

Optimization of Route Planning using Simulated Ant Agent System

Kashif Zafar
National University of Computer &
Emerging Sciences, Islamabad
A. K. Brohi Road
H-11/4, Islamabad

Abdul Rauf Baig
National University of Computer &
Emerging Sciences, Islamabad
A. K. Brohi Road
H-11/4, Islamabad

Ayesha Khan
National University of Computer &
Emerging Science, Islamabad
A. K. Brohi Road
H-11/4, Islamabad

Nabeel Bukhari
National University of Computer &
Emerging Sciences, Islamabad
A. K. Brohi Road
H-11/4, Islamabad

ABSTRACT

This research presents an optimization technique for route planning using simulated ant agents for dynamic online route planning and optimization of the route. It addresses the issues involved during route planning in dynamic and unknown environments cluttered with obstacles and objects. A simulated ant agent system (SAAS) is proposed using modified ant colony optimization algorithm for dealing with online route planning. It is compared with evolutionary technique on randomly generated environments, obstacle ratio, grid sizes, and complex environments. The SAAS generates and optimizes routes in complex and large environments with constraints. The SAAS is shown to be an efficient technique for providing safe, short, and feasible routes under dynamic constraints and its efficiency has been tested in a mine field simulation.

General Terms

Swarm Intelligence, Artificial Intelligence

Keywords

Agent, Route Planning, Ant Colony Optimization

1. INTRODUCTION

A swarm is defined as a set of nature inspired mobile agents that collectively carry out a distributed problem solving. Swarm intelligence [1] is the property of a system whereby the collective behaviors of unsophisticated agents interacting locally with their environment cause coherent functional global patterns to emerge.

Ant colony optimization [1] and particle swarm optimization [1] are two major techniques in the family of swarm intelligence. In ACO, ants are able to find their way to and from food, and the method by which they do it is modeled in a new approach called the ant colony optimization algorithm. It is a population based approach for optimization problems. A number of artificial "ants" construct solutions to the problem at hand by the repeated selection of parts from a predefined set of solution components. The only communication between ants is called stigmergy [2] i.e. indirect communication using

pheromone. These ants select components probabilistically biased by heuristic information (a problem specific heuristic measure of a component's utility) and pheromone information, and directly associated with the solution components. The route planning is an important problem in artificial intelligence research and has been addressed over the years and still considered to be a challenging area and requires efficient and robust technique to solve [3-6]. This paper presents simulated ant agents with route planning capabilities. The physical existence of simulated ant agents is conceived in the form of a simulation that runs on computing devices. The ant agents use the modified ant algorithm for finding optimum route between start and the goal.

The major contributions of this paper are; optimization of route planning, multi-objective optimization, and dynamic route planning. The section 2 describes the problem formulation, section 3 explains the simulated ant agent system, section 4 gives the experimentation details, section 5 presents the results and section 6 is the conclusion.

2. PROBLEM FORMULATION

The route planning and optimization of routes have been considered as NP complete problem [7]. Non-deterministic polynomial time complete is a complexity class of problems with two properties, i.e. any given solution to the problem can be verified quickly and if the problem can be solved in polynomial time then every problem in NP class can be solved. Such problems are usually tackled by approximation algorithms [7] and meta-heuristic approach is one of them.

An offline planner generates the complete plan before the task is performed [3]. The traditional offline planners often assume that environment is completely known and they try to find the plan based on shortest distance criteria. They cannot handle dynamic environments and are limited to their initial plan. When environment remains constant throughout the planning process, the route planning is called static offline route planning. The static environment can be known, partially known [9] or unknown and requires different approach for the each category.

On the other hand, an online planner generates a partial plan during the execution of the task. Online planners are mostly used for dynamic route planning. When the state space changes during the planning phase, the route planning is called dynamic online route planning [3]. There are number of constraints to be considered during planning phase. The obstacle avoidance and finding the shortest route for each changing goal are the complex tasks to deal with.

3. SIMULATED ANT AGENT SYSTEM

Ants are tiny insects capable of finding shortest routes from their nest to the food source. These ants heuristically find the shortest route and save time to get the food. Ant colony optimization [1] is the technique inspired by these tiny ants for optimization problems like traveling salesman problem, job scheduling, network routing, and classification. Ant colony optimization has been applied to solve combinatorial complex problems.

This research presents a simulated ant agent system (SAAS) for solving route planning problem both in static as well as in dynamic environments. SAAS has been applied for online planning in dynamic environments with different environmental configurations and constraints. Ant agents have been used in SAAS and they are capable of traversing the environment based on ACO algorithm technique and collectively find the shortest route from the initial state to the goal state by using indirect communication.

The SAAS has been applied for static route planning using modified ACO algorithm. The goal state remains fixed and environment is static and unknown for ant agents. After loading of map with specific environment, the map is prepared for SAAS. The pheromone is initialized randomly to a small value from 0.01 to 0.09. The start state and the goal state are selected and SAAS starts execution. The ant agent starts exploring the environment randomly and the selection of the next node takes place by using the probability of selection formula.

