the simulation based approaches on which the present work is based on.

[5] uses virtual pheromone messaging as a communication technique between a group of robots to achieve tasks like surveillance, reconnaissance, hazard detection and path planning. Each robot is equipped with infrared transmitters and receivers which they use to relay messages between themselves to perform a task collectively. In this way the robots manage to stay together at a certain distance from each other and can also exhibit behaviours like hiding behind a wall and guiding other robots to do the same.

In [6] robots make use of colour sensors to determine the next action based on colour information of the pattern projected on the floor. The authors also reason that at this stage, most researchers do not use chemical pheromones because of 2 main reasons. Firstly it is difficult to manufacture chemical sensors that perform the task according to the requirements and secondly it is difficult to observe how the invisible gases or chemicals spread and affect the robots behaviour.

Moorebots equipped with single odour sensor is used in [7] to sense water vapour. The odour sensor detects the presence of an airborne substance through a change in the electrical resistance of a chemically sensitive carbon-doped polymer resistor. A water plume is generated using a pan of hot water and an array of fans. The authors have described a distributed algorithm by which a group of agents can solve the full odour localisation task more efficiently than a single agent.

[8] combines ubiquitous computing methods with pheromone techniques for a practical application to track everyday objects that may be misplaced. The virtual pheromone used here is in the form of postage-stamp sized radio RFID transceivers that can be attached to objects. Each tag is marked with a unique identifier which can be accessed by multiple RFID readers. The pheromones are created by means of data-structures stored in the RFID tags. This data-structure contains a hop count which is analogous to diffusion and a counter which is analogous to evaporation in real pheromones.

[9] takes a simulation approach presenting an application of genetic programming to search for foraging behaviours. Their work demonstrates that it is possible to have the entire foraging behaviour discovered by the learning system. In an accompanying paper [10] the same authors have presented a hard-coded ant foraging algorithm for more complex environments that contain obstacles. Our present work follows this algorithm very closely.

3 Methodology

In this paper one ant makes use of the A* algorithm to find an optimal path from the nest to the food source. The path planning ant traverses to and fro between the food source and the nest location depositing pheromones along the optimal path that has been found. Since the task of path planning is computationally intensive only one ant is entrusted with this task. Adding more path planning ants was found not to improve the efficiency since only one set of nest and food source was used in all the simulations. In future work we will consider a scenario where ants can forage food from multiple food sources. Here we would need to assign at least one or more path planning ant to each food source.

The other ants are entrusted with exploring the region for food. They do not drop any pheromones until they have found the food source, or have stumbled upon the nest, or have sensed a food or nest pheromone. When they have sensed a food or nest pheromone they add to the pheromone concentration according to the procedures outlined in [10]. Since the pheromone adding mechanism has been proved to be better than one that simply deposits fixed pheromones, we have used the pheromone adding mechanism in all our simulations. In real life, ants decrease or increase the amount of pheromones they lay depending on the availability of food at the food source. Other wise they would just drop a fixed amount of pheromones.

The only function of the path planning ant is to find the optimal path and drop the respective pheromones. They do not transport food. An ant which comes across a pheromone when it is carrying food or is located in the food source follows the nest pheromone trail to reach the nest location. An ant which comes across a pheromone when it is not carrying a food source or is located in the nest location follows the food pheromone trail to reach the food source.

4 The Algorithm

All ants except the path finding ant execute the Ant-Forage procedure and the path planning ant executes the Find-Path-Lay-Pheromones procedure at each time step. The path planning ant is initially located at the nest and waits there until it computes the optimal path to the food source. This is given as follows:

```
Ant-Forage-With-Path-Planning
  Ants [Ant-Forage]

  Path-Planning-Ant [ Find-Path-Lay-Pheromones]
```

The algorithm for ant foraging is similar to the one detailed in [10]. The only difference is that we have used a *conformance* factor rather than the stochastic method for the ant to explore for food. The lower the conformance value the more the probability that the ants will explore, and the higher the conformance value the more the ants will conform to the general behaviour of moving to the location with maximum food pheromones. In general it was observed that a value of 0.9 or 0.95 for the conformance value produced an optimal behaviour for the ants. In our simulations the ants are initially located at random positions within the world.

Like the conformance factor, we have also used a *max-time* factor for ants to reach the nest after finding the food. The max-time factor encourages the ants which are carrying food to explore the region to find the nest. A counter is set when the ant reaches the food source. The counter keeps increasing as the ant goes back to the nest. Ants whose counter value exceeds the max-time exhibit a random movement once in every 5 timesteps. This encourages the ants to explore the region to find the nest.