4. EXPERIMENTATION

The experimentation has been divided into two phases, i.e. static and dynamic environment. For static environment, goal state remains static. We use four different size environments, i.e. 20x20, 40x40, 60x60, and 80x80 grid maps as shown in Fig.1. Each grid map has different percentage of obstacles and mines. The starting and the ending points have been fixed for experiments. The number of ants is also fixed for each experiment. After loading of the map, we need to set the parameters for experiment. There is an option for enabling dynamic environment by randomly changing the goal during online planning phase. The first experiment deals with static environment and we use 50 iterations, 1 ant and 30 experimental runs. The results are shown in Table 1. We used different maps and each with different obstacle ratio and grid size. A high obstacle ratio and randomly generated maps have been used to simulate complex environments. The main parameters for the simulation are alpha and beta value that primarily controls the convergence of the optimum solution. We use

different values for alpha and beta for each of the maps. The values for alpha and beta have been varied from 0.1 to 1.0 for each experiment and results have been recorded for comparison.

5. RESULTS

The experiments have been conducted separately for static goal and for dynamic goal environment. The starting point and the goal point remains constant throughout each run. Each run consists of 50 iterations with single ant. The maps have been generated randomly with different obstacle ratios and placement of mines. The environments are from simple to complex. The SAAS has been used to find the optimum route between the starting point and the goal point. The probability of selection for next node depends on the values of alpha and beta. The alpha value represents the contribution of pheromone level and the value of beta represents the contribution of heuristic function. The higher the value of beta, the more is the influence of the distance in the selection of the next node. The figure below shows the effect of changing the values for alpha and beta on the convergence of SAAS. The paths converge to the optimum values with increasing beta value. The experiments have been conducted for 20x20, 40x40, 60x60, and 80x80 map sizes and the obtained results are consistent in all the maps. The performance of SAAS remains consistent with the increase in map size. The experiments are conducted with 200 iterations, 5 dynamic targets generated randomly on a 20x20 grid size as shown in Table 2. The intervals are fixed to 40 for each run. The distance of the new goal along with the distance from the previous goal was taken into account for comparison. The path traversed by SAAS has been compared with both the heuristics i.e. Manhattan distance as well as the Euclidean distance. Each experiment is conducted with different obstacle ratios and map sizes. The tracking of moving target has been implemented using simulated ant agent system with different parameters and for different grid sizes. Each experiment uses 5 different goals generated at random locations with 200 iterations as one run. The Table 2 shows the results for each goal and compares the route generated by SAAS as cells traversed with Manhattan distance as well as with the Euclidean distance. The experimentation uses different obstacle ratio, grid sizes and heuristic measure. The simulation has been tested with different options to generate new goals randomly. In option 1, we select the number of time to change the goal. In option 2, we provide the percentage of goal change during the complete run of iterations while the third option provides the exact gap between iterations to change the goal. Finally, the results obtained by SAAS in a static environment are compared with evolutionary technique using genetic algorithm as shown in Table 1.

Table 1. SAAS and GA with average of 30 runs

Map size	Alpha	Beta	SAAS	GA
20x20	0.2	1	27.52	28.65
40x40	0.1	1	54.57	66.78
60x60	0.6	0.9	185.62	212.32
80x80	0.1	1	424.56	753.56

The paths are represented as chromosomes and a population size of 20 has been used. The mutation and one point crossover have been used along with the fitness function. A separate module for route optimization has been implemented to repair the generated routes. The number of v-edges are counted in each generated route and fixed by taking the route with same number of cells. It does not reduce the number of cells traversed. It only increases the smoothness of the route. By replacing the v-edge with the adjacent cell, the route becomes smooth and straight. The route optimizer compares all the combinations of the v-edges along the route and removes the v-edges by replacing with diagonal cells along with keeping same cells and distance. The SAAS not only finds the shortest distance route, it also optimizes the generated route as

shown in Figure 1 and Figure 2. The route for each goal in the moving target search has also been optimized by using route optimizer. The routes generated by moving target search are available in the menu list with number of v-edges in each route. By applying the route optimizer, we repair each route and it provides clear and smooth routes from start state to each of the goals in the moving target search. The diagonal movement generates v-edges that need to be repaired by using route optimizer. The smoothness of the routes depends on the number of v-edges and clearness of the route depends on the obstacle avoidance and avoiding mines

Table 2. Comparison of simulated ant agent system for moving target search with A* algorithm