The Find-Path-Lay-Pheromones procedure implemented by the path planning ant is as follows:

```
Find-Path-Lay-Phermones

  If Path-Not-Found
    Path-List = Find-Path
  Else
    If Located-At-Nest
     Move-To-Food-Source
    If Located-At-Food-Source
    Move-To-Nest

Move-to-food-source

  If path planning ant is located at nest
   Drop Max-Nest-Pheromones
  While-not-reached-food
   Go to next location in the path list (forward)
   Drop Max-Nest-Pheromone - 1
  End

Move-To-Nest

  If path planning ant is located at food source
   Drop Max-Food-Pheromone
  While-not-reached-nest
   Go to next location in the path list (reverse)
   Drop Max-Food-Pheromone - 1
  End
```

The path planning ant is always initially placed in the nest, where it stays till a path is found. It uses the A* algorithm to find the path. The path found starts from the nest as the start point, and the food source as the end point. This explains why the ant moves forward in the Move-To-Food-Source procedure and backward in the Move-To-Nest procedure.

When the path planning ant finds that the food location has changed it goes back to the new nest location and recalculates the path to the new food source. The previous pheromone trail it had established evaporates with time. It finds the optimal path between the new food source and nest location and once again forms a double pheromone trail to guide the other ants.

The A* algorithm is a well-known best-search-first graph algorithm, commonly used to find the destination position in a map, given an initial position. We have previously implemented the A* algorithm in a single robot environment [11]. The algorithm used here is the same as in the previous work except that we use diagonal distances for heuristic calculations rather than Manhattan distances. The additional heuristic suggested in the previous work has not been used due to the complexity of the scenarios used in this work. In future work we will consider using better heuristics or use more efficient search algorithms like the D* algorithm.

5 Implementation

The Netlogo simulation tool has been used to perform the experiments. The parameters used in the simulations are detailed in table 1.

Parameters	Values
Number of ants	10 or 20 or 30
Number of path planning ants	1
Maximum Ants per location	10
Minimum amount of Pheromone	0.0
Maximum amount of Pheromone	1000
Environment	20x20,non-toroidal
Evaporation ratio	99%
Diffusion ratio	1%
Duration of Simulation	5000 time steps
Rate of change of food and nest	400 time steps

Table 1 shows the values of the parameters used in the simulations.

Three different types of obstacles populate the world in our simulations. Figure 1 shows an environment with blob like obstacles.

Figure 1. Ants using path planning to forage in an environment filled with blob like obstacles.

In our simulation the ants originally appear red. They turn brown after finding the food. They turn pink after dropping the food in the nest. The Welch two sample test was used to compare the performance of ant foraging with path planning and without path planning taking 50 independent samples each for different scenarios and number of ants.

Ant Number	Mean	Standard Deviation
10	33.34	12.4566
20	91.96	17.8176
30	151.96	26.5221

Table 2. Mean and Standard Deviation of the amount of food collected over 50 independent runs without the invocationof path planning in the scenario with blob obstacles.

Ant Number	Mean	Standard Deviation
10	123.88	12.1649
20	249.1	27.7446
30	367.1	42.5207

Table 3. Mean and Standard Deviation of the amount of food collected over 50 independent runs with path planning invoked in the scenario with blob obstacles.

Table 2 shows the performance of the ants in the blob scenario without path planning. Table 3 shows the performance of the ants in the same scenario with path planning invoked. The performance of foraging shows a vast improvement with path planning invoked especially when very few ants are available for foraging.

We then carried out the same comparison with a random scattering of obstacles as shown in figure 2

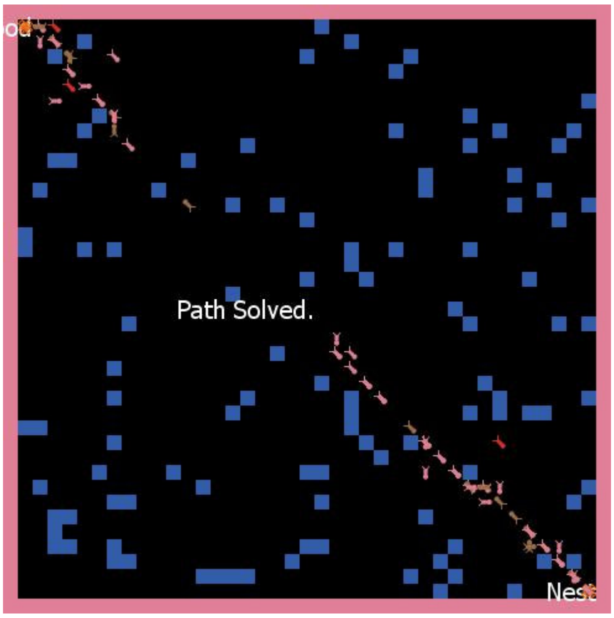


Figure 2. Ant foraging in an environment where obstacles are scattered at random.

This scenario provided a more complex area for foraging and the effect of path planning has been more prominent here than in the previous scenario. Table 4 shows the performance of the ants in the randomly scattered obstacle scenario without path planning invoked. Table 5 shows the performance of the ants in the same scenario with path planning invoked. The efficiency when using path planning is 3 times more than when path planning is not used.

Ant Number	Mean	Standard Deviation
10	27.4	11.7403
20	70.22	22.8243
30	131.14	28.4088

Table 4. Mean and Standard Deviation of food collected without path planning in the random obstacle scenario.

Ant Number	Mean	Standard Deviation
10	126.06	5.3276
20	251.38	12.2888
30	375.08	14.3382

Table 5. Mean and Standard Deviation of food collected over 50 independent runs with path planning in the random obstacle scenario.

The third scenario shown in figure 3 consists of bar like obstacles.

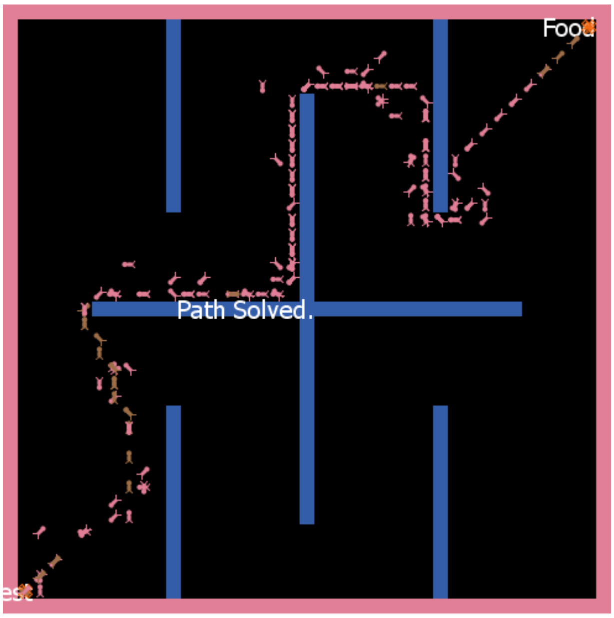


Figure 3. Ant foraging in an environment consisting of bar like obstacles.

Ant Number	Mean	Standard Deviation
10	20.5	5.7008
20	56.26	14.2468
30	111.46	29.0089

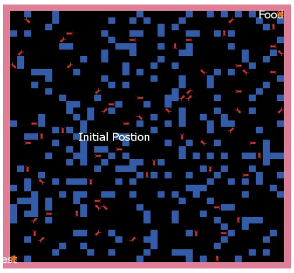
Table 6 Mean and Standard Deviation of food collected over 50 independent runs without the invocation of path planning in the bar like obstacle scenario.

Table 6 shows the performance of the ants in the bar like obstacles scenario without the invocation of path planning. Table 7 shows the performance of the ants in the bar like obstacle scenario with path planning. The increase in efficiency has doubled with the introduction of path planning. In this scenario the time taken to find the path was higher and as a result the invocation of path planning doubles the performance unlike the previous scenario where the efficiency tripled.

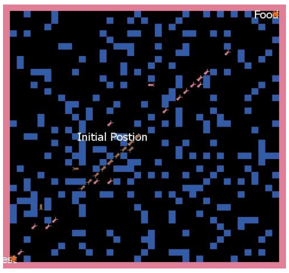
Ant Number	Mean	Standard Deviation
10	69.74	4.3415
20	143.96	5.9520
30	218.76	9.9150

Table 7. Mean and Standard Deviation of food collected over 50 independent runs with path planning invoked in the bar like obstacle scenario.