Map = 20x20	Moving Target Search using Simulated Ant Agent System (alpha=0.1 & beta=0.9)									
Iterations = 200										
Case	1	2	3	4	5	6	7	8	9	10
Initial State	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
Goal 1	(3,11)	(0,18)	(1,15)	(1,14)	(0,18)	(3,11)	(2,15)	(6,18)	(7,16)	(6,19)
Manhattan Distance	14	18	16	15	18	14	17	24	23	25
Cells traversed-SAAS	11	18	15	14	18	11	15	18	16	19
Cells traversed-A*	11	18	15	14	18	11	15	18	16	19
No. of iterations	40	40	40	40	40	40	40	40	40	40
Goal 2 random	(2,10)	(3,16)	(2,17)	(4,13)	(1,17)	(2,10)	(4,13)	(5,16)	(5,18)	(3,16)
Distance from Start	12	19	19	17	18	12	17	21	23	19
Distance from Goal 1	2	5	3	4	2	2	4	3	4	6
Cells traversed-SAAS	10	16	17	13	17	10	13	17	18	16
Cells traversed-A*	10	16	17	13	17	10	13	17	18	16
No. of iterations	40	40	40	40	40	40	40	40	40	40
Goal 3 random	(1,7)	(1,14)	(4,15)	(2,10)	(2,18)	(1,7)	(3,11)	(8,13)	(8,16)	(1,14)
Distance from Start	8	15	19	12	20	8	14	21	24	15
Distance from Goal 2	4	4	4	5	2	4	3	6	5	4
Cells traversed-SAAS	7	14	20	10	18	7	11	13	16	14
Cells traversed-A*	7	14	16	10	16	7	11	13	16	14
No. of iterations	40	40	40	40	40	40	40	40	40	40
Goal 4 random	(3,4)	(3,11)	(1,12)	(1,9)	(4,17)	(3,4)	(2,9)	(5,11)	(7,18)	(3,12)
Distance from Start	7	14	13	10	21	7	11	16	25	15
Distance from Goal 3	5	5	6	2	3	5	3	5	3	4
Cells traversed-SAAS	4	11	12	9	18	4	9	11	19	12
Cells traversed-A*	4	11	12	9	17	4	9	11	19	12
No. of iterations	40	40	40	40	40	40	40	40	40	40
Goal 5 random	(6,1)	(6,8)	(2,9)	(2,7)	(5,15)	(6,1)	(1,8)	(3,9)	(6,15)	(6,11)
Distance from Start	7	14	11	9	20	7	9	12	21	17
Distance from Goal 4	6	6	4	3	3	6	2	4	4	4
Cells traversed-SAAS	6	10	9	7	17	6	8	9	15	11
Cells traversed-A*	6	10	9	7	17	6	8	9	15	11
No. of iterations	40	40	40	40	40	40	40	40	40	40

The clearness of the route has been incorporated in the fitness function but the smoothness of the route requires a separate module. The main objective of this module is to repair the path without altering the original route plans. For this purpose, each generated route has been compared with number of cells traversed for each of the repaired path and the resultant selected route will have the same number of cells and distance measure.



Figure 1. Shortest path without optimization



Figure 2. Shortest path with optimization

6. CONCLUSIONS

The simulated ant agent system has been used successfully for route planning and optimization of routes in static and dynamic environments. The SAAS has been tested with simple to complex environments and found to be robust, efficient and scalable online line route planning system. The route optimizer repairs the generated routes while keeping the number of cells and distance same. For dynamic environments, the SAAS has been compared

with moving target search algorithm and proved to be an efficient method for dealing with dynamic environment. The SAAS can be further tested for other constraint satisfaction and multi-objective optimization problems in different application areas.

7. REFERENCES

- [1] A. P. Engelbrecht. Computational Intelligence; An introduction. John Wiley & Sons, ISBN 0-470-84870-7, 2002.
- [2] M. Dorigo and K. Socha. An introduction to ant colony optimization. IRIDIA, Bruxelles, technical report series, 2007.
- [3] J. Xiao, Z. Michalewicz, L. Zhang, and K. Trojanowski. Adaptive evolutionary planner/navigator for mobile robots. IEEE Transactions on Evolutionary Computation, vol. 1, no.1, 1997.
- [4] Z. Qiang and Z. Xing. Global path planning approach based on ant colony optimization algorithm. LNCS, Springer-Verlag, 2006.
- [5] N. A. Vien and N. H. Viet. Obstacle avoidance path planning for mobile robot based on Ant-Q-reinforcement learning algorithm. ISNN, LNCS, Springer-Verlag, 2007.
- [6] H. Mei, Y. Tian, and L. Zu. A hybrid ant colony optimization algorithm for path planning of robot in dynamic environment. International Journal of Information Technology, vol.12, no.3, 2006.
- [7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to algorithms. MIT Press, Cambridge, MA, ISBN-81-203-2141-3, 2001.
- [8] S. J. Russell and P. Norvig. Artificial intelligence-A modern approach. Prentice Hall, Englewood Cliffs, New Jersey, ISBN 0-13-103805-2, 1995.
- [9] A. Stentz. Optimal and efficient path planning for partially-known environments. In Proceedings of the IEEE International Conference on Robotics and Automation, May 1994.