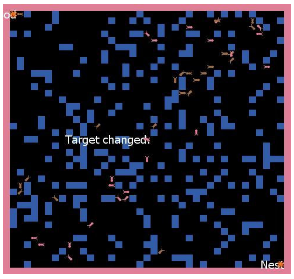
The various timesteps involved in a single example simulation are shown in a step by step manner with diagrams below. Figure 5 shows ant foraging without path planning and the figure 6 shows the same steps with path planning.



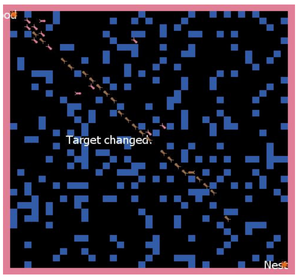
(a) 500 timesteps



(b) 800 timesteps



(c) 1100 timesteps

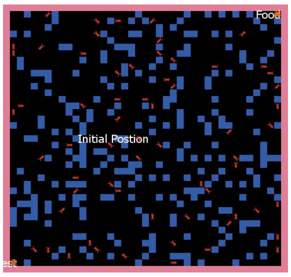


(d) 1500 timesteps

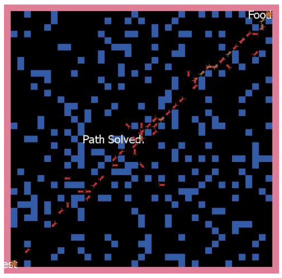
Figure 5. Foraging process without the invocation of path planning, at various when the food source and nest changes locations at 1000 timesteps.

The food source is initially located at the top right hand corner. The food source has been intentionally placed at a position that is rather difficult to reach as it is surrounded in all directions by obstacles, with only one free patch space to access it. The nest is located at the bottom left hand corner. Figure 5(a) shows a snapshot of the simulation at 500 timesteps. All ants are red which means that the food source has not yet been detected by the exploring ants. Figure 5(b) shows the ants converging to an optimum path at 800 timesteps after a random ant chanced upon the food source. The position of the nest and food source change at 1000 timestep. The nest is now located at the bottom right hand corner and the food source is in the top left hand corner. Figure 5(c) shows a snapshot of the simulation at 1100 timesteps. The movement of the ants is now indeterminate as they go about searching for the new food source and nest. The targets are now more easily accessible and the ants have converged to an optimal path in 500 timesteps as shown in figure 5(d)

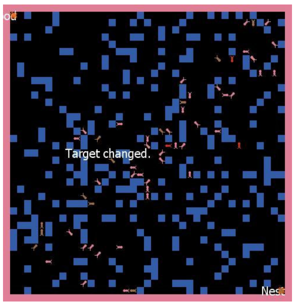
Figure 6 shows snapshots taken at various timesteps to show the same procedure with path planning introduced. The same scenario is considered and the ants and obstacles are placed in the same locations as before. The food source and nest locations change at 400 timesteps in this case because the rate of convergence to the optimal path is much faster with path planning invoked.



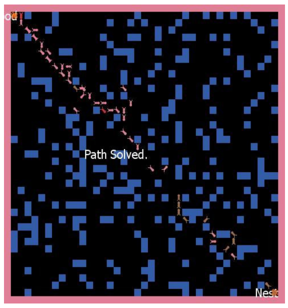
(a) 45 timesteps



(b) 150 timesteps



(c) 450 timesteps



(d) 550 timesteps

Figure 6 shows the foraging process with path planning invoked, at different timesteps with the food source and nest change locations at 400 timesteps

Figure 6(a) shows the ants exploring for food after 45 time steps. The path planning ant has found the food source yet. In figure 6(b) the path planning ant has found the optimal path and all ants converge to the same optimal path in just 150 timesteps as compared to the 800 timesteps without path planning. A snapshot taken just after the target changes and before the path planning ant recalculates the new path is shown in figure 6(c). The ants then converge to the optimal path as shown in figure 6(d) taking just 150 timesteps compared to 500 timesteps without path planning.

Figure 7 shows a graph of food collected versus time for ant foraging with and without path planning. Nine sets of random ant and obstacle positions have been chosen with the same random seed for both methods. Although the increase in performance with path planning invoked is different for different conditions, from the graphs, there is a marked increase in performance in all the nine conditions.

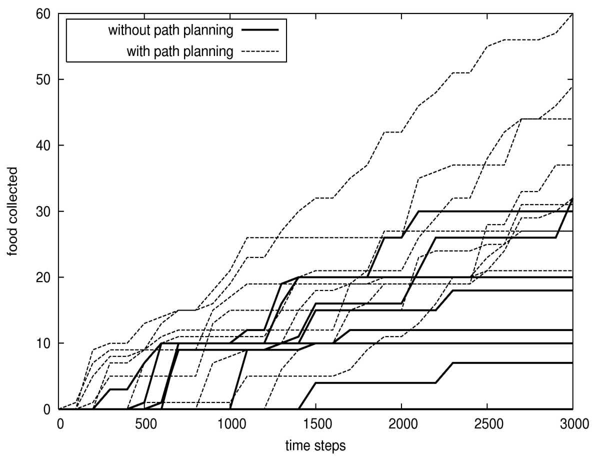


Figure 7. Graph of food item collected versus time for ant foraging with and without the invocation of path planning.

6 Conclusion and future work

In this paper we have shown that the performance of the ant foraging problem can be improved dramatically by the introduction of path planning. Three different scenarios were considered and the efficiency of foraging with and without path planning was compared. All three scenarios demonstrate an improvement in performance. All the simulations in this work were carried out with extreme parameter values, at a very low rate of diffusion and a very high rate of evaporation. The path planning ant acts as a good balancing factor which helps the other ants to work very efficiently even at these extreme conditions. This hybridisation of swarm intelligence with the A* path planning algorithm may provide a very important framework for designing robots which have to work under similar situations in a real world scenario.

The ant foraging process can be made more efficient by encouraging the ants to explore more effectively than the random exploration techniques used in this work. More efficient search algorithms can be used to make the search process faster. In is intended that the same experiment be implemented in a 3-D world. One interesting future work would be to have a swarm of map building robots communicating with a swarm of path planning robots continuously in order to find safe paths dynamically. A swarm of worker robots could then communicate with the path planning robots and take people to safety in sites of natural or man made disasters.

7 Acknowledgements

We would like to thank

- James P. Steiner for so magnanimously allowing us to modify and use his Netlogo A* implementation for our work.
- Liviu Panait for his prompt responses and insight into the ant forage algorithm.
- Anish Mohammed for his very original ideas on path planning.

8 References

[1] M.G. Hinchey, C.A. Rouff, J.L. Rash, W.F. Truszkowski, “Requirements of an Integrated Formal Method for Intelligent Swarms”, Proceedings of the 10th International Workshop on Formal Methods for Industrial Critical Systems FMICS, Lisbon, pp. 125–133, September 2005.

[2] R.A. Abu Zitar, “Optimum gripper using ant colony intelligence”, Industrial Robot: An International Journal, Volume 32, Number 1, 2005, pp. 17-23(7), Jan. 2005.

[3] P. Tarasewich, P.R. McMullen, “Swarm intelligence: power in numbers”, Communications of the ACM, vol 45, no. 8, pp. 62-67, August 2002

[4] E. Bonabeau, M. Dorigo, and G. Theraulaz, “Swarm Intelligence: From Natural to Artificial Systems”, Oxford University Press, New York, 1999.

[5] Payton D, Estkowski R, Howard M, “Compound behaviors in pheromone robotics”, Robotics and Autonomous Systems, Volume 44, Number 3, pp. 229-240, Sept. 2003.

[6] Ken Sugawara, Toshiya Kazama and Toshinori Watanabe, "Foraging Behavior of Interacting Robots with Virtual Pheromone", Proceedings of 2004 1EEElRS.J International Conference on Intelligent Robots and Systems,Sendai, pp.3074-3079, Sep-Oct. 2004.

[7] Adam T. Hayes, Alcherio Martinoli, Rodney M. Goodman, "Swarm Robotic Odor Localization", Robotica archive, Volume 21 , Issue 4, pp.427-441, Aug. 2003.

[8] M. Mamei, and F. Zambonelli, “Spreading Pheromones in Everyday Environments Through RFID Technology”, 2nd IEEE Symposium on Swarm Intelligence, pp.281-288, Piscataway, June 2005.

[9] Liviu Panait and Sean Luke, “Learning Ant Foraging Behaviours”, Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9), Boston, pp. 575-580, Sept. 2004.

[10] Liviu Panait and Sean Luke, “Ant Foraging Revisited”, Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9), Boston, pp. 569-574, Sept. 2004.

[11] A. Veeraswamy and B.P. Amavasai, “Optimal path planning using an improved A* algorithm for homeland security robots”, 24th IASTED International Multi-Conference on Artificial Intelligence and Applications, Innsbruck, pp. 50-55, February 2006